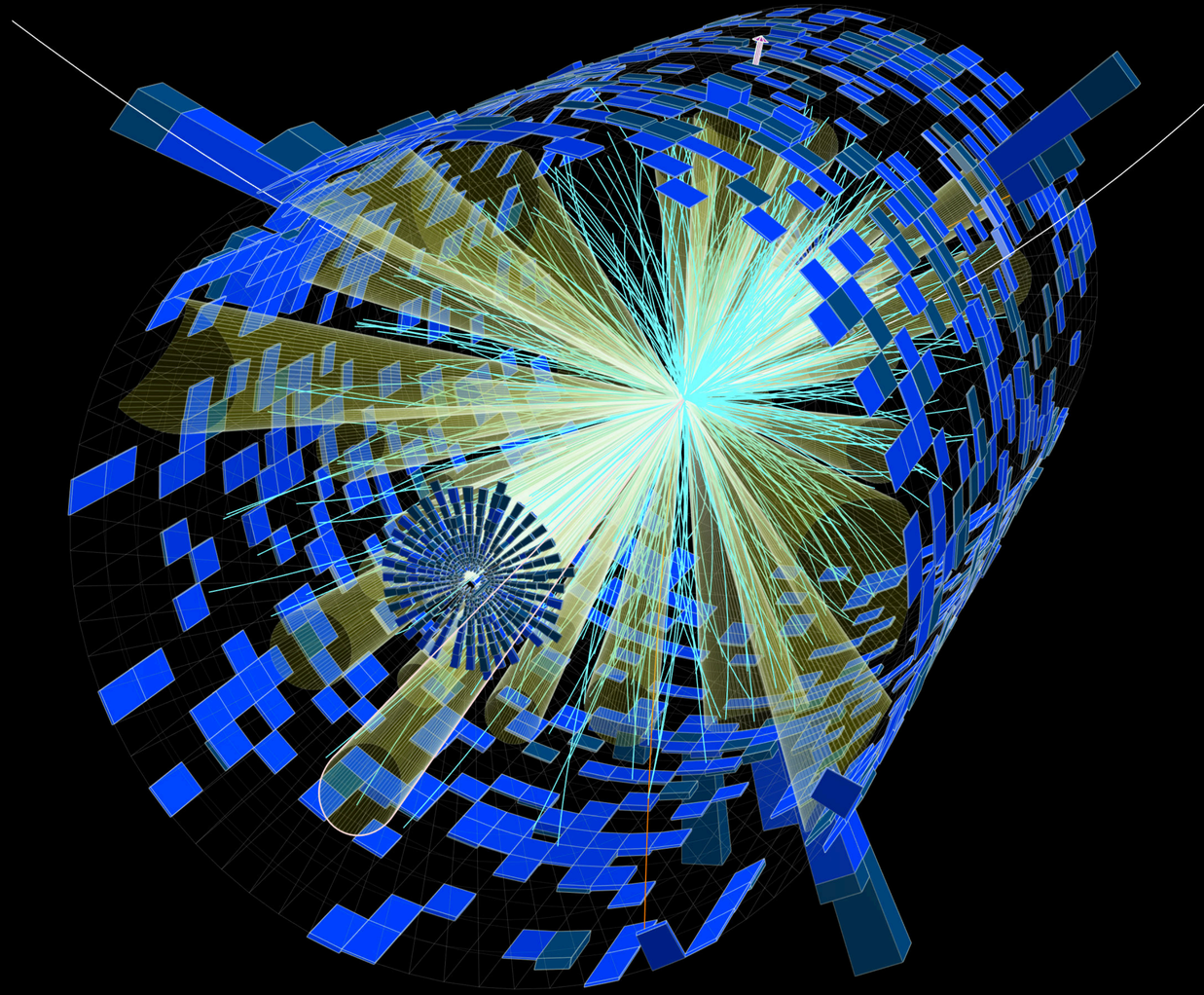
SciPy 2024

Particles, People, and Pull Requests

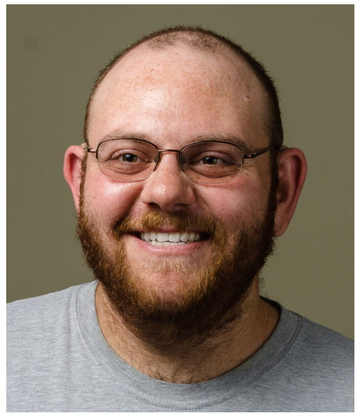


@KyleCranmer
University of Wisconsin-Madison
Data Science Institute, Physics, Statistics, Computer Science

Acknowledgements

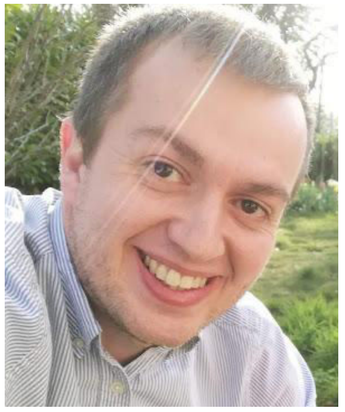

Matthew Feickert


Lukas Heinrich


Giordon Stark


Alex Held


Oksana Shadura


Gilles Louppe


Johann Brehmer


Jim Pivarski


Ianna Osborne


Henry Schreiner


Jonas Eschle


Gordon Watts


Vangelis Kourlitis


Patrick Rieck

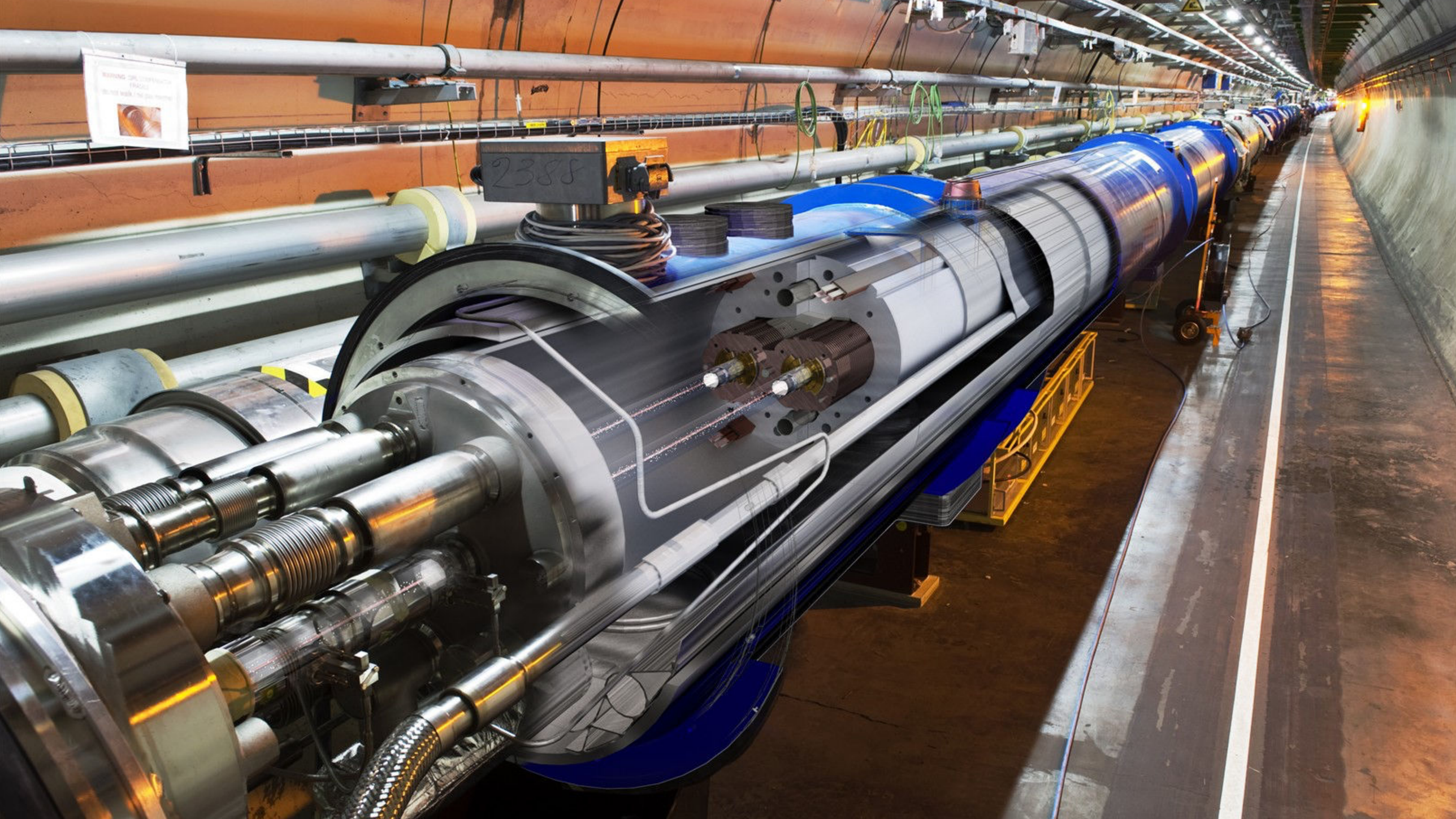

Zubair Bhatti


Irina Espejo

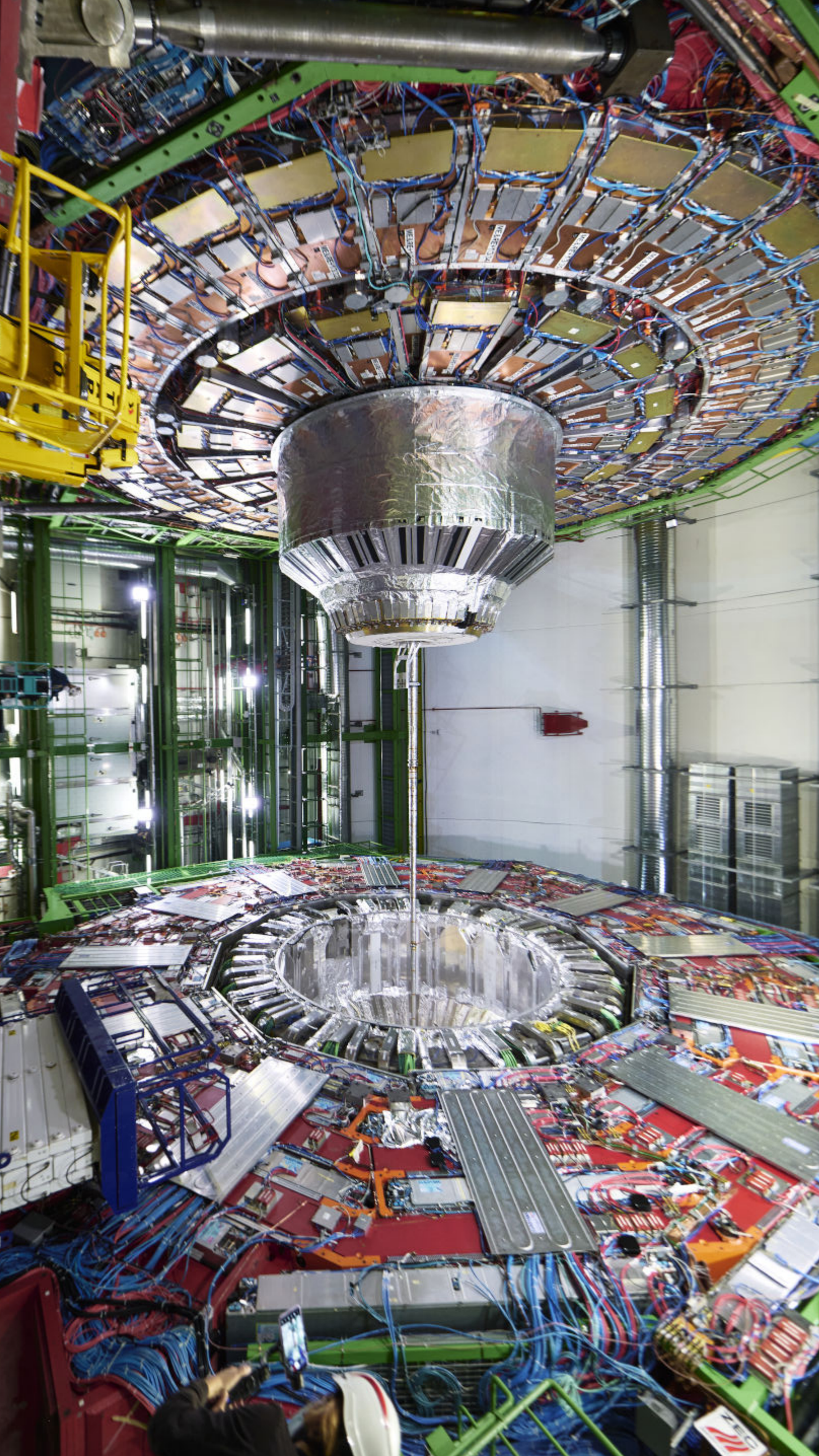
This work was supported by the National Science Foundation awards ACI-1450310, ACI-1450319, ACI-1450323, ACI-1450377, OAC-1836650, OAC-2226378, OAC-2226379, OAC-2226380, and PHY-2323298.

High Energy Physics
&
The Large Hadron Collider


LHCb
ATLAS
CERN Meyrin
CERN Prévessin
PS ø 0.2 km
SPS 7 km
SUISSE
FRANCE
CMS
ALICE
LHC 27 km


2388

LAYHER




CMS
Compact Muon Solenoid
1992
2017


ATLAS
OVERVIEW
WEEK
New York University, USA
27 June to 1 July 2016

Big Numbers

40 quadrillion

collisions

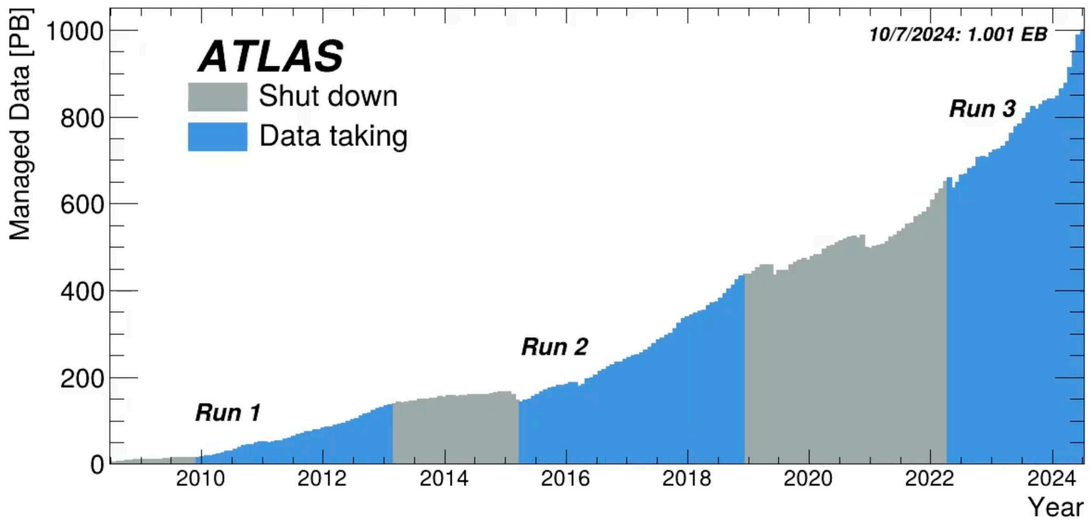
The number of collisions recorded by the four main experiments at the LHC is close to 40 quadrillion, or, as physicists say it, 395 “inverse femtobarns.” (Each inverse femtobarn corresponds to about 100 trillion collisions.)

7.5 billion

Worldwide LHC Computing Grid requests

Physicists need a huge amount of computing power to do their research—much more than a standard laptop can support. Every day several thousand physicists submit a total of about 2 million “jobs” to the WLCG. Each “job” is an important brick in the growing body of scientific work.

1 exabyte

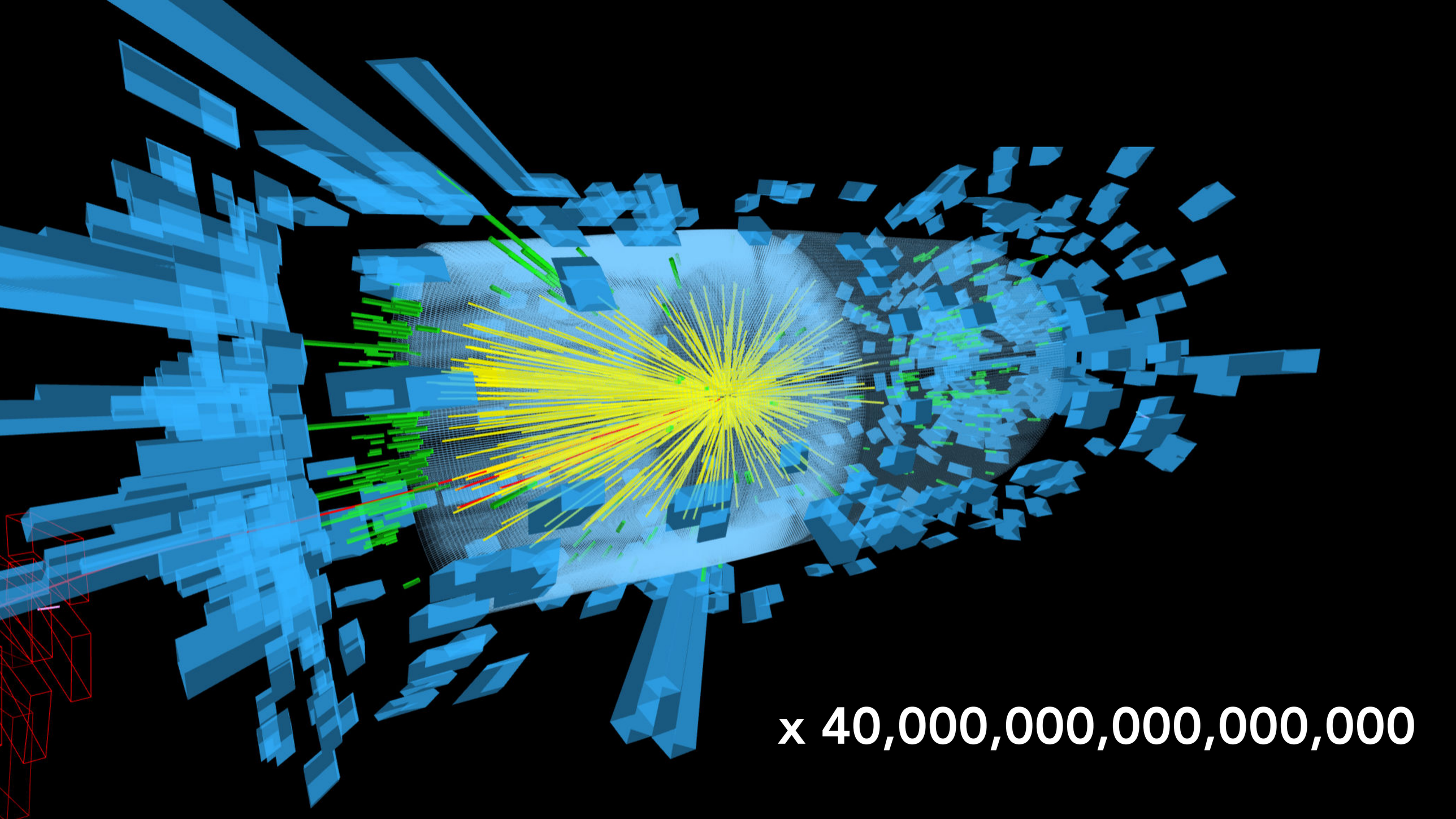


2,725

scientific papers

Every week the number of scientific papers that LHC scientists have published steadily increases as they comb through the data to study rare phenomena and search for new physics. This includes work by thousands of graduate students on their way to earning their PhDs.

https://www.symmetrymagazine.org/article/10-years-of-lhc-physics-in-numbers


x 40,000,000,000,000,000

40,000,000,000,000,000

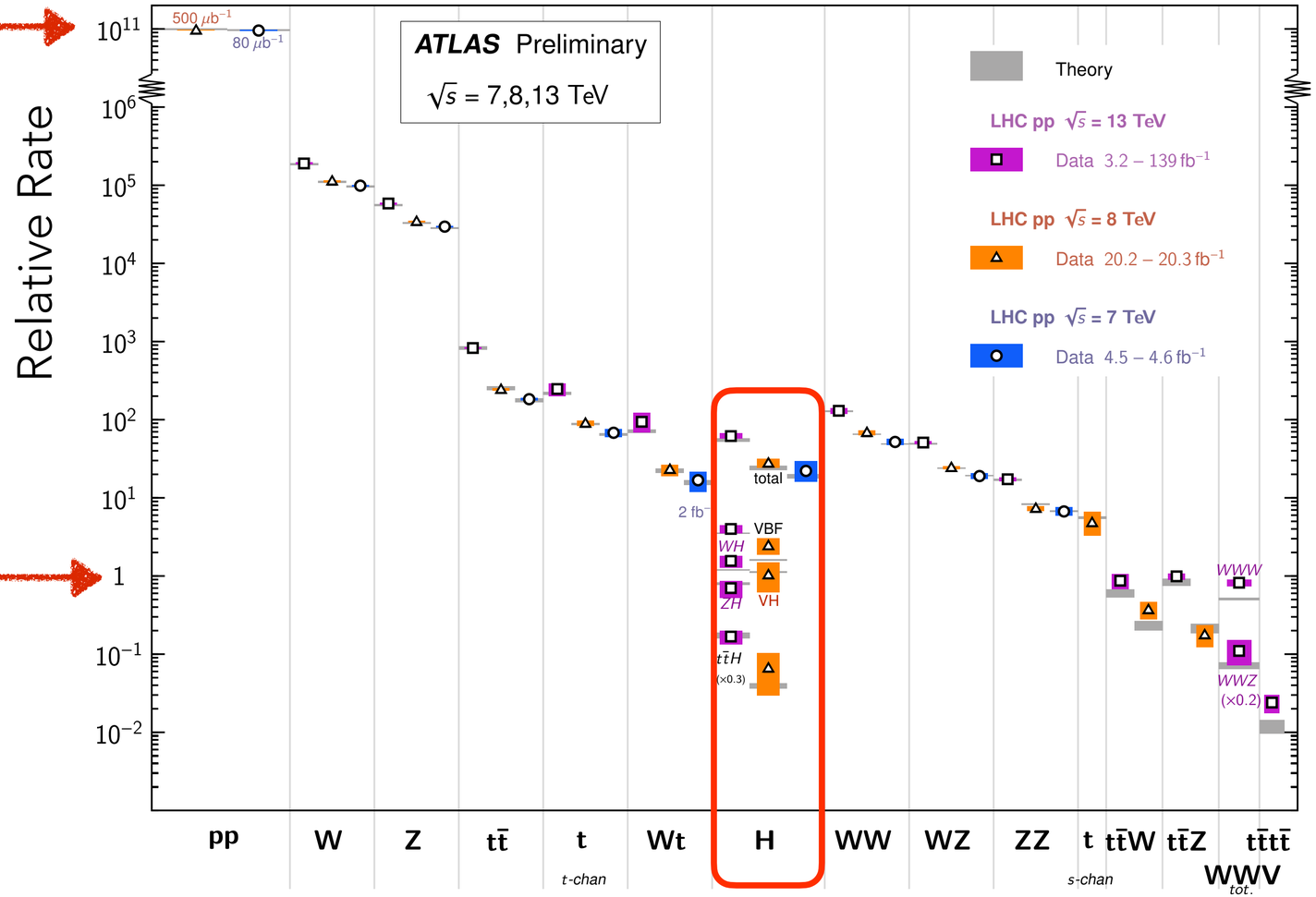
Needle in haystack

LHC collides bunches of protons at 40 MHz. Detector has ~100M sensors

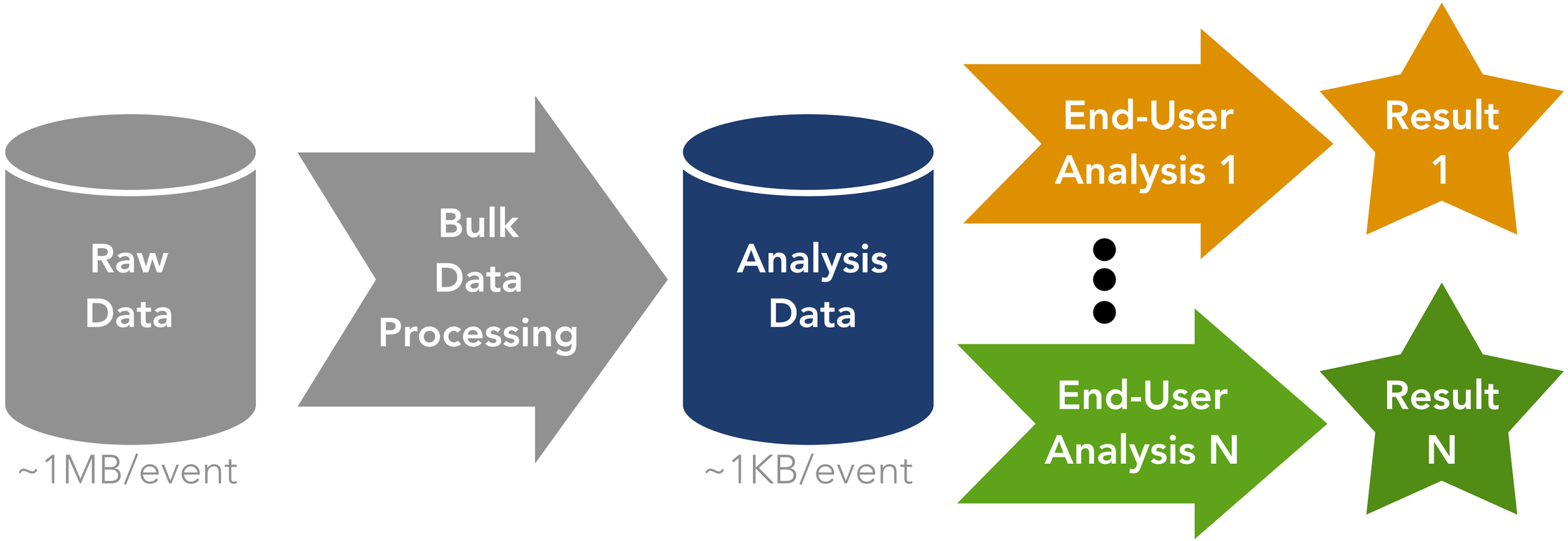
- Signals of interest are rare, with S/B ~10^{-10}

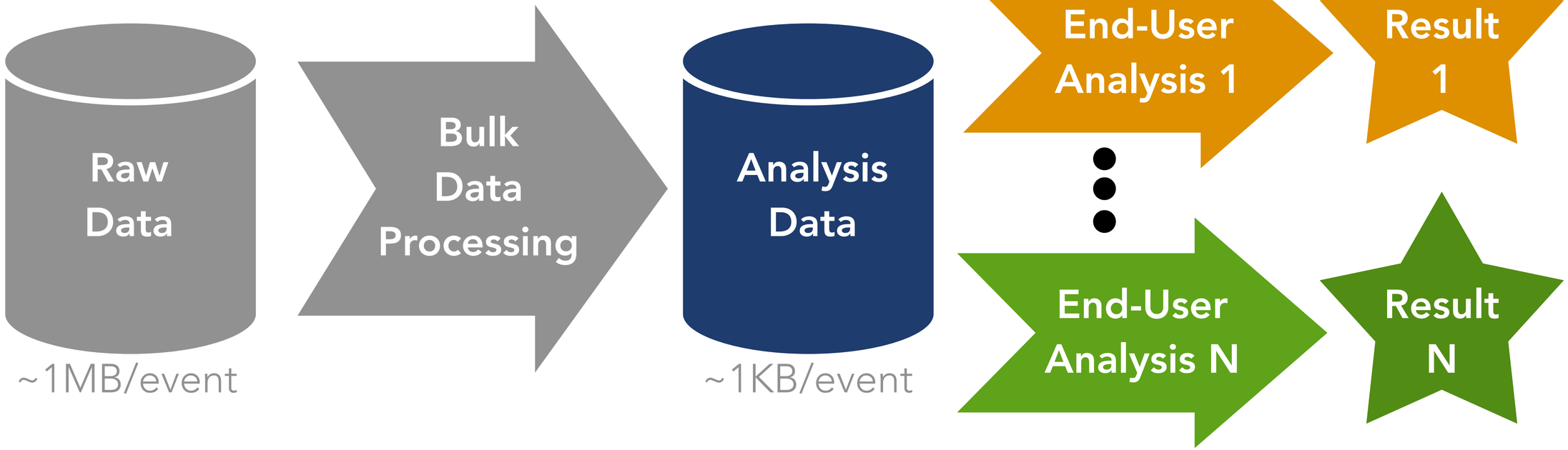
Background rate

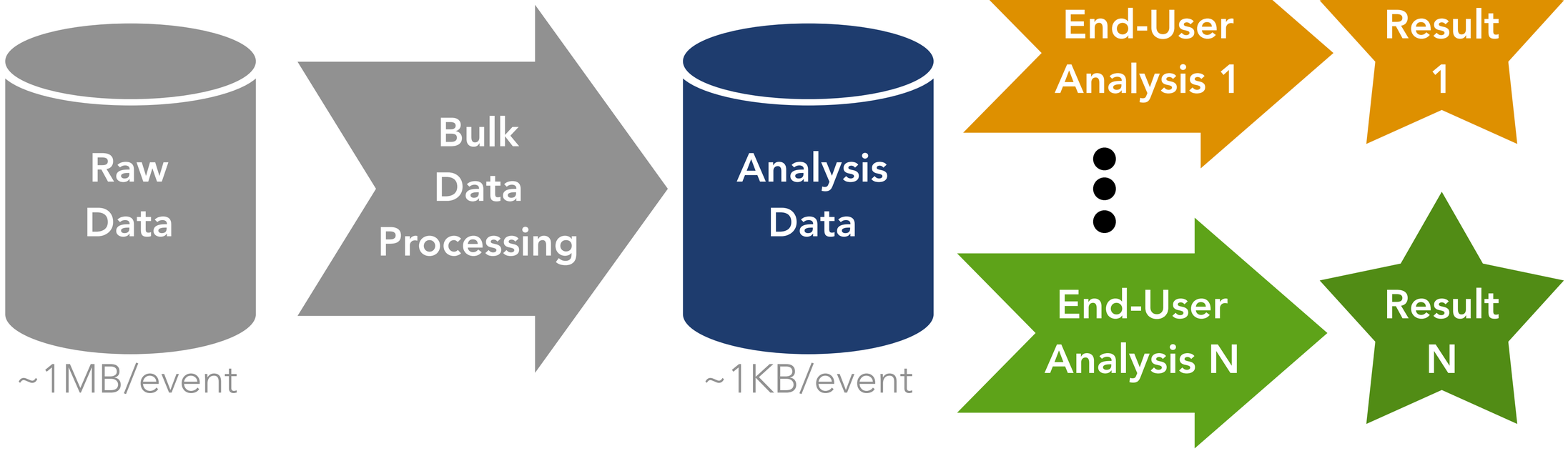
Signal Rate

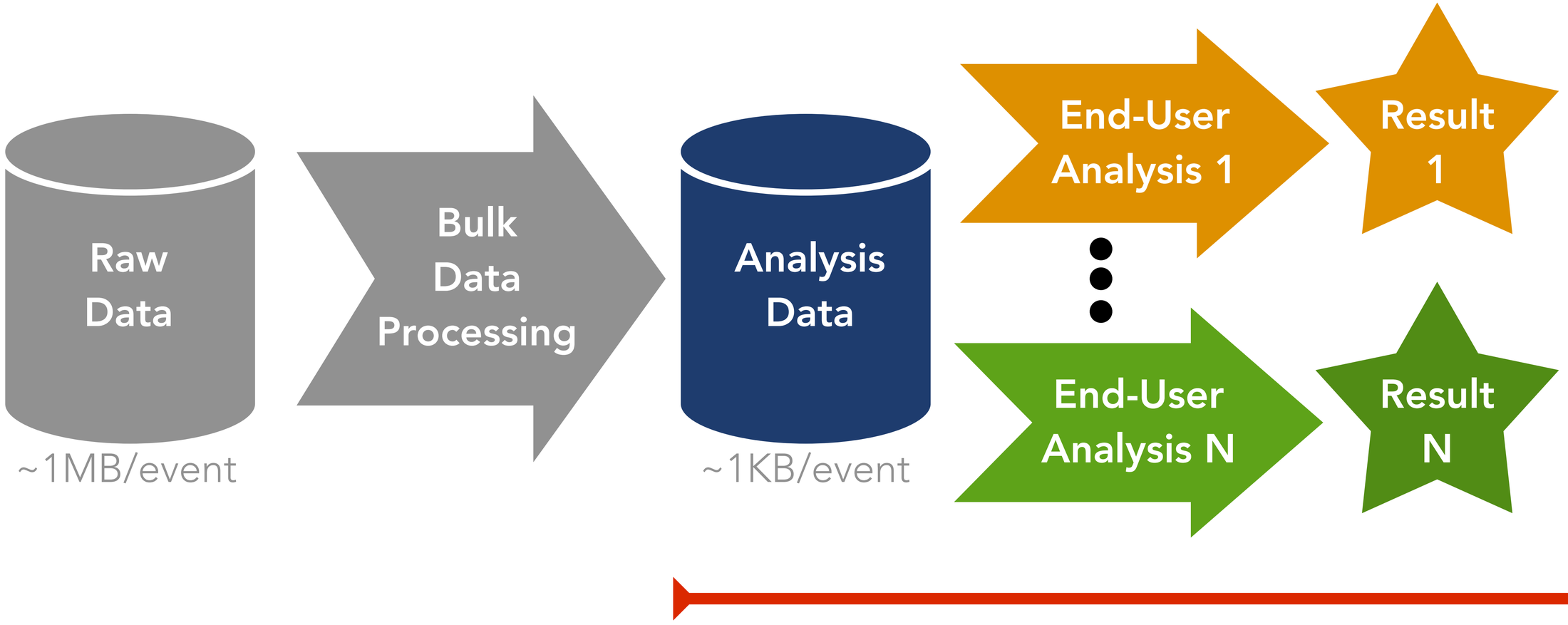



Raw
Data
Bulk
Data
Processing
Analysis
Data
~1MB/event
~1KB/event


Raw
Data
~1MB/event
Bulk
Data
Processing
Analysis
Data
~1KB/event
End-User
Analysis 1
Result
1
End-User
Analysis N
Result
N


Raw
Data
~1MB/event
Bulk
Data
Processing
Analysis
Data
~1KB/event
End-User
Analysis 1
Result
1
End-User
Analysis N
Result
N


Raw
Data
~1MB/event
Bulk
Data
Processing
Analysis
Data
~1KB/event
End-User
Analysis 1
Result
1
End-User
Analysis N
Result
N

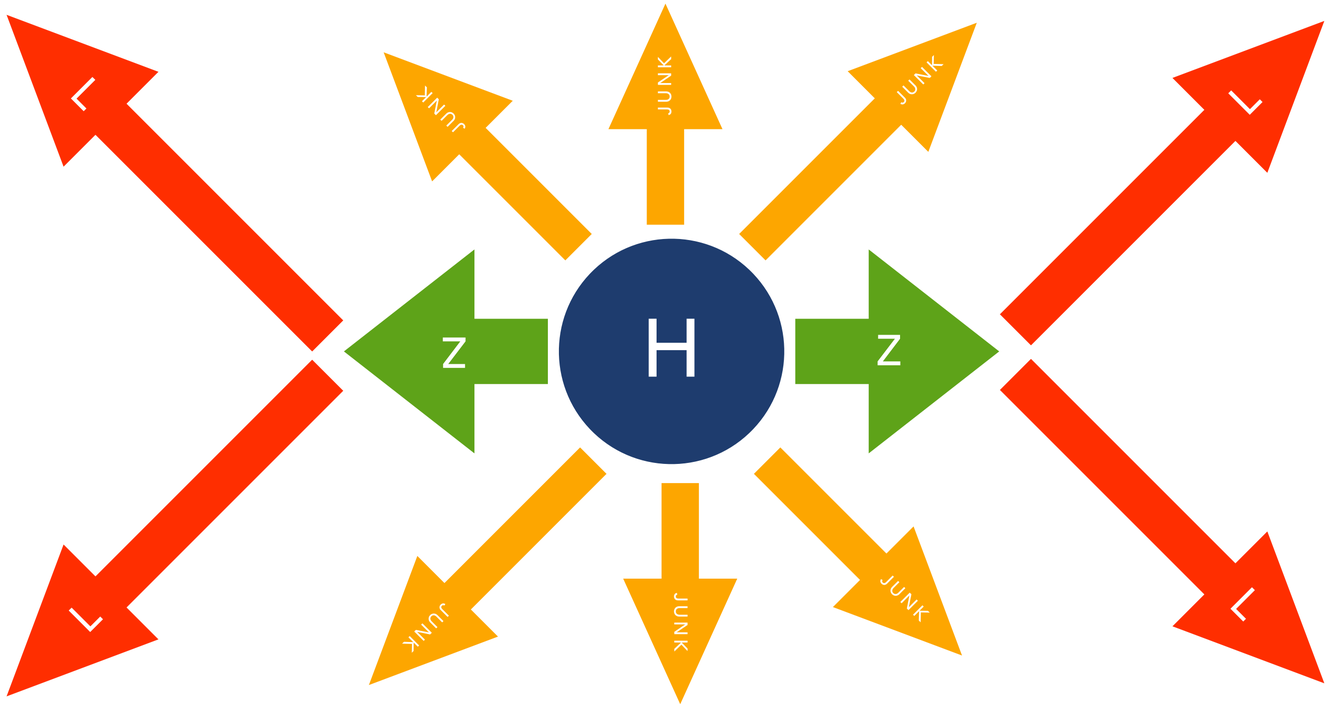

Raw
Data
~1MB/event
Bulk
Data
Processing
Analysis
Data
~1KB/event
End-User
Analysis 1
Result
1
End-User
Analysis N
Result
N
Focus of this talk

Example hypothesis

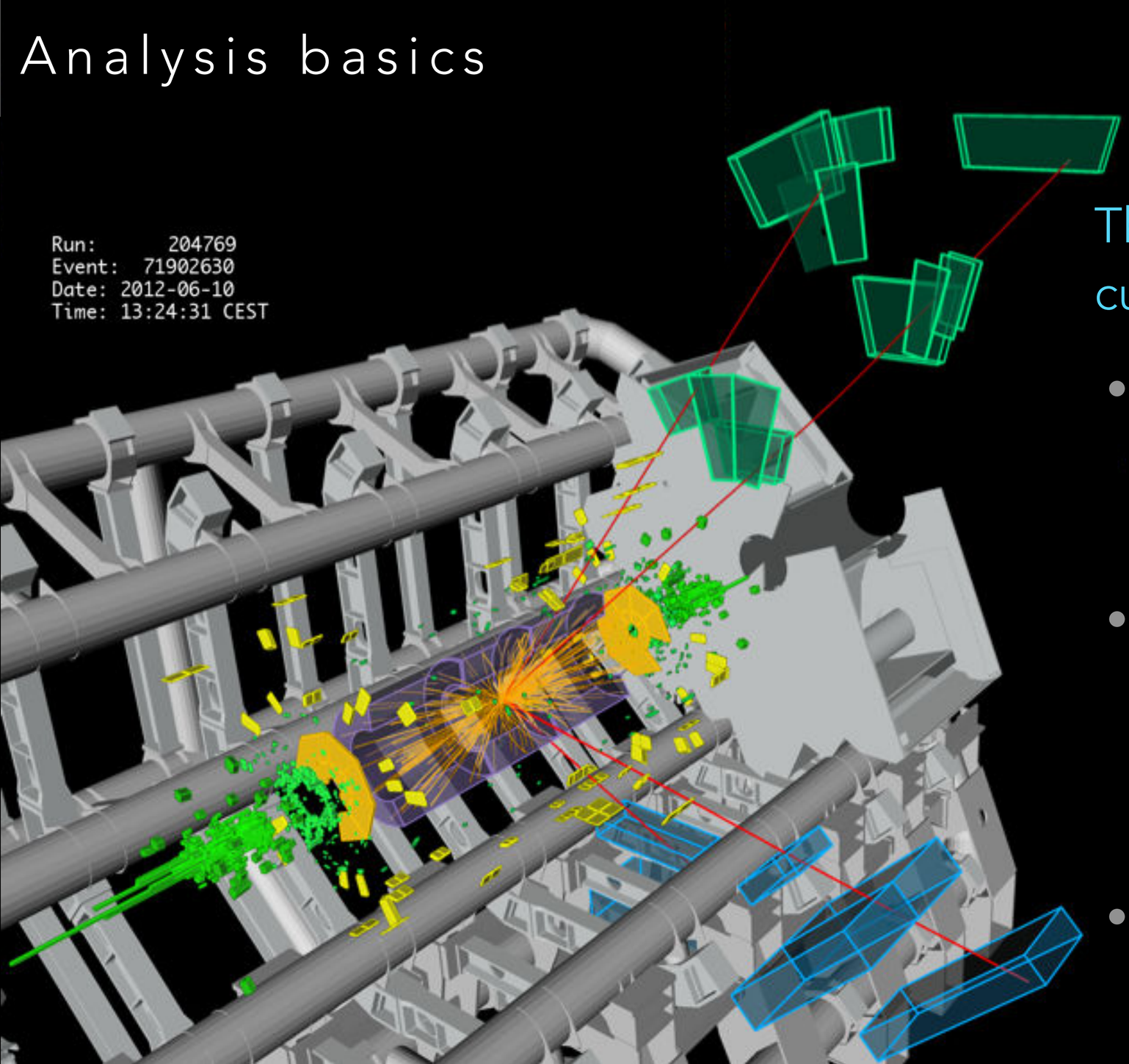
What if we produce (along with some other **junk**) an unstable particle **H**

- that decays into two other unstable particles **Z**,
- that each decay into two particles **L** that we detect in our sensors

What would be a good strategy to search for **H**?



Analysis basics



This “end-user analysis” is highly customized for a particular goal

- First **filter the data** to find collisions that are useful for testing a particular hypothesis
- Experts design *features / observables / summary statistics* that distinguish signal from background
- Perform statistical analysis

Expert feature engineering

Don’t believe the media:

$$E \neq mc^2$$

What Einstein really said:

$$E^2 = (mc^2)^2 + (|\vec{p}|c)^2$$

Every physics student knows energy and momentum are conserved

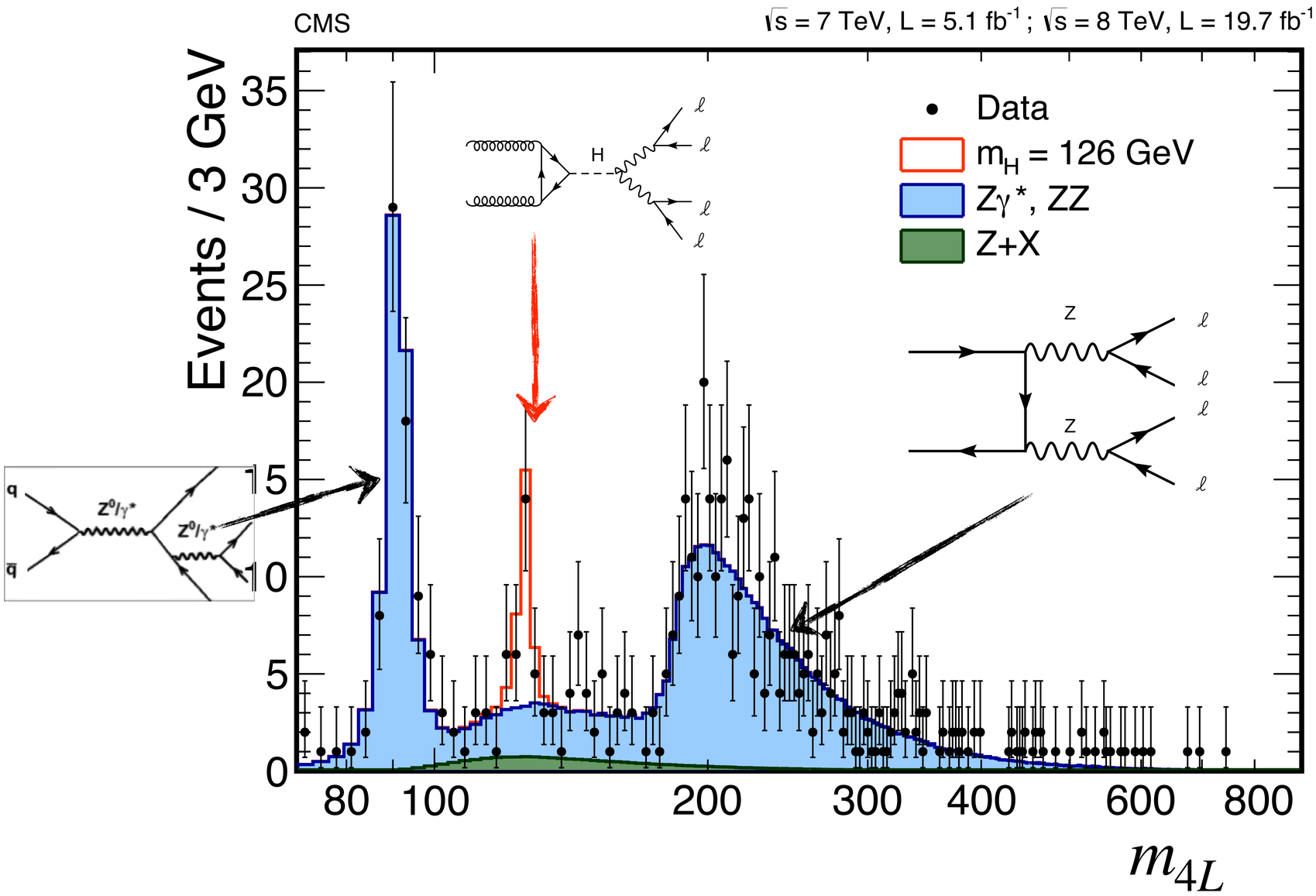
$$E_{\text{Higgs}} = E_{\text{before}} = E_{\text{after}} = \sum_i E_i$$

$$\vec{p}_{\text{Higgs}} = \vec{p}_{\text{before}} = \vec{p}_{\text{after}} = \sum_i \vec{p}_i$$

Thus, for our hypothesis, we expect a peak in this feature / observable:

$$m_{4L} = \sqrt{E_{\text{after}}^2/c^4 - |\vec{p}_{\text{after}}|^2/c^2}$$

A signal emerges

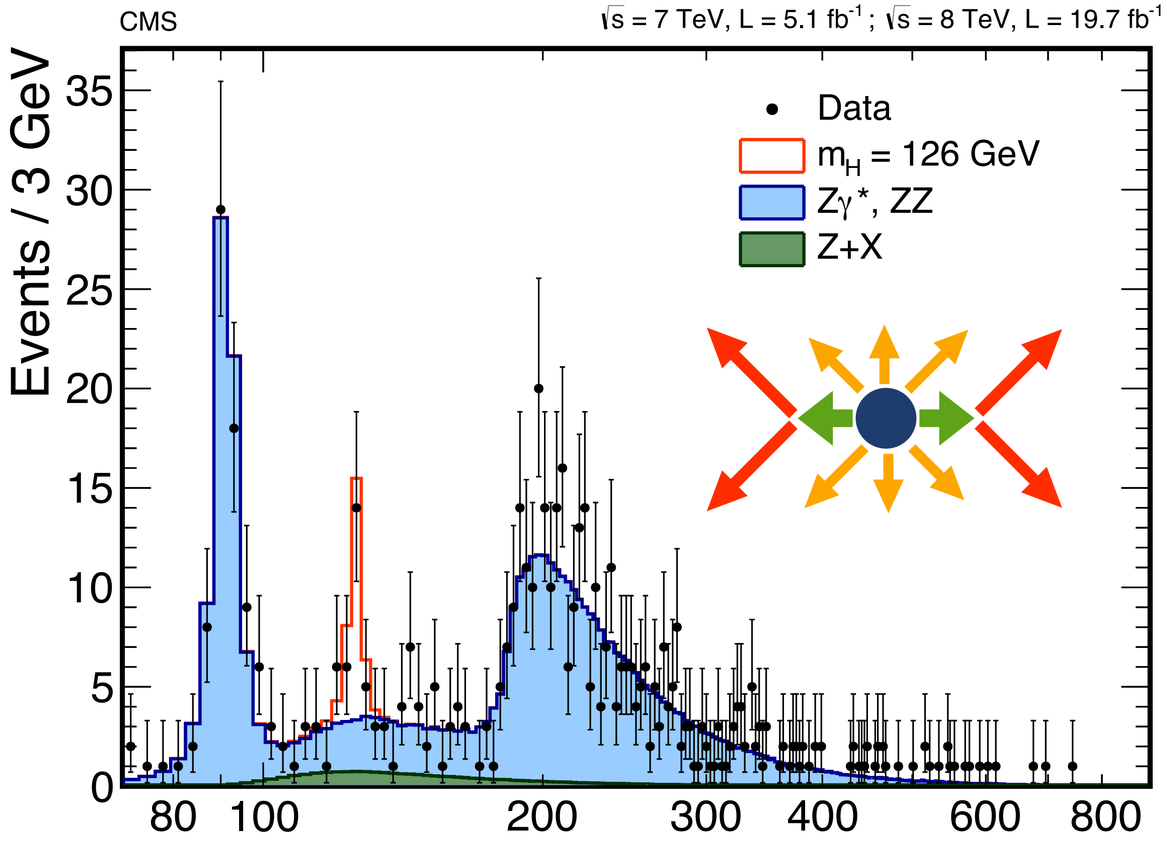


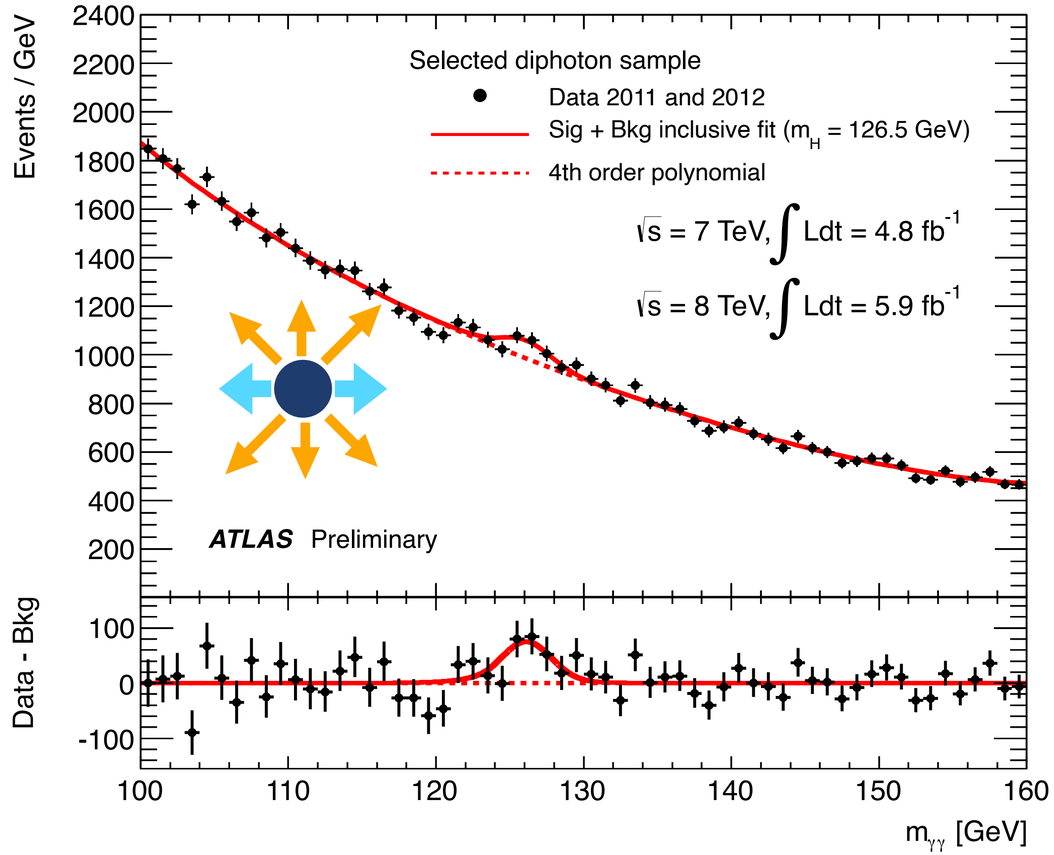
Collaborative Statistical Modeling

The power of an underlying theory

Hypotheses are connected to an underlying theory, which allows us to relate their predictions across multiple signatures in the data

- Unstable particle **H** can decay to **Z** or **G**

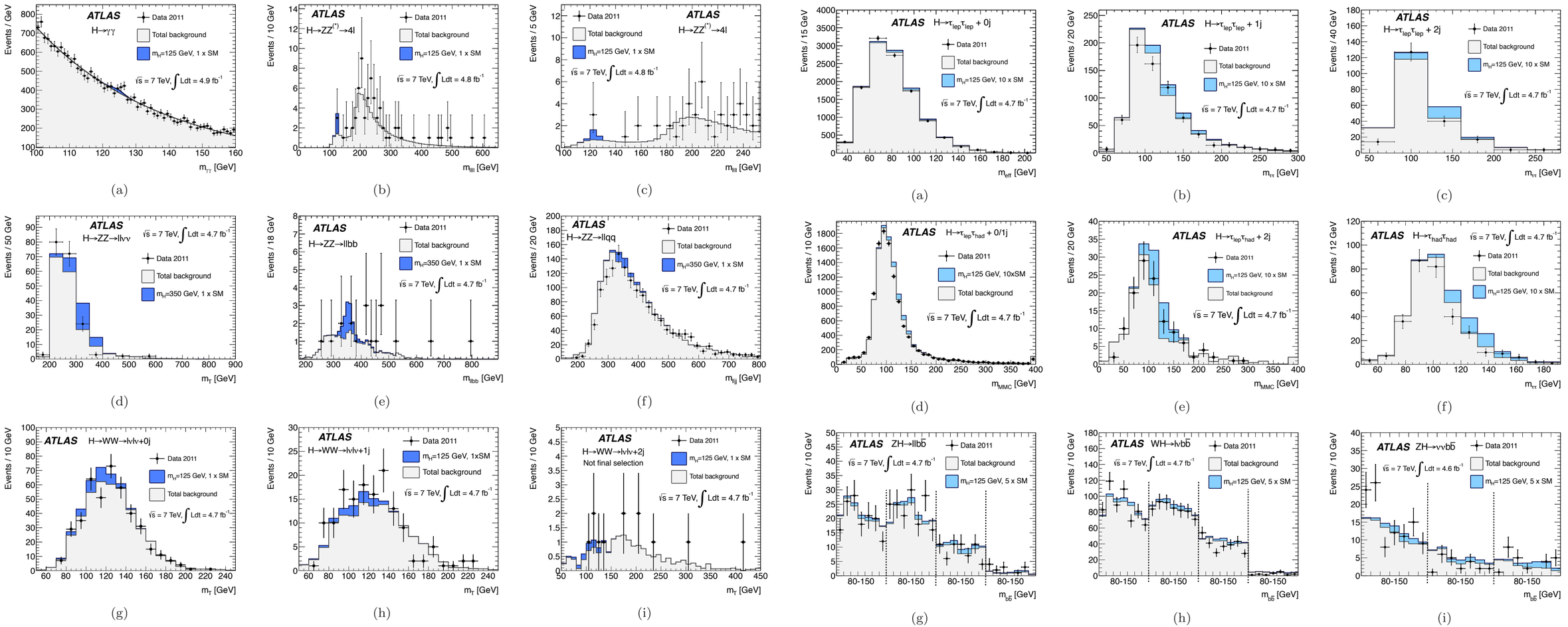




We want to combine the evidence statistically

The traditional approach was very top-down and centralized

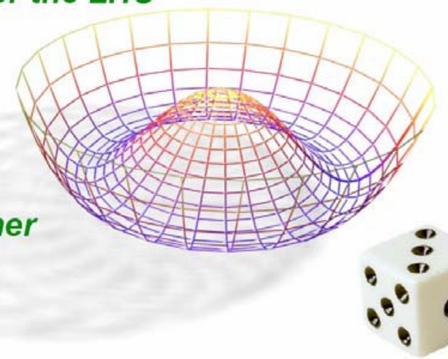
- Tightly coupled, required a lot of coordination, moved slowly, very rigid.



Preparing for the LHC in 2007

BROOKHAVEN NATIONAL LABORATORY

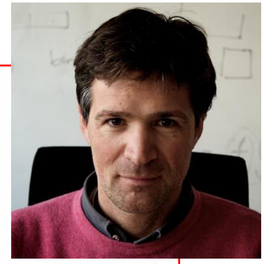
Progress, Challenges, and Future of Statistics for the LHC



Kyle Cranmer (BNL)

Kyle Cranmer (BNL) PhyStat 2007, CERN, June 26, 2007 1

ROOT Statistical Software



Lorenzo Moneta (CERN, PH-SFT)

on behalf of the ROOT Math Work Package

(R. Brun, A. Kreshuk, E. Offermann + many others contributors)

ROOT

An Object-Oriented Data Analysis Framework

Statistics software for the LHC

Wouter Verkerke

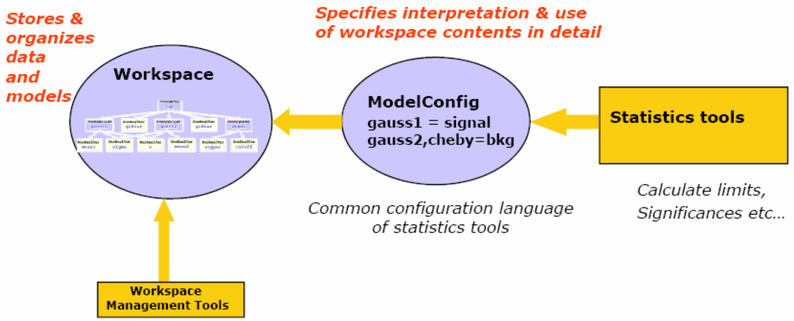
The Workspace as publication

- Now have functional `RooWorkspace` class that can contain
  - Probability density functions and its components
  - (Multiple) Datasets
  - Supporting interpretation information (`RooModelConfig`)
  - Can be stored in file with regular ROOT persistence

- **Ultimate publication of analysis…**
  - Full likelihood available for Bayesian analysis
  - Probability density function available for Frequentist analysis
  - Information can be easily extracted, combined etc…
  - Common format for sharing, combining of various physics results

Framework design & RooFit adaptations

- Have had more meetings last 3 months to review RooFit lessons from BaBar
  - Kyle, Amir Farbin (ex-Babar), Frank Wrinklmeyer (ex-Babar), WV
  - Design for *WorkSpace* and *ModelConfig* concept in RooFit to interface with statistics tools



ROOT Team Meeting

Friday 18 Apr 2008, 11:00 → 13:00 Europe/Zurich

32/1-A24 (CERN)

11:00 → 11:05 **News**

-New schema Evolution document will be ready early next week (Lukasz)
-Bertrand poster
-Document by Axel/Philippe on new multi-threaded CINT C++ design
-Progress report next meeting
-David/Lorenzo,
-Axel
-Olivier: progress with TGraph restructure

11:05 → 11:30 **SVN restructuring status**

-still to be done
-/doc and ReleaNotes.html in each top level dir
-$ROOTSYS/doc pointing to the top chapters and release notes

-misc/table
/minicern (still to be ported on Windows and Solaris)
removal of references to cernlib/shift from configure

11:30 → 11:50 **5.19/04 dev release scheduled for May 7**

-expect new roofit/roostats package from Kyle Cranmer and Wouter Verkerke
-material in branches must be moved before middle of next week
-developments by Ilka/Roj will be introduced after the release
-fixes to get "make static" and the code checker working again
-THtml must be modified to support the new dir structure (urgent)

Bob Cousins Slide: http://indico.cern.ch/conferenceDisplay.py?confId=100458

Object Serialization in C++ with ROOT

Obvious idea now, but an important shift was moving from **centralized model**:

- scripts that create everything from scratch at run-time

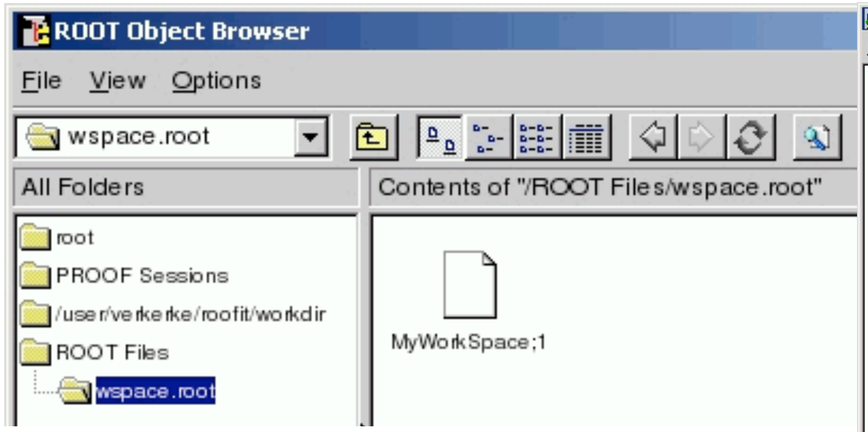
to **distributed model**:

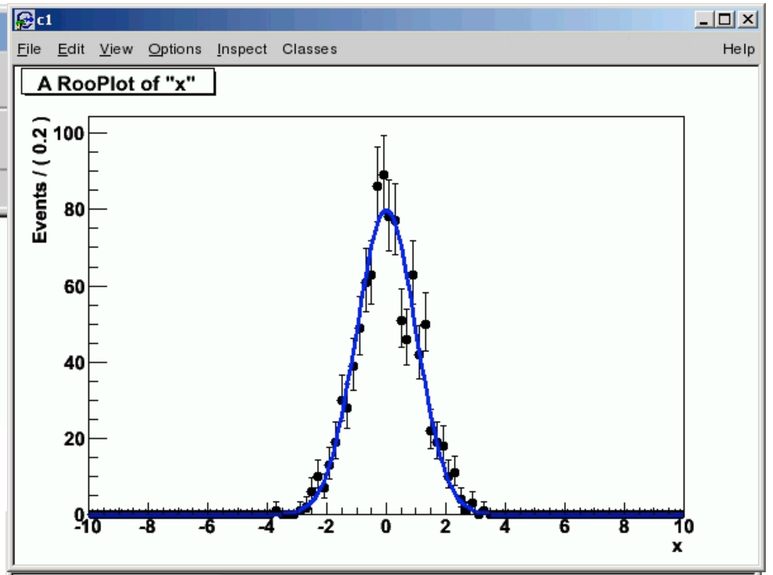
- Different groups **sharing digital artifacts**

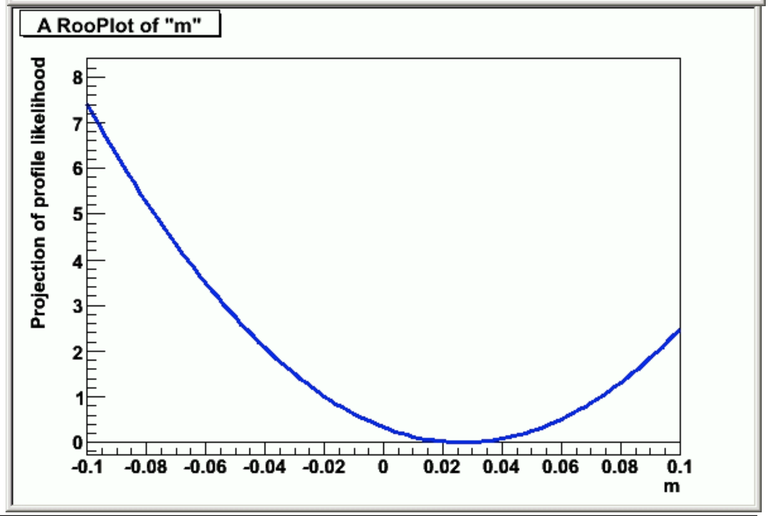
At the time serializing complicated data structures composed of C++ objects was non-trivial

Example of Digital Publishing

BROOKHAVEN NATIONAL LABORATORY







Wouter recently demonstrated the ability to save the function $L(x|\theta_r, \theta_s)$ in a Root file with minimal data necessary to reproduce likelihood function.

Can also evaluate integrals over x necessary for Neyman construction!

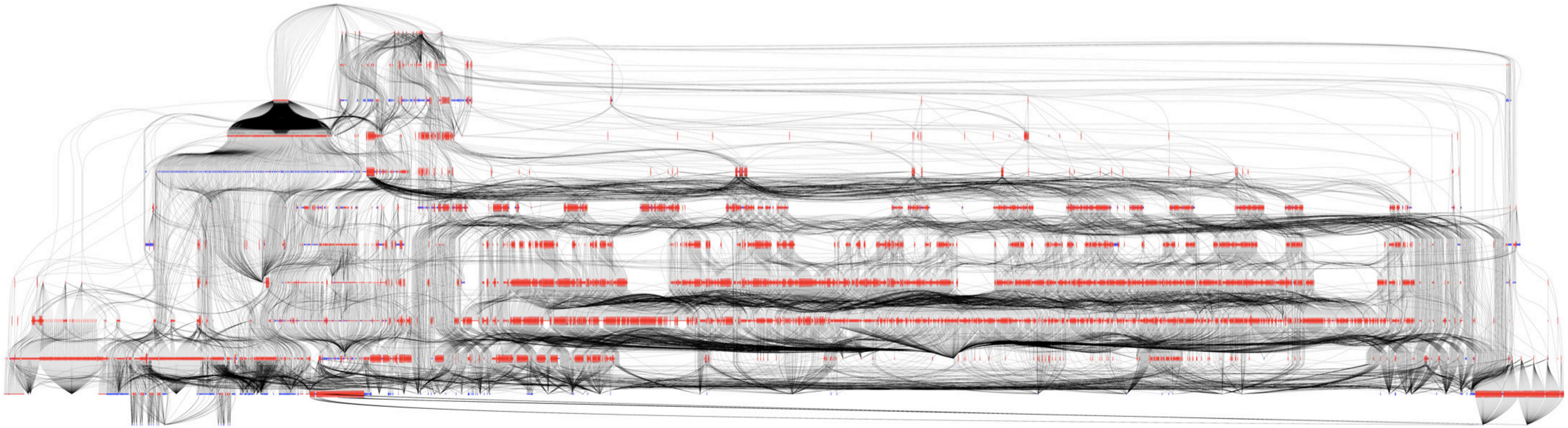
Need this for combinations, we should publish them to some repository!

Combining evidence from multiple datasets

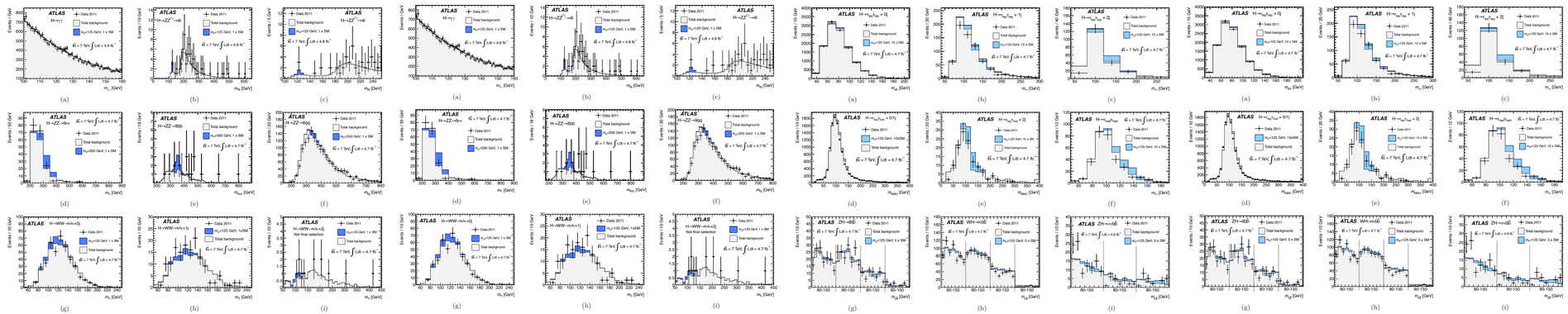
We developed a formalism, **a specification**, and a software implementation that allowed us **to combine** (statistically) **evidence** from **multiple datasets**

- Provides enormous flexibility to **remix** the evidence (different map theories to same signature)

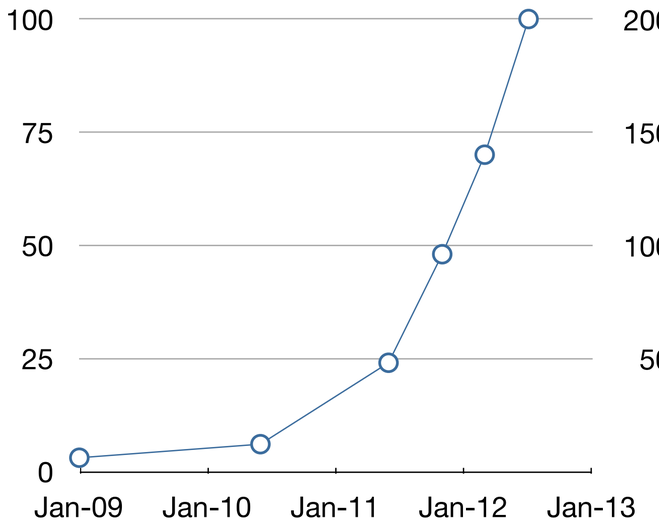
$$\mathbf{f}_{\text{tot}}(\mathcal{D}_{\text{sim}}, \mathcal{G}|\boldsymbol{\alpha}) = \prod_{c \in \text{channels}} \left[\text{Pois}(n_c|\nu_c(\boldsymbol{\alpha})) \prod_{e=1}^{n_c} f_c(x_{ce}|\boldsymbol{\alpha}) \right] \cdot \prod_{p \in \mathbb{S}} f_p(a_p|\alpha_p)$$



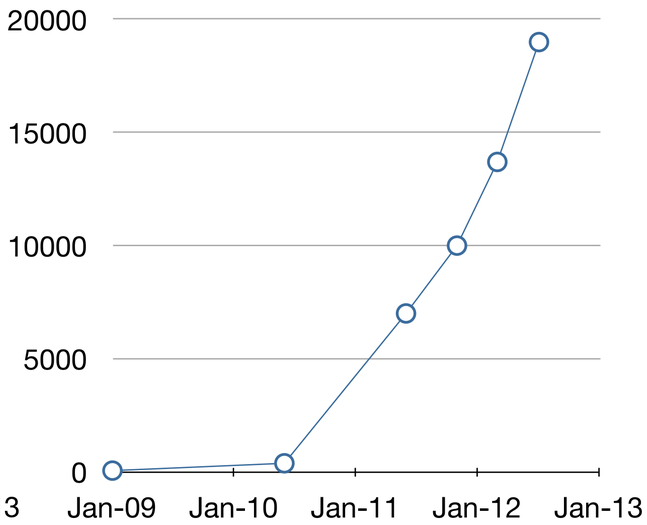
Collaborative Statistical Modeling



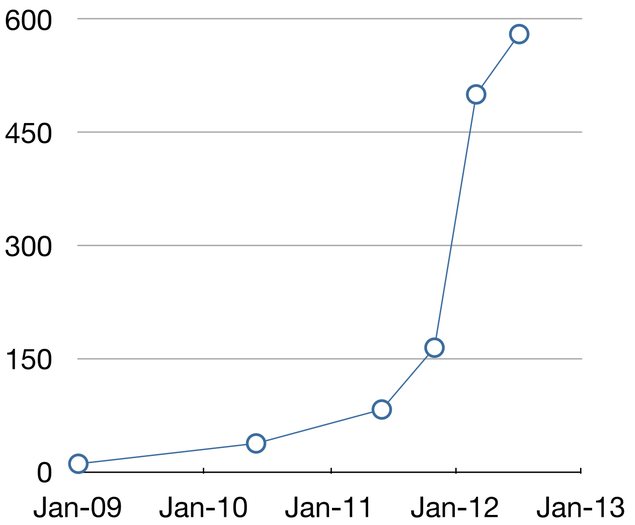
Number of Datasets Combined

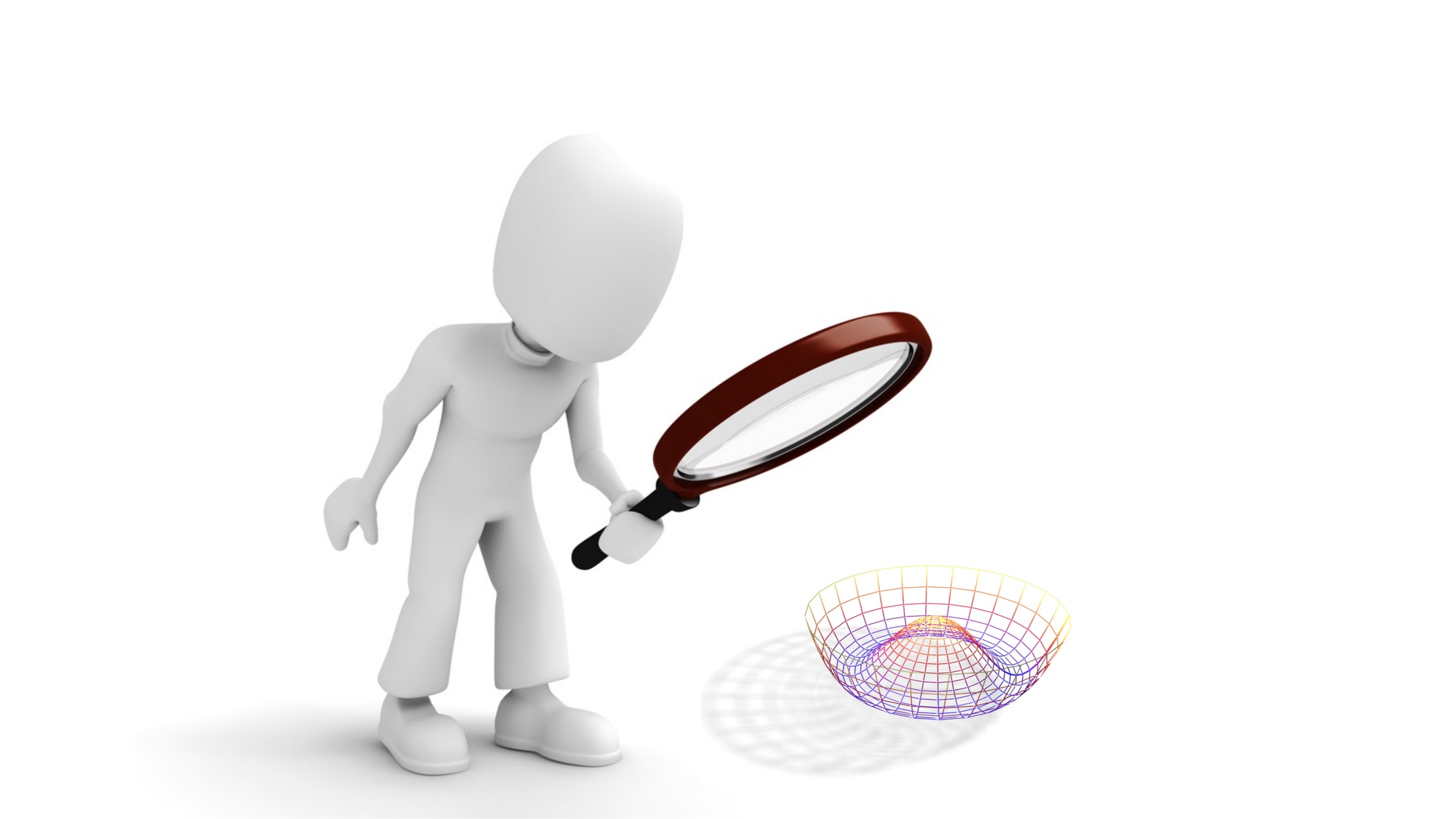


Number of Model Components



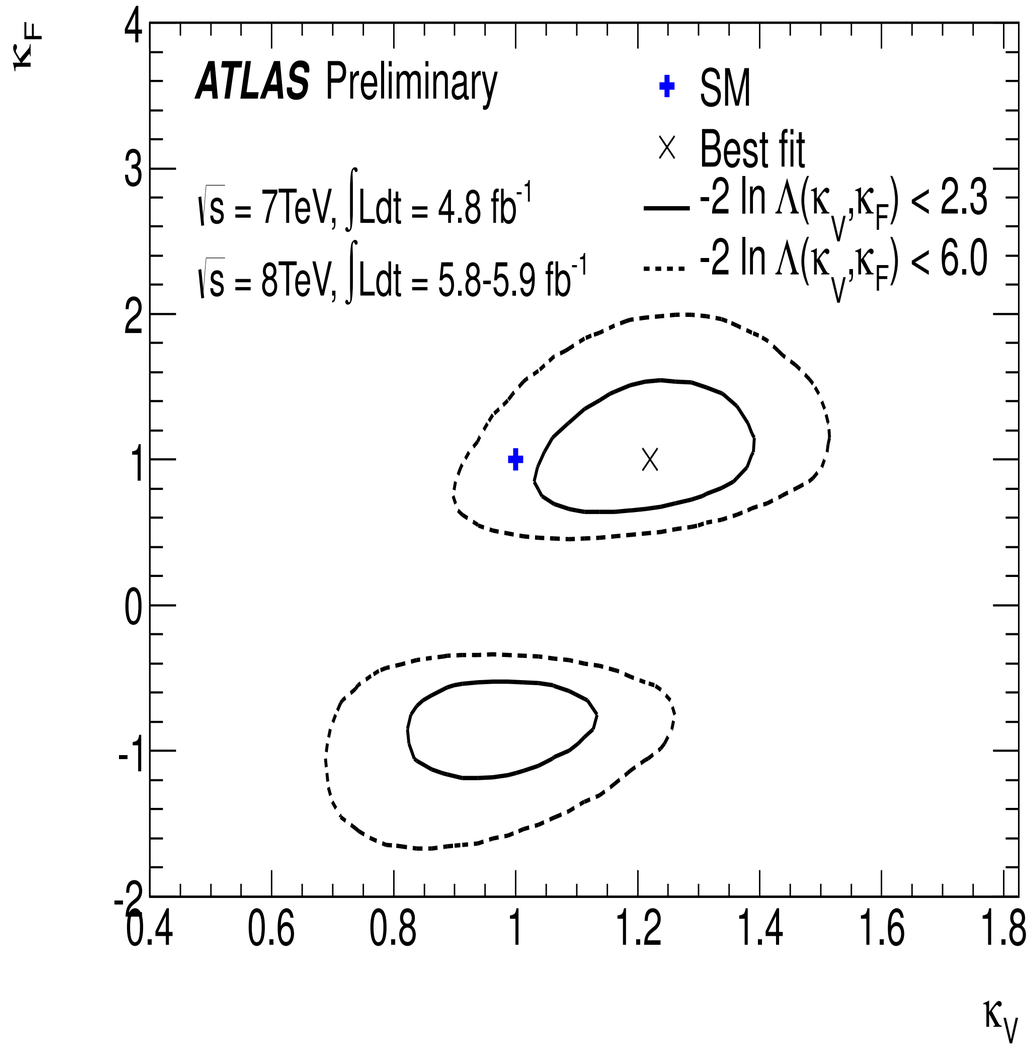
Number of Parameters in Likelihood





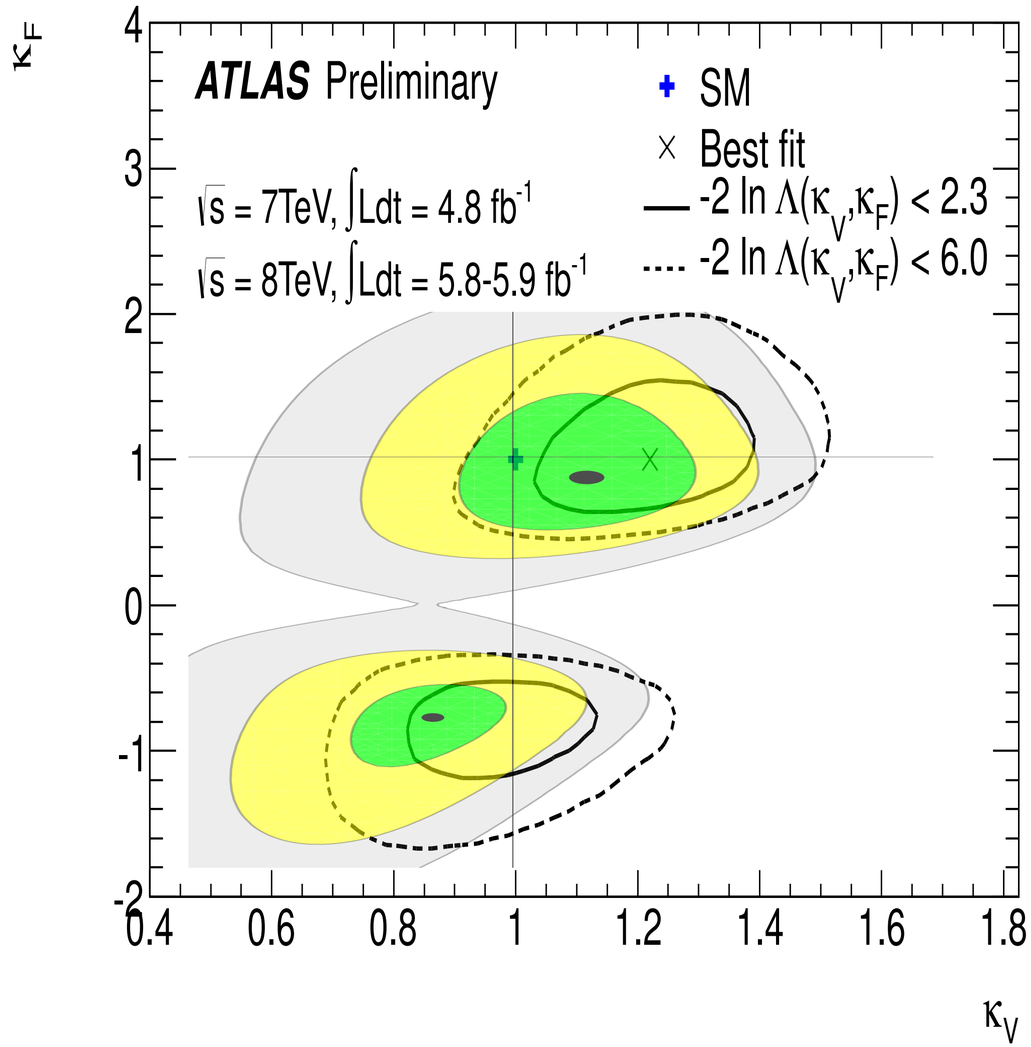
Reproducibility problem

Not possible for others to reproduce results from paper.



Reproducibility problem

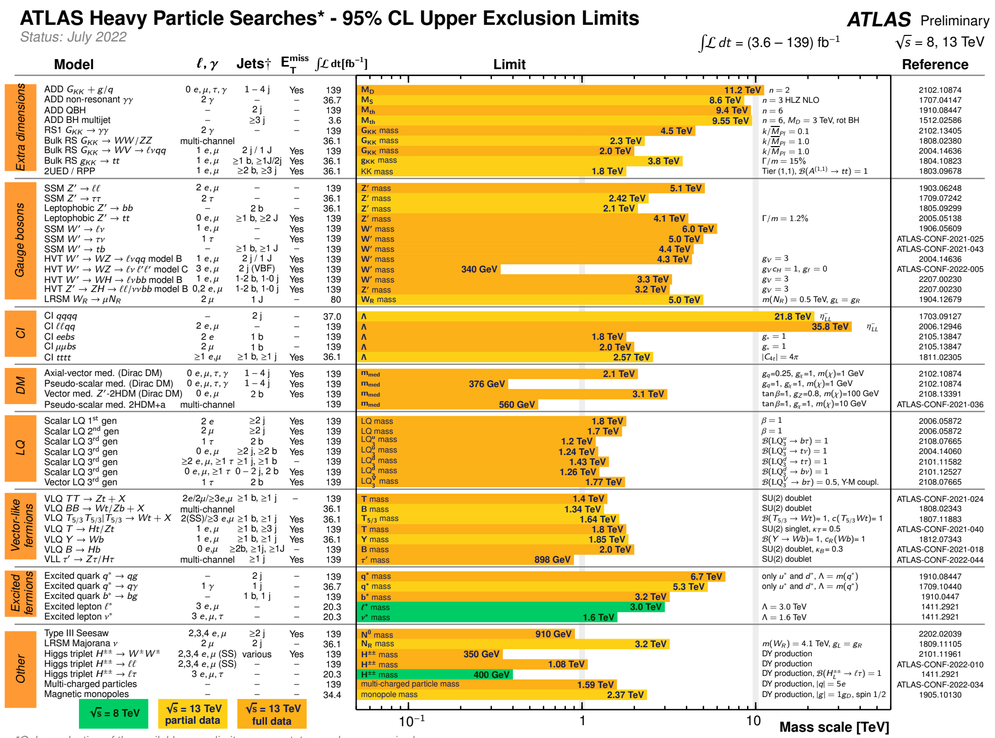
Not possible for others to reproduce results from paper.

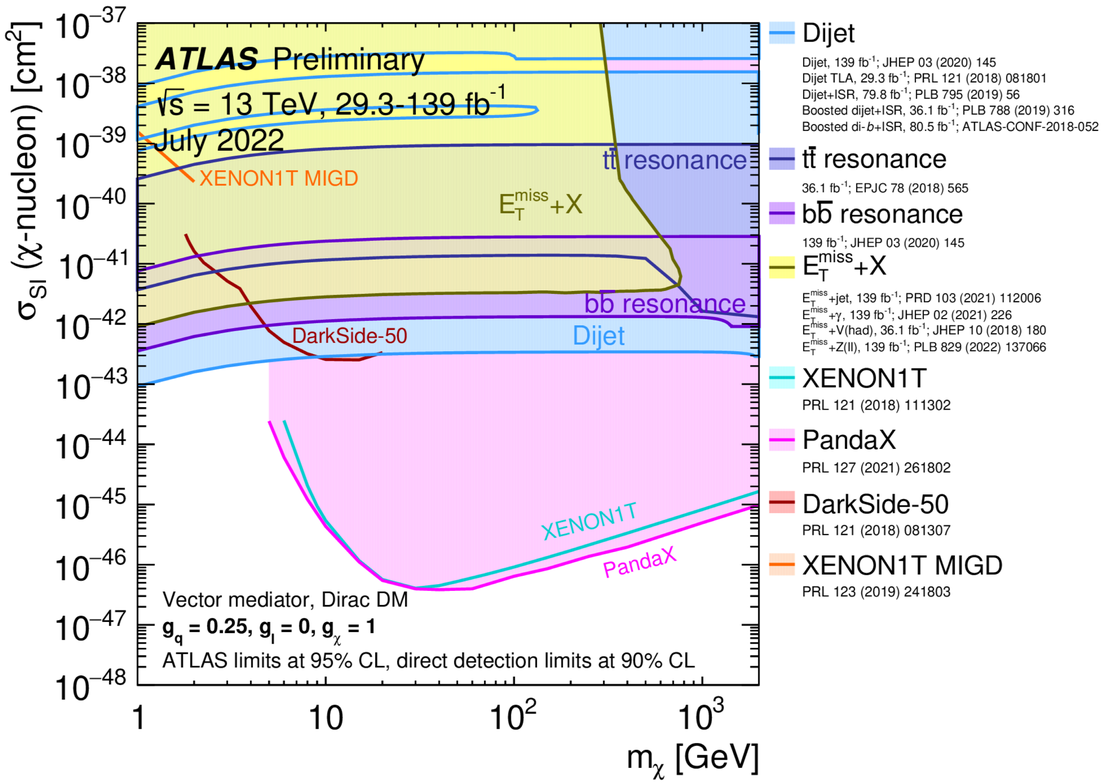


Null results and reinterpretation

Usually, we **do not see** the signature that would be evidence for a hypothesis

- **Is that because we weren't sensitive to it, or because it just wasn't there?**
- Huge unmet need for reinterpretation — many questions went unanswered





Reinterpretation as a Service

recast

$$\mathcal{L}_{SM} = \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^{a}_{\mu\nu}G^{\mu\nu}_{a}}_{\text{kinetic energies and self-interactions of the gauge bosons}}$$

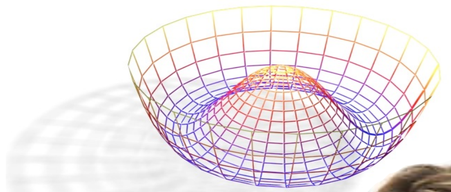
$$+ \underbrace{\bar{L}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})L + \bar{R}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g'YB_{\mu})R}_{\text{kinetic energies and electroweak interactions of fermions}}$$

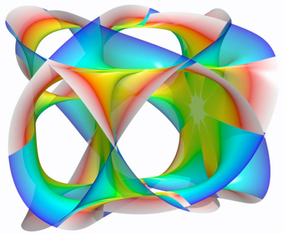
$$+ \underbrace{\frac{1}{2}\left|(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})\,\phi\right|^{2} - V(\phi)}_{W^{\pm},Z,\gamma,\text{and Higgs masses and couplings}}$$

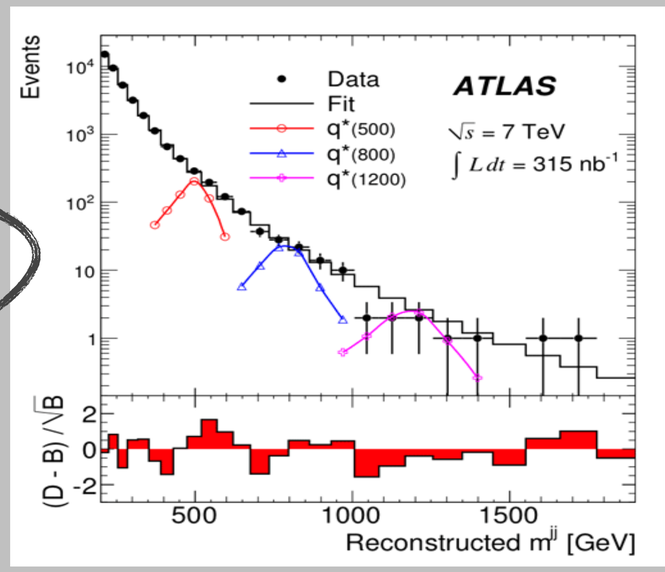
$$+ \underbrace{g''(\bar{q}\gamma^{\mu}T_{a}q)\,G^{a}_{\mu}}_{\text{interactions between quarks and gluons}} + \underbrace{(G_1\bar{L}\phi R + G_2\bar{L}\phi_c R + h.c.)}_{\text{fermion masses and couplings to Higgs}}$$

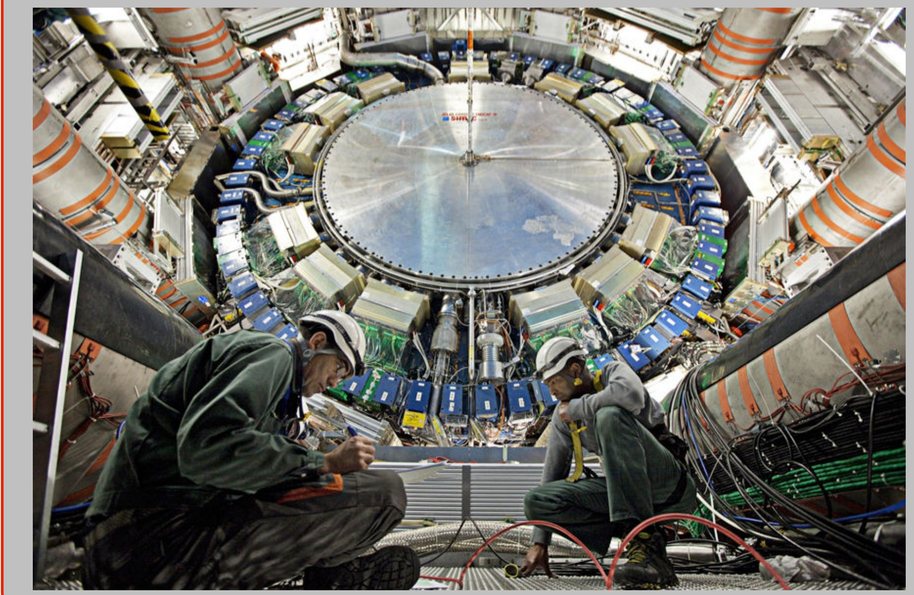
Q

A









https://doi.org/10.1007/JHEP04%282011%29038

Where were we 10 years ago?

Incentives

- **Bulk data processing** was respected as essential, given resources and incentives, well managed with dedicated professionals
- **End-user analysis**, in contrast, was considered ‘the fun part’ and not given resources or incentives
  - Neglected the need for high-quality, specialized tools used by many
  - Wild West. Lack of harmonization led to inefficient process
  - Analysis code rarely preserved, results were difficult to reproduce or reinterpret

ROOT (a general-purpose C++ based framework for analysis) was used by almost everyone

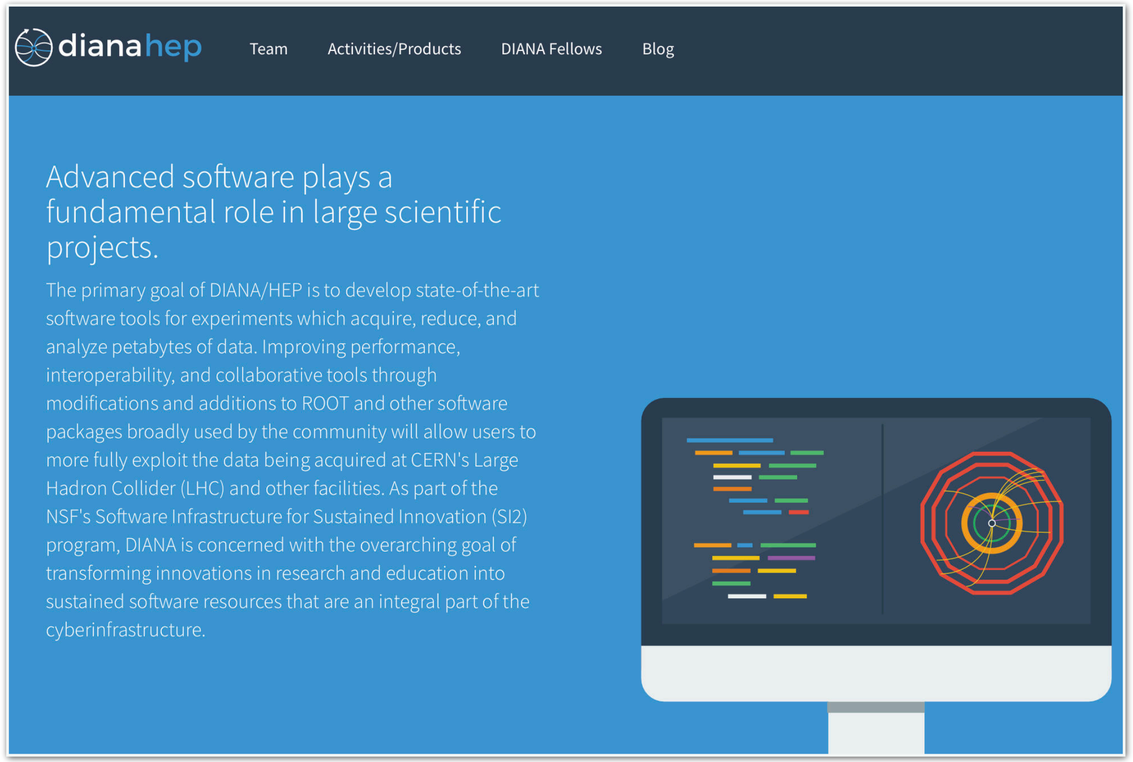
- Open source, but had a small development team with few contributions from community

Overall our toolkit was not aligned with industry, data science, or other areas of science

- Better alignment ⇒ better leverage of tools in the data science ecosystem, better competitiveness for PhD’s entering job market, better appreciation of HEP's value to society

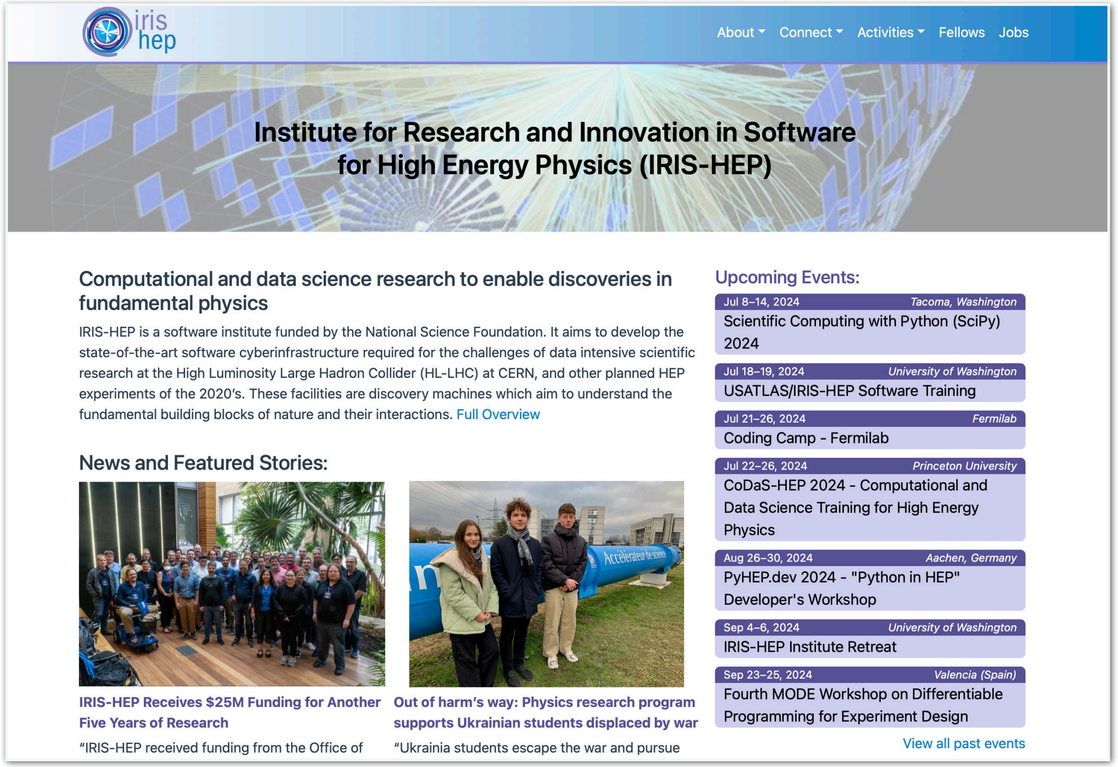
A new approach

In 2014 we were awarded NSF funding for a new software effort aimed at an upgrade of the LHC with an approach that was more aligned with python & data science



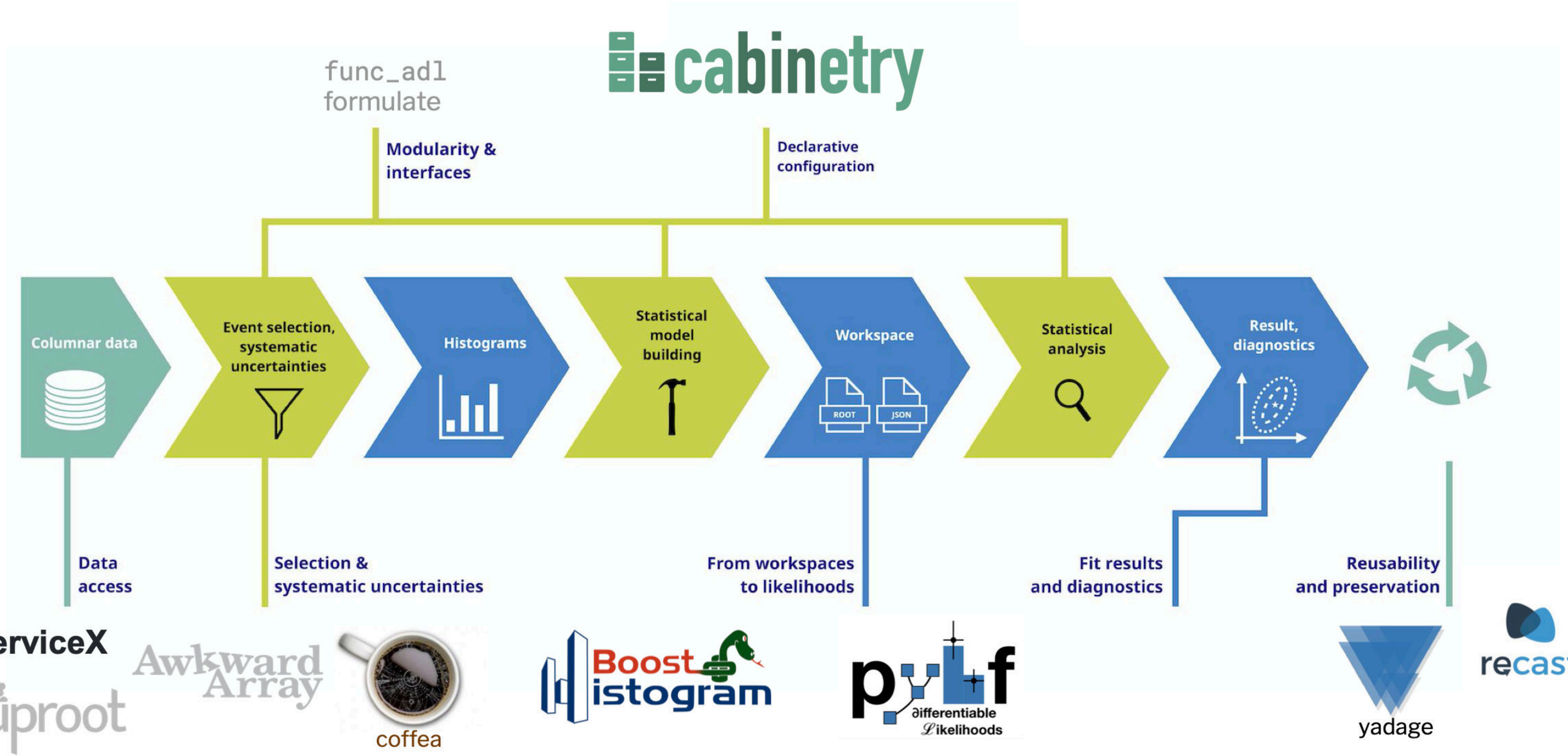
2014 - 2018

Bogdan Mihaila

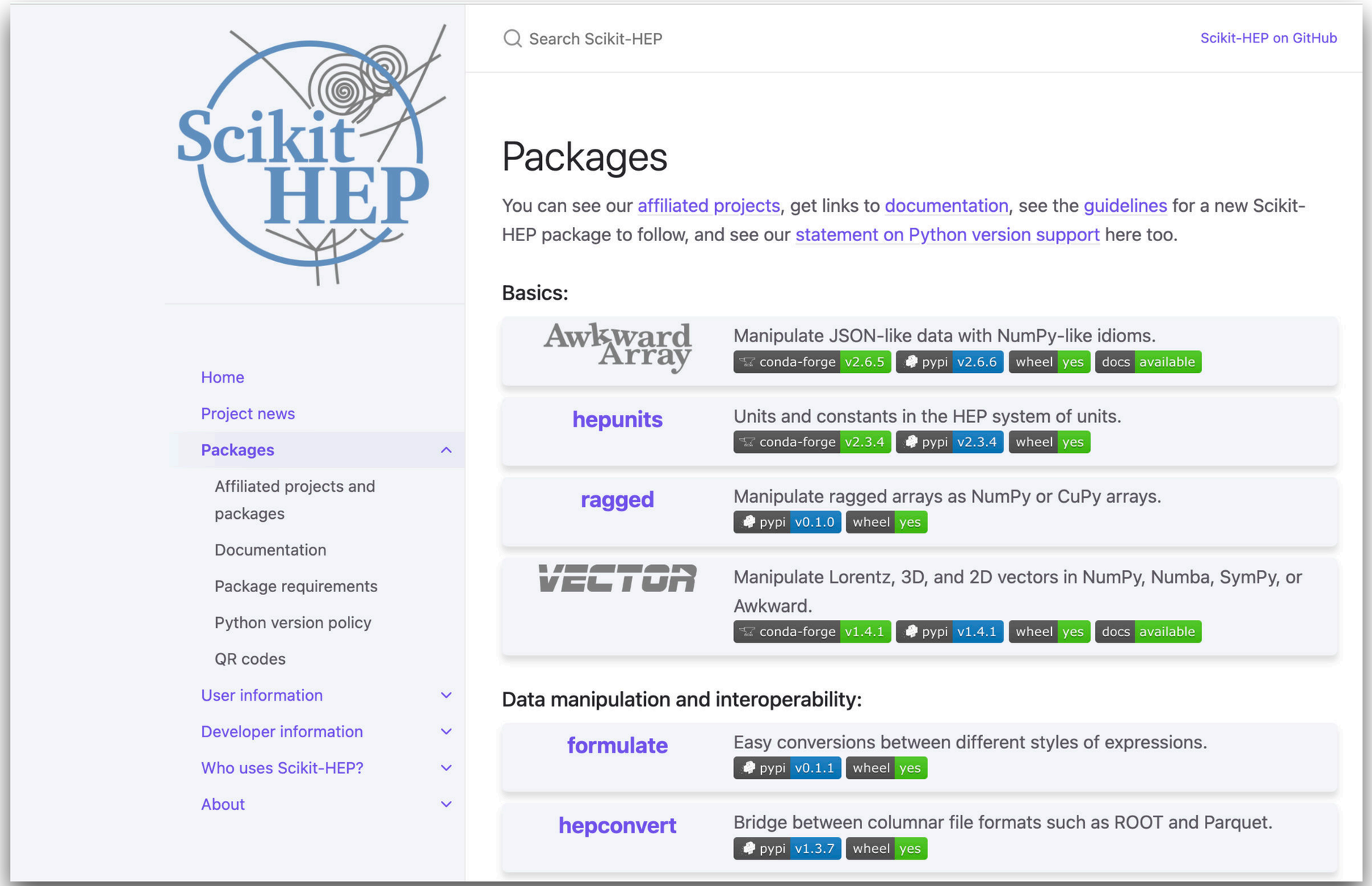


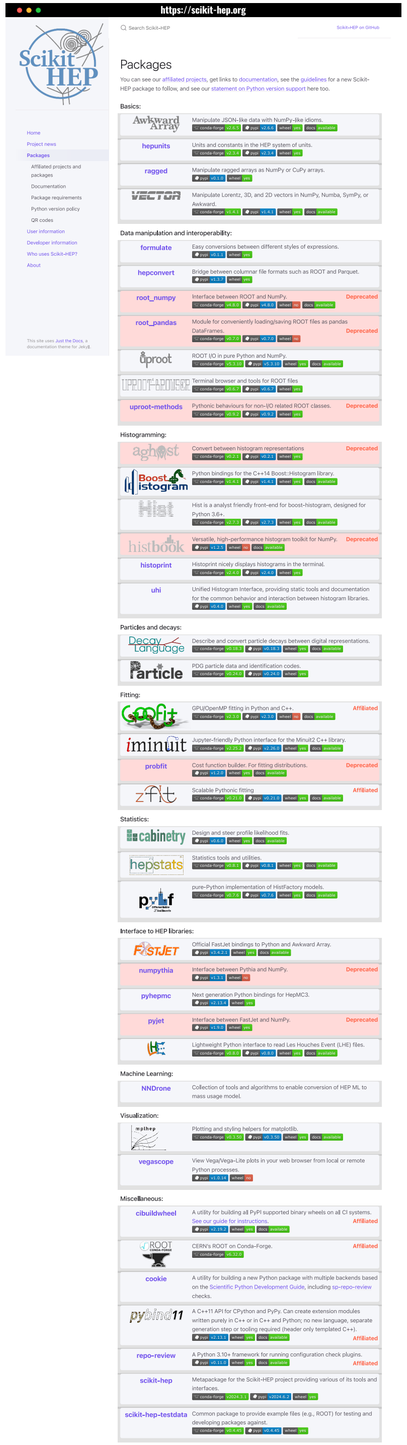
2018 - 2029

An ecosystem of analysis tools



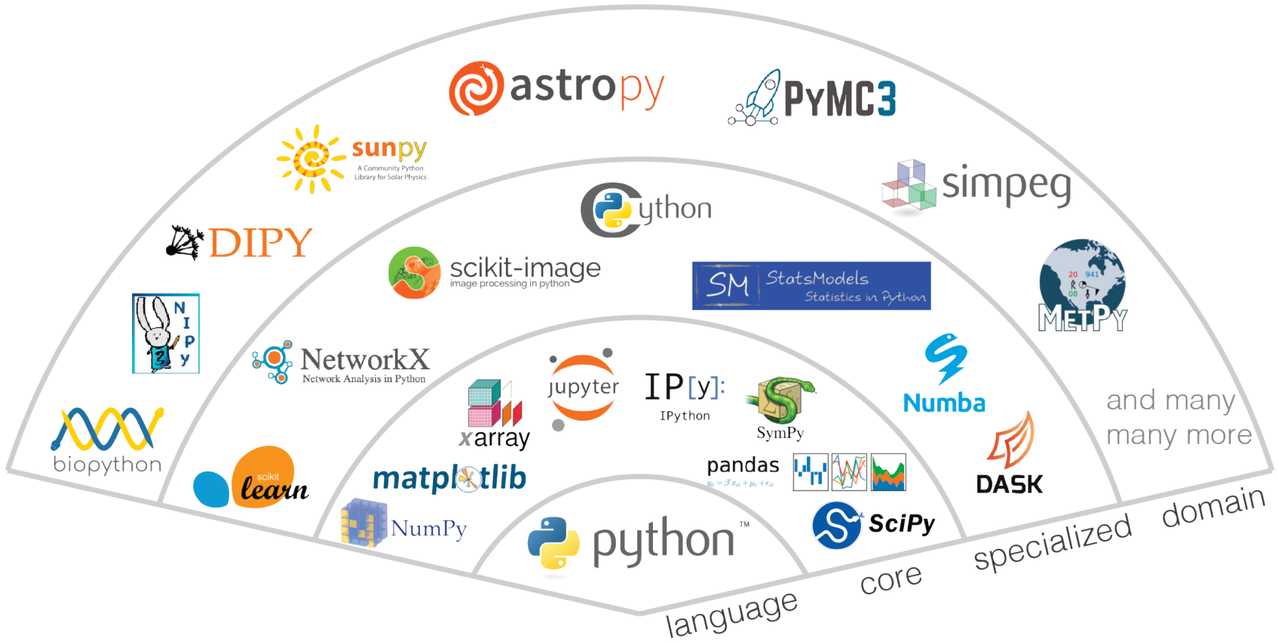
A thriving open source community





Recognition within broader scientific community

Until recently, HEP was still not well integrated or represented in broader scientific python/software community

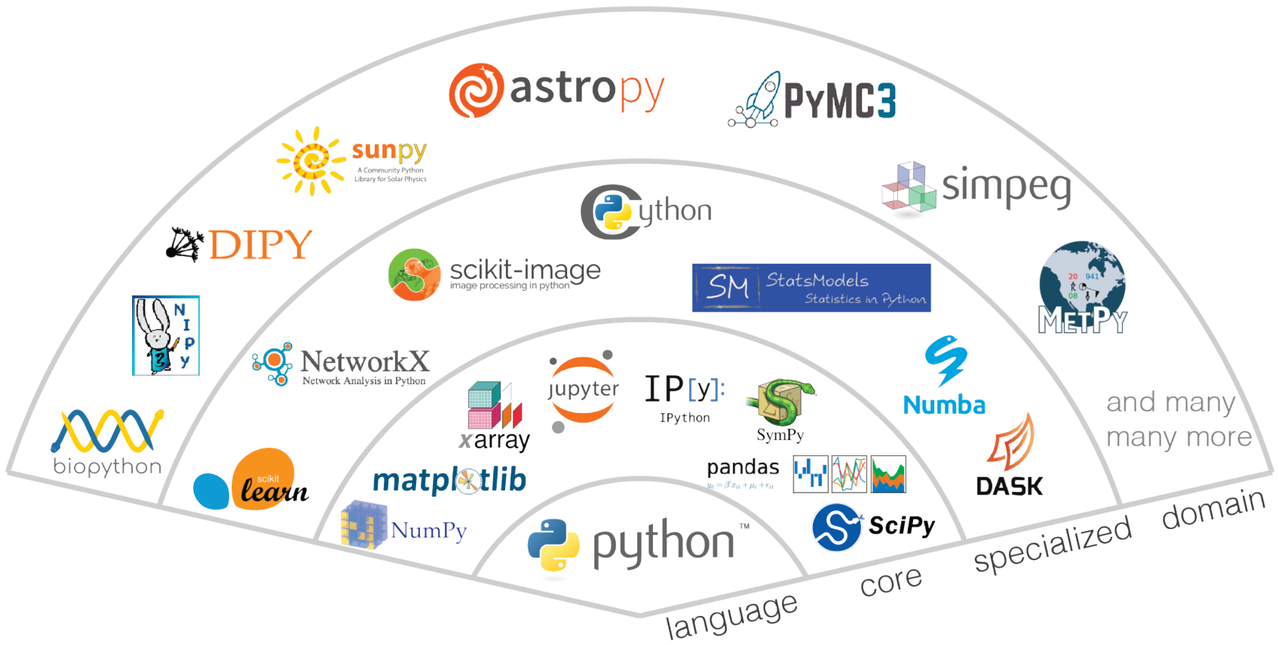


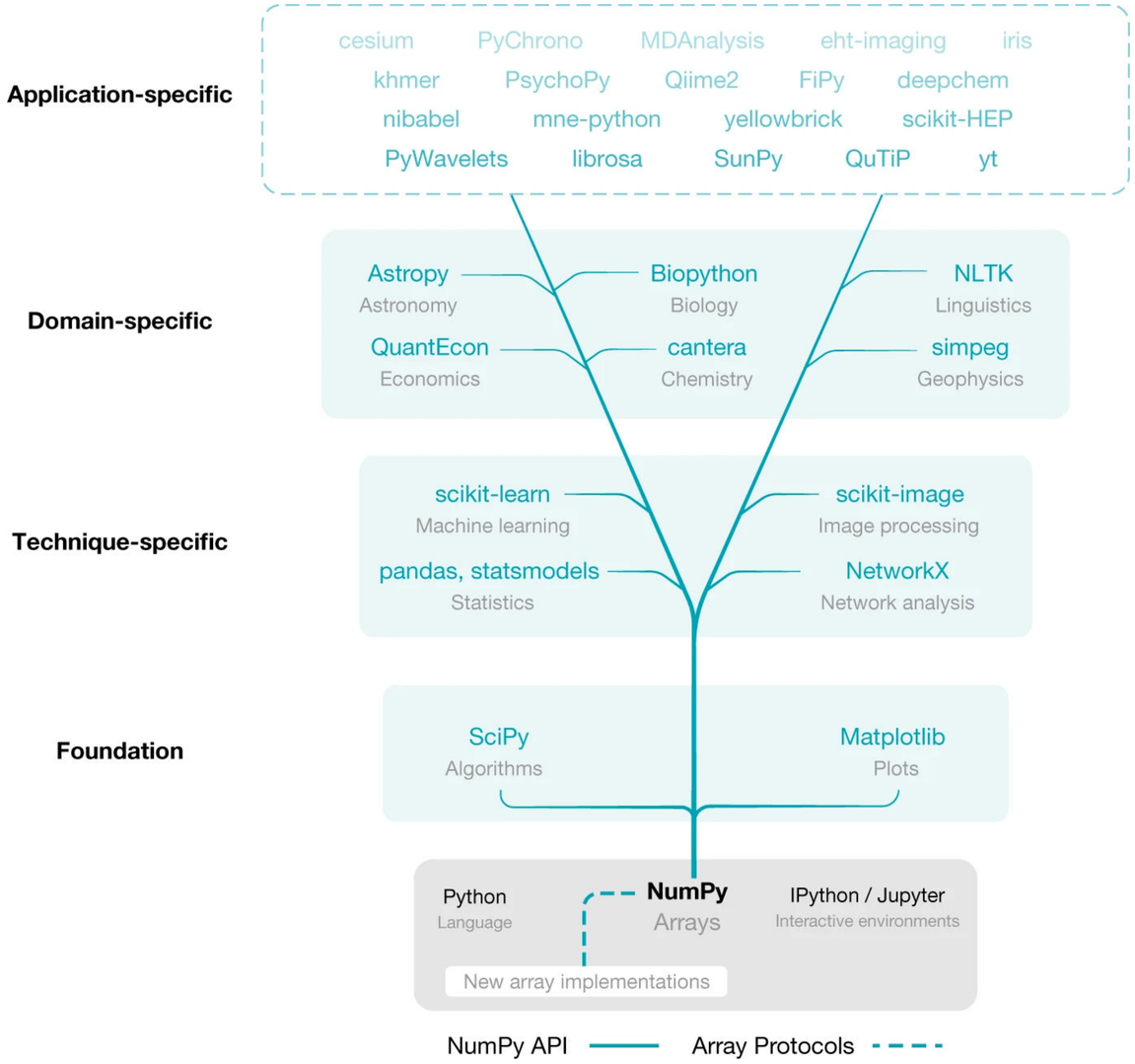
Jake Vanderplas; Jupyter Meets the Earth

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). https://doi.org/10.1038/s41586-020-2649-2

Recognition within broader scientific community

Until recently, HEP was still not well integrated or represented in broader scientific python/software community



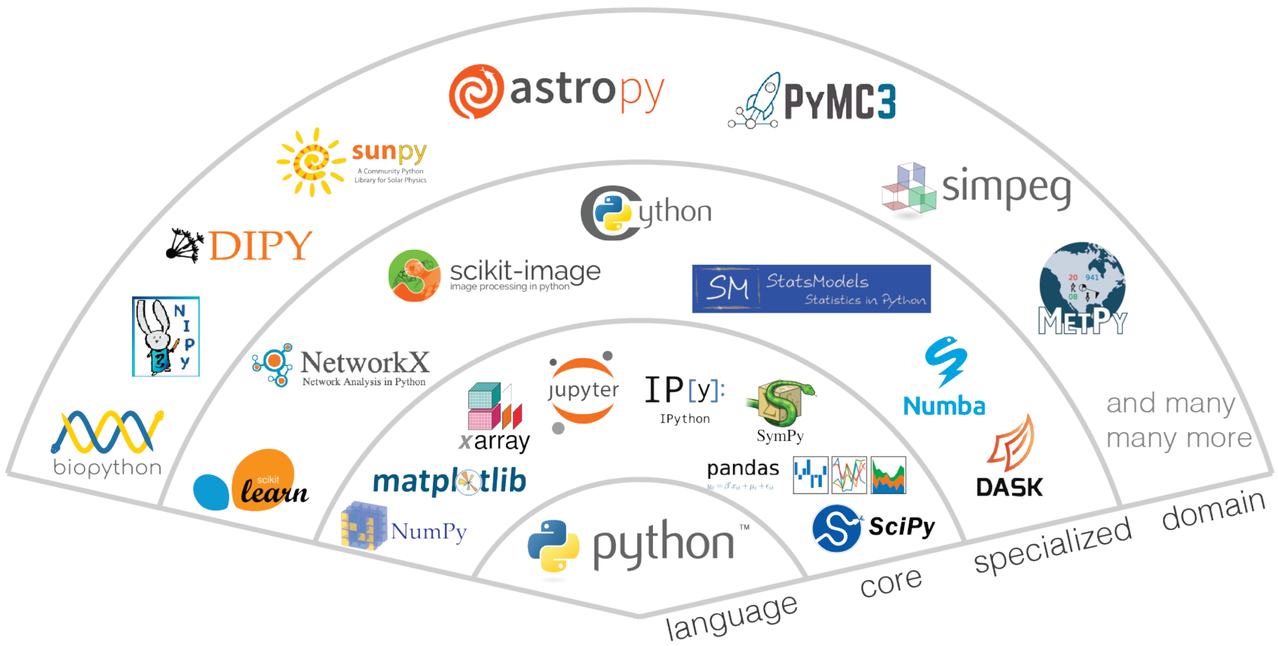


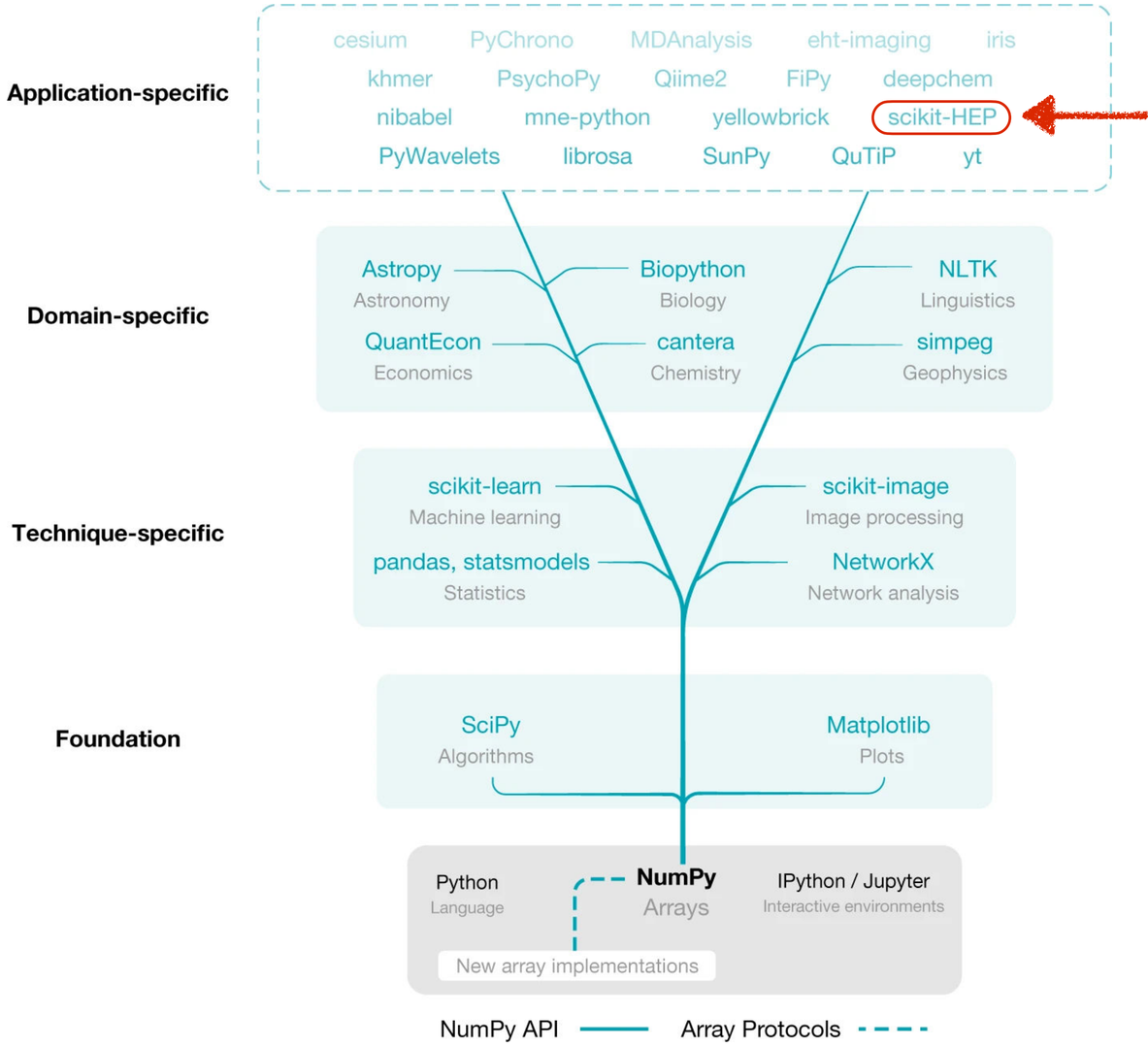
Jake Vanderplas; Jupyter Meets the Earth

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). https://doi.org/10.1038/s41586-020-2649-2

Recognition within broader scientific community

Until recently, HEP was still not well integrated or represented in broader scientific python/software community





Jake Vanderplas; Jupyter Meets the Earth

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). https://doi.org/10.1038/s41586-020-2649-2

Some Highlights

The power of a good specification

The original C++ version of the HistFactory tool distributed in ROOT used a configuration file with clear semantics

- The statistical model was very well defined mathematically

Later recognized this as embracing a **declarative specification.**

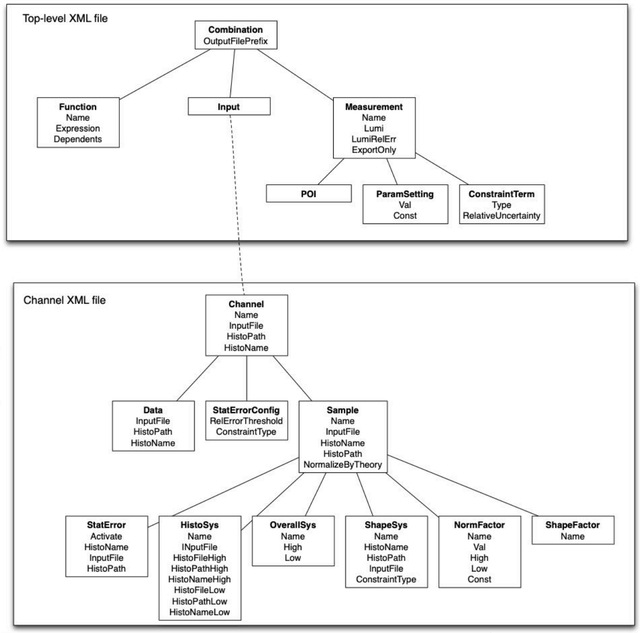
- Facilitated a pythonic implementation

HistFactory

CENTER FOR COSMOLOGY AND PARTICLE PHYSICS

HistFactory tool that ships with ROOT targeting binned analyses

- XML files organize the histograms
- conventions define model exactly
  - CERN-OPEN-2012-016
- command line tool creates likelihood



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

```
<!DOCTYPE Channel  SYSTEM 'HistFactorySchema.dtd'>

  <Channel Name="A" InputFile="./data/ABCD.root"  >
    <Data HistoName="A_data" HistoPath="" />

    <!-- This is the signal (eg. mu)-->
    <Sample Name="A_signal" HistoPath="" HistoName="unit_histogram">
      <!-- now mu is number of events-->
      <NormFactor Name="mu" Val="1" Low="0" High="200" />
      <OverallSys Name="syst1" High="1.01" Low="0.99" />
    </Sample>

    <!-- This bkg is estimated from MC (eg. mu_A^K) -->
    <Sample Name="A_backgroundMC" HistoPath="" NormalizeByTheory="True" HistoName="unit_histogram"  >
      <NormFactor Name="mu_K_A" Val="100" Low="0" High="200" />
    </Sample>

    <!-- Background 2 is completely Data-Driven -->
    <Sample Name="A_backgroundDD" HistoPath="" NormalizeByTheory="False" HistoName="unit_histogram" >
      <NormFactor Name="mu_D_U" Val="100" Low="24500" High="26000" />
      <NormFactor Name="etaB" Val="1" Low="0." High="0.02" Const="False" />
      <NormFactor Name="etaC" Val="1" Low="0." High="0.3" Const="False" />
      <!-- NormFactor and ShapeFactor same for a 1-bin histogram. But we can name NormFactor-->
    </Sample>

  </Channel>
```

The impact of AI/ML frameworks

In 2014, I was investigating how to leverage advances in AI/ML in statistics (more on that later)

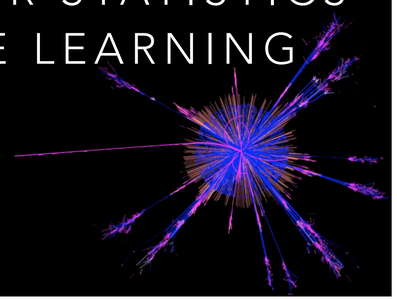
- In 2015, TensorFlow was announced, and we immediately saw advantages of reimplementing statistical tools in a AI/ML framework with automatic differentiation

NYU Center for Data Science

Center for Cosmology and Particle Physics

EMERGING TRENDS IN

SOFTWARE FOR STATISTICS AND MACHINE LEARNING



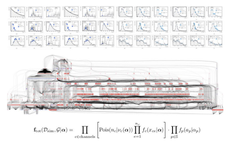
@KyleCranmer
New York University
Department of Physics
Center for Data Science

2nd S2I2 HEP/CS Workshop

1 May 2017, 07:00 → 3 May 2017, 13:35 US/Eastern

Princeton University

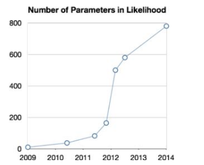
Probabilistic programming frameworks



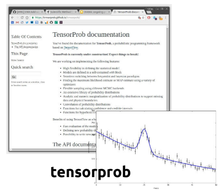
RooFit serves us well, but shows limits in terms of **scalability**.

Using a data flow graph framework, RooFit would be **distributed**, **GPU-enabled** and automatically **differentiable**.

Feasibility? Certainly **within reach!** As illustrated by our tentative proof-of-concepts carl.distributions [Gilles Louppe] and tensorprob [Igor Babuschkin, now at DeepMind]. See also Edward.



carl.distributions



tensorprob

Edward

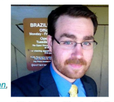
A library for probabilistic modeling, inference, and criticism.

Edward is a Python library for probabilistic modeling, inference, and criticism. It is a testbed for fast experimentation and research with probabilistic models, ranging from classical hierarchical models on small data sets to complex deep probabilistic models on large data sets. Edward fuses three fields: Bayesian statistics and machine learning, deep learning, and probabilistic programming.

It supports **modeling** with

Dustin Tran

Ph.D. Student
Columbia University
dustin@cs.columbia.edu (@dustinvtran, http://dustintran.com



Matthew Feickert

High Energy Physics Ph.D. Candidate
Southern Methodist University
matthew.feickert@cern.ch or mfeickert@smu.edu
GitHub: matthewfeickert @HEPfeickert

The modern AI/ML software stack

Recent switch to

- Numerical computations with data flow graphs
  - TensoFlow, Theano, MXNet, etc
  - Support for CPUs and GPUs out of the box.
  - Automatic differentiation
  - Enable new ways of thinking (model composition, learning to learn, etc)
- Probabilistic programming languages
  - Stan, Anglican, Edward, etc

Recommendation. The next generation of physics software for high-level analysis should take notice and inspiration from the AI/ML community.

TensorFlow

PYTORCH

Edward

10

https://indico.cern.ch/event/622920/timetable/#sc-5-2-lightning-talk-emerging

pyhf matures

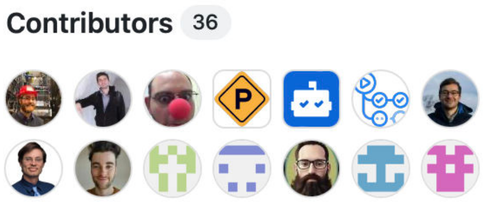
Thanks to the amazing developer team, pyhf quickly matured along with scikit-hep &PyHEP community

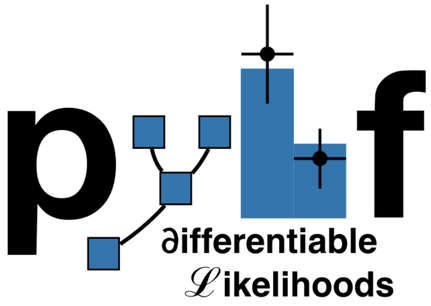
- Now it is a NumFOCUS affiliated project


Matthew Feickert


Giordon Stark


Lukas Heinrich





pyhf: pure-Python implementation of HistFactory statistical models

Lukas Heinrich[1], **Matthew Feickert**[*2], **Giordon Stark**[3], **and Kyle Cranmer**[4]

1 CERN **2** University of Illinois at Urbana-Champaign **3** SCIPP, University of California, Santa Cruz **4** New York University

DOI: 10.21105/joss.02823

Software
- Review
- Repository
- Archive

Editor: Eloisa Bentivegna
Reviewers:
- @suchitakulkarni
- @bradkav

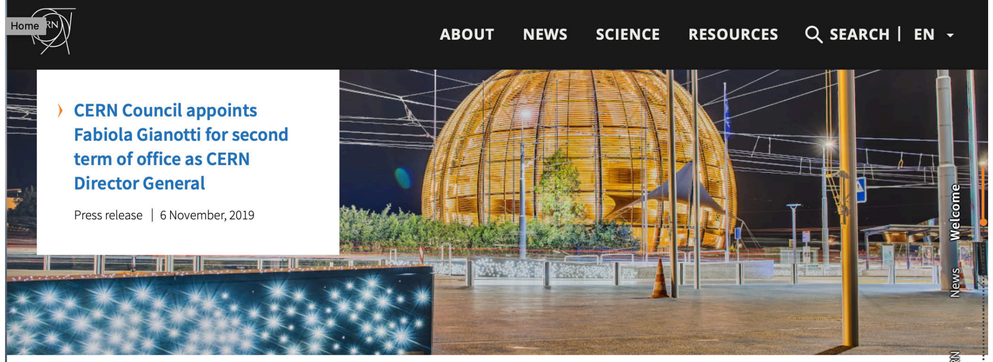
Submitted: 07 October 2020
Published: 04 February 2021

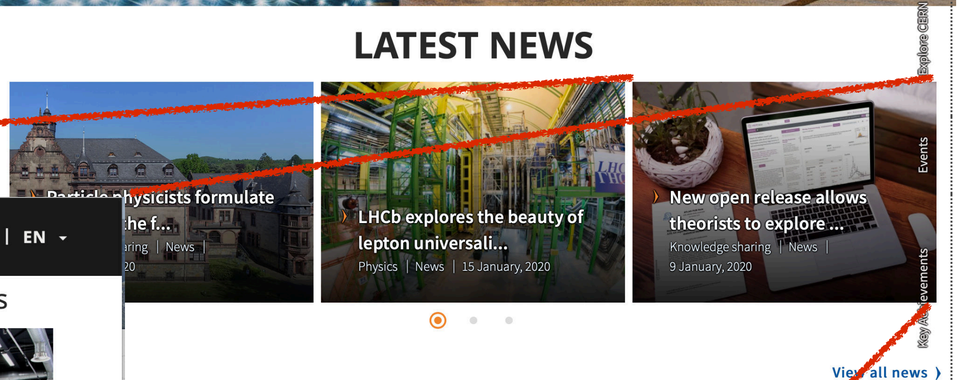
Summary

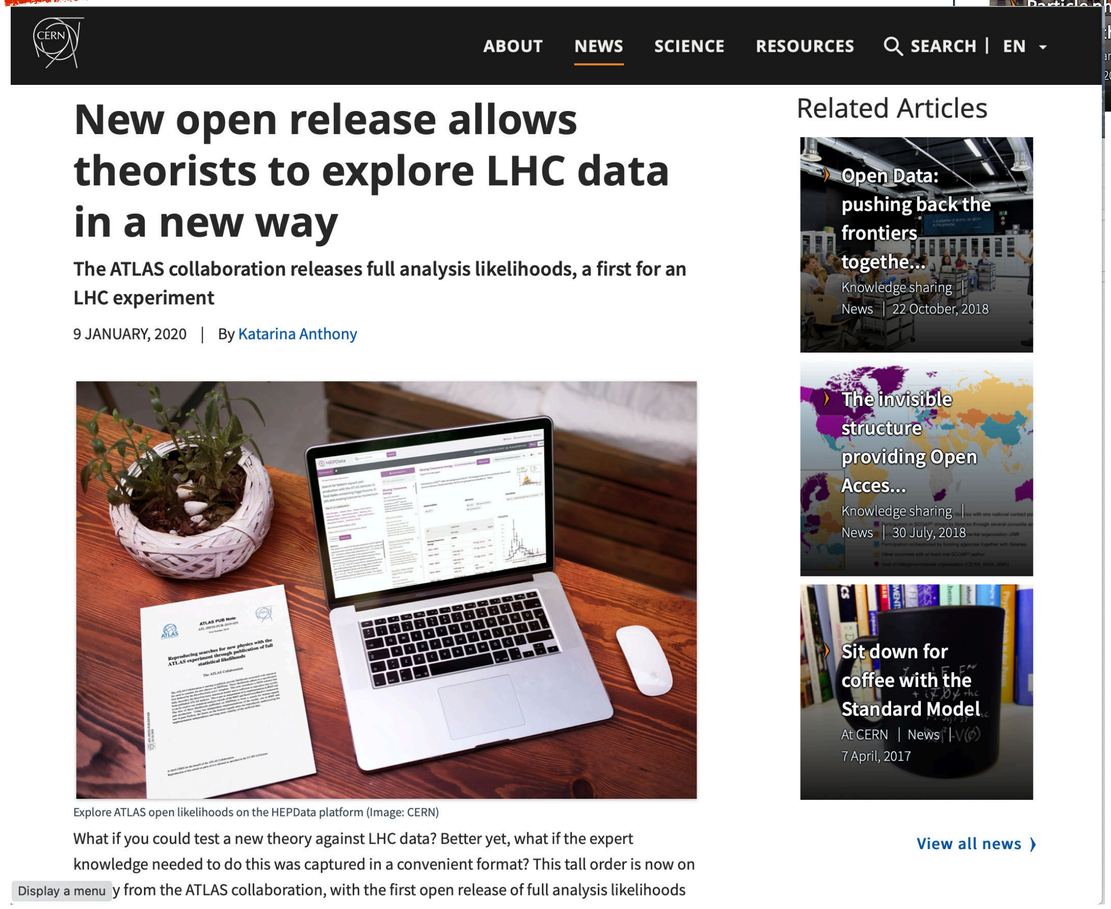
Statistical analysis of High Energy Physics (HEP) data relies on quantifying the compatibility of observed collision events with theoretical predictions. The relationship between them is often formalised in a statistical model $f(\mathbf{x}|\phi)$ describing the probability of data $\mathbf{x}$ given model parameters ϕ. Given observed data, the likelihood $\mathcal{L}(\phi)$ then serves as the basis for inference on the parameters ϕ. For measurements based on binned data (histograms), the `HistFactory` family of statistical models (Cranmer et al., 2012) has been widely used in both Standard Model measurements (ATLAS Collaboration, 2013) as well as searches for new physics (ATLAS Collaboration, 2018). `pyhf` is a pure-Python implementation of the `HistFactory` model specification and implements a declarative, plain-text format for describing `HistFactory`-based likelihoods that is targeted for reinterpretation and long-term preservation in analysis data repositories such as HEPData (Maguire et al., 2017). The source code for `pyhf` has been archived on Zenodo with the linked DOI: (Heinrich, Lukas and Feickert, Matthew and Stark, Giordon, 2020). At the time of writing this paper, the most recent release of `pyhf` is `v0.5.4`.

Realizing a 20 year old dream

The pyhf JSON serialization format facilitated publishing statistical models, realizing a 20 year old dream for the field.







pyhf
differentiable Likelihoods

https://scikit-hep.org/pyhf/

SciPost Physics [arxiv:2109.04981] Submission

Publishing statistical models: Getting the most out of particle physics experiments

Kyle Cranmer[1*], Sabine Kraml[2‡], Harrison B. Prosper[3§] (editors), Philip Bechtle[4], Florian U. Bernlochner[4], Itay M. Bloch[5], Enzo Canonero[6], Marcin Chrzaszcz[7], Andrea Coccaro[8], Jan Conrad[9], Glen Cowan[10], Matthew Feickert[11], Nahuel Ferreiro Iachellini[12,13] Andrew Fowlie[14], Lukas Heinrich[15], Alexander Held[1], Thomas Kuhr[13,16], Anders Kvellestad[17], Maeve Madigan[18], Farvah Mahmoudi[15,19], Knut Dundas Morå[20], Mark S. Neubauer[11], Maurizio Pierini[15], Juan Rojo[8], Sezen Sekmen[22], Luca Silvestrini[23], Veronica Sanz[24,25], Giordon Stark[26], Riccardo Torre[8], Robert Thorne[27], Wolfgang Waltenberger[28], Nicholas Wardle[29], Jonas Wittbrodt[30]

It's a reality

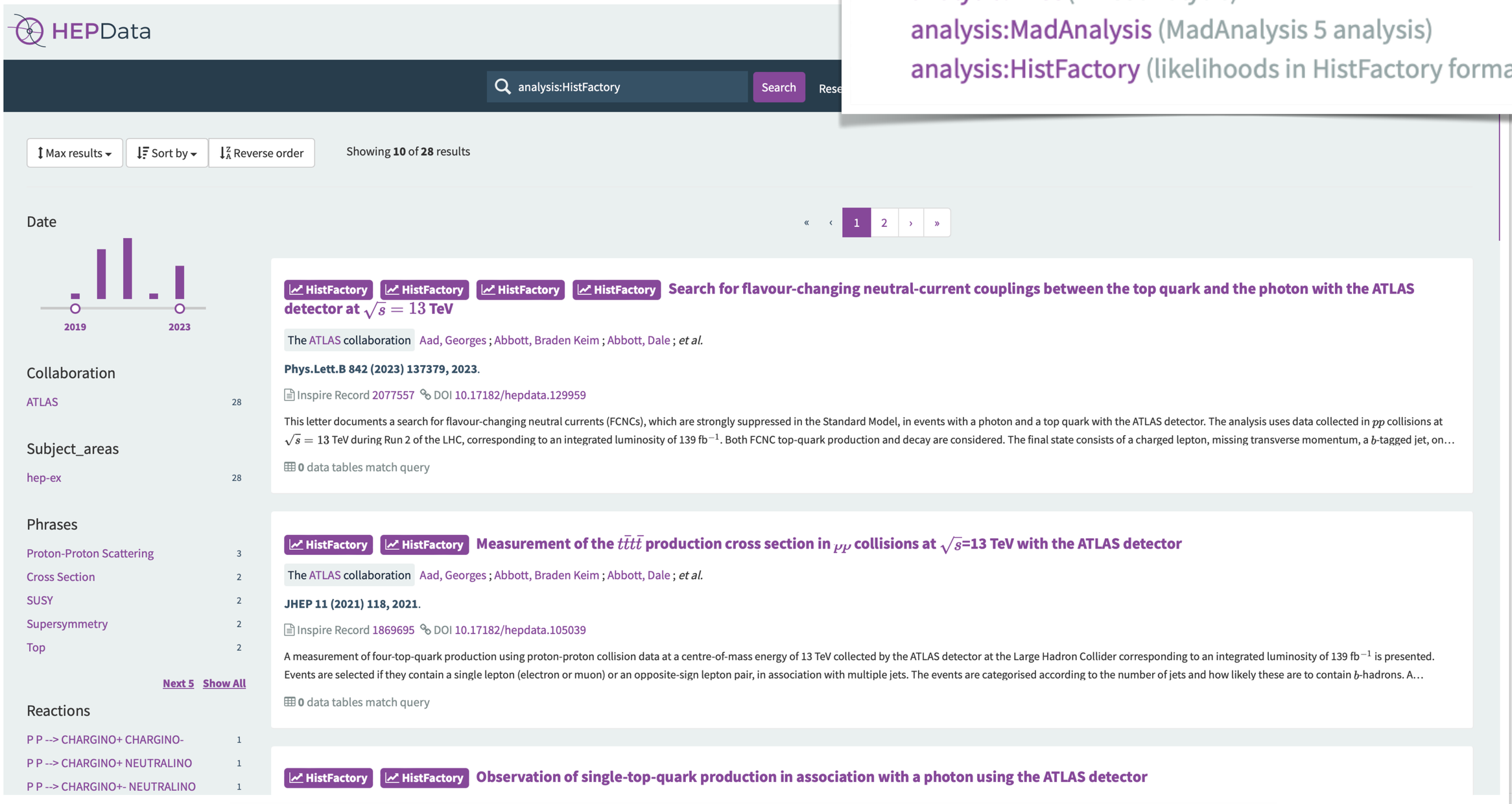
https://www.hepdata.net/search/?q=analysis:HistFactory

Find all papers which include specific types of **analysis**.

analysis:rivet (Rivet analysis)

analysis:MadAnalysis (MadAnalysis 5 analysis)

analysis:HistFactory (likelihoods in HistFactory format)



It's a reality

INSPIRE HEP | literature | Help | Submit | Login

Literature | Authors | Jobs | Seminars | Conferences | More...

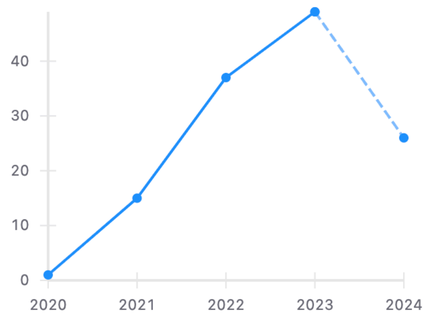
pyhf: pure-Python implementation of HistFactory statistical models

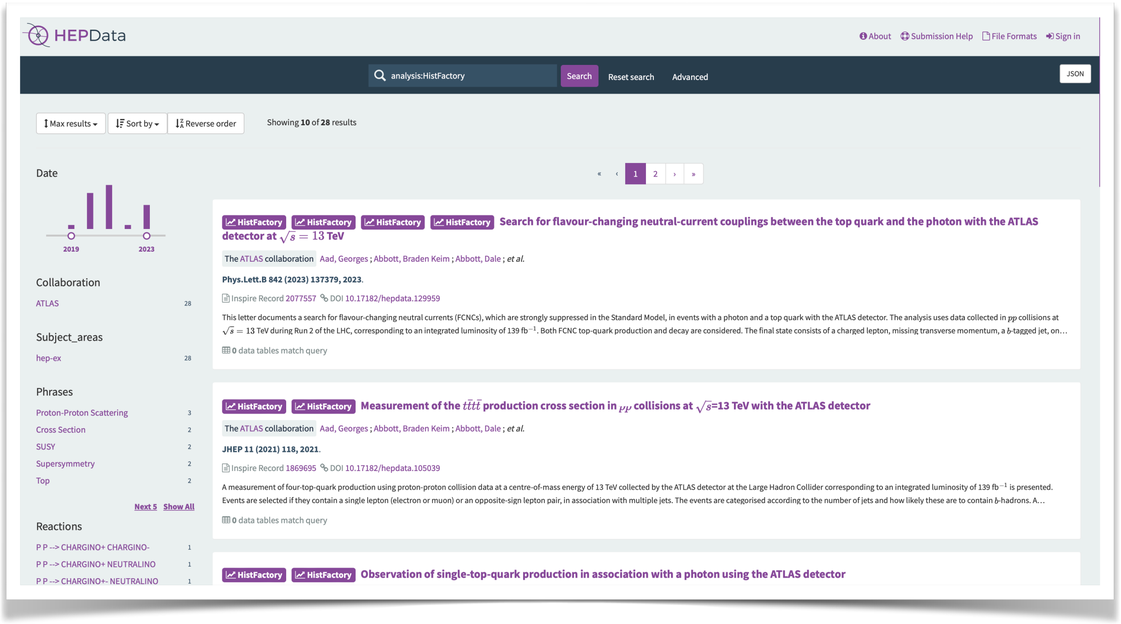
Lukas Heinrich (CERN), Matthew Feickert (Illinois U., Urbana), Giordon Stark (UC, Santa Cruz, Inst. Part. Phys.), Kyle Cranmer (New York U.)
Feb 4, 2021

2 pages
Published in: *J.Open Source Softw.* 6 (2021) 58, 2823
Published: Feb 4, 2021
DOI: 10.21105/joss.02823
View in: CERN Document Server

pdf | cite | claim | reference search | 128 citations

Citations per year





https://scikit-hep.org

pyhf 0.7.1.dev266 documentation

Use and Citations

Use and Citations

Warning: This is a development version and should not be cited. To find the specific version to cite, please go to ReadTheDocs.

Citation

The preferred BibTeX entry for citation of `pyhf` includes both the Zenodo archive and the JOSS paper:

```
@software{pyhf,
  author = {Lukas Heinrich and Matthew Feickert and Giordon Stark},
  title = "{pyhf: v0.7.6}",
  version = {0.7.6},
  doi = {10.5281/zenodo.1169739},
  url = {https://doi.org/10.5281/zenodo.1169739},
  note = {https://github.com/scikit-hep/pyhf/releases/tag/v0.7.6}
}

@article{pyhf_joss,
  doi = {10.21105/joss.02823},
  url = {https://doi.org/10.21105/joss.02823},
  year = {2021},
  publisher = {The Open Journal},
  volume = {6},
  number = {58},
  pages = {2823},
  author = {Lukas Heinrich and Matthew Feickert and Giordon Stark and Kyle Cranmer},
  title = {pyhf: pure-Python implementation of HistFactory statistical models},
  journal = {Journal of Open Source Software}
}
```

Use in Publications

The following is an updating list of citations and use cases of `pyhf`. There is an incomplete but automatically updated list of citations on INSPIRE as well.

Use Citations

1. Alexander Held, Elliott Kauffman, Oksana Shadura, and Andrew Wightman. Physics analysis for the HL-LHC: concepts and pipelines in practice with the Analysis Grand Challenge. 1 2024. arXiv:2401.02766.
2. Priyotosh Bandyopadhyay, Snehashis Parashar, Chandrima Sen, and Jeonghyeon Song. Probing Inert Triplet Model at a multi-TeV muon collider via vector boson fusion with forward muon tagging. 1 2024. arXiv:2401.02697.
3. Elliott Kauffman, Alexander Held, and Oksana Shadura. Machine Learning for Columnar High Energy Physics Analysis. 1 2024. arXiv:2401.01802.
4. Mohammad Mahdi Altakach, Sabine Kraml, Andre Lessa, Sahana Narasimha, Timothée Pascal, Théo Reymermier, and Wolfgang Waltenberger. Global LHC constraints on electroweak-inos with SModelS v2.3. 12 2023. arXiv:2312.16635.
5. MicroBooNE Collaboration. First search for dark-trident processes using the MicroBooNE detector. 12 2023. arXiv:2312.13945.
6. Leandro Da Rold, Manuel Epele, Anibal D. Medina, Nicolás I. Mileo, and Alejandro Szynkman. Double Higgs production at the HL-LHC: probing a loop-enhanced model with kinematical distributions. 12 2023. arXiv:2312.13149.

It's a reality

INSPIRE HEP | literature | Help | Submit | Login

Literature | Authors | Jobs | Seminars | Conferences | More...

pyhf: pure-Python implementation of HistFactory statistical models

Lukas Heinrich (CERN), Matthew Feickert (Illinois U., Urbana), Giordon Stark (UC, Santa Cruz, Inst. Part. Phys.), Kyle Cranmer (New York U.)

Feb 4, 2021

2 pages

Published in: *J.Open Source Softw.* 6 (2021) 58, 2823

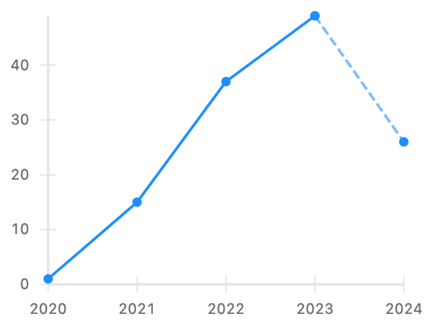
Published: Feb 4, 2021

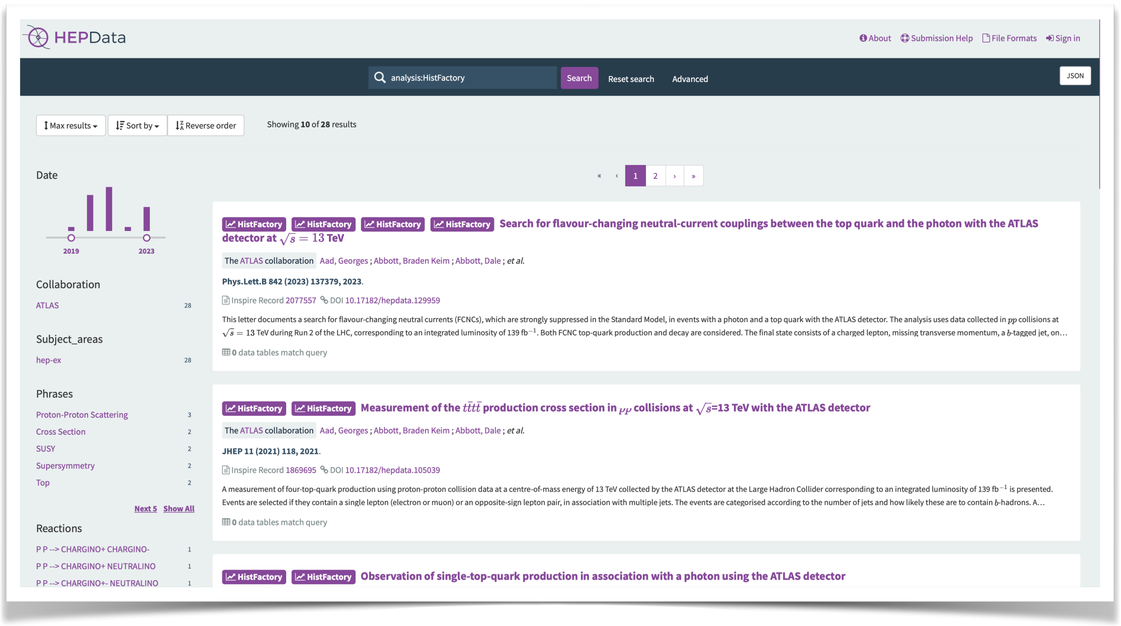
DOI: 10.21105/joss.02823

View in: CERN Document Server

pdf | cite | claim | reference search | 128 citations

Citations per year





11. Measurements of the inclusive and differential production cross sections of a top-quark-antiquark pair in association with a Z boson at $\sqrt s = 13$ TeV with the ATLAS detector. 2021. URL: https://doi.org/10.17182/hepdata.100351, doi:10.17182/hepdata.100351.
12. Search for pair production of third-generation scalar leptoquarks decaying into a top quark and a τ-lepton in pp collisions at $ \sqrt s $ = 13 TeV with the ATLAS detector. 2021. URL: https://doi.org/10.17182/hepdata.100174, doi:10.17182/hepdata.100174.
13. Search for squarks and gluinos in final states with one isolated lepton, jets, and missing transverse momentum at $\sqrt s=13$ with the ATLAS detector. 2021. URL: https://doi.org/10.17182/hepdata.97041, doi:10.17182/hepdata.97041.
14. Search for trilepton resonances from chargino and neutralino pair production in $\sqrt s$ = 13 TeV pp collisions with the ATLAS detector. 2020. URL: https://doi.org/10.17182/hepdata.99806, doi:10.17182/hepdata.99806.
15. Search for Displaced Leptons in $\sqrt s = 13$ TeV pp Collisions with the ATLAS Detector. 2020. URL: https://doi.org/10.17182/hepdata.98796, doi:10.17182/hepdata.98796.
16. Search for squarks and gluinos in final states with jets and missing transverse momentum using 139 fb$^-1$ of $\sqrt s$ =13 TeV pp collision data with the ATLAS detector. 2020. URL: https://doi.org/10.17182/hepdata.95664, doi:10.17182/hepdata.95664.
17. Evidence for $t\bar tt\bar t$ production in the multilepton final state in proton-proton collisions at $\sqrt s=13$ $\text TeV$ with the ATLAS detector. 2020. URL: https://doi.org/10.17182/hepdata.100170, doi:10.17182/hepdata.100170.
18. Measurement of the $t\bar t$ production cross-section in the lepton+jets channel at $\sqrt s=13$ TeV with the ATLAS experiment. 2020. URL: https://doi.org/10.17182/hepdata.95748, doi:10.17182/hepdata.95748.
19. Search for long-lived, massive particles in events with a displaced vertex and a muon with large impact parameter in pp collisions at $\sqrt s = 13$ TeV with the ATLAS detector. 2020. URL: https://doi.org/10.17182/hepdata.91760, doi:10.17182/hepdata.91760.
20. Search for chargino-neutralino production with mass splittings near the electroweak scale in three-lepton final states in $\sqrt s$ = 13 TeV pp collisions with the ATLAS detector. 2019. URL: https://doi.org/10.17182/hepdata.91127, doi:10.17182/hepdata.91127.
21. Searches for electroweak production of supersymmetric particles with compressed mass spectra in $\sqrt s=$ 13 TeV pp collisions with the ATLAS detector. 2019. URL: https://doi.org/10.17182/hepdata.91374, doi:10.17182/hepdata.91374.
22. Search for direct stau production in events with two hadronic τ-leptons in $\sqrt s = 13$ TeV pp collisions with the ATLAS detector. 2019. URL: https://doi.org/10.17182/hepdata.92006, doi:10.17182/hepdata.92006.
23. Search for direct production of electroweakinos in final states with one lepton, missing transverse momentum and a Higgs boson decaying into two b-jets in (pp) collisions at $\sqrt s=13$ TeV with the ATLAS detector. 2019. URL: https://doi.org/10.17182/hepdata.90607.v2, doi:10.17182/hepdata.90607.v2.
24. Search for squarks and gluinos in final states with same-sign leptons and jets using 139 fb$^-1$ of data collected with the ATLAS detector. 2019. URL: https://doi.org/10.17182/hepdata.91214.v3, doi:10.17182/hepdata.91214.v3.
25. Search for electroweak production of charginos and sleptons decaying into final states with two leptons and missing transverse momentum in $\sqrt s=13$ TeV pp collisions using the ATLAS detector. 2019. URL: https://doi.org/10.17182/hepdata.89413, doi:10.17182/hepdata.89413.
26. Search for bottom-squark pair production with the ATLAS detector in final states containing Higgs bosons, b-jets and missing transverse momentum. 2019. URL: https://doi.org/10.17182/hepdata.89408, doi:10.17182/hepdata.89408.

Note

ATLAS maintains a public listing of all published statistical models on the ATLAS public results page which can be found by filtering all public results by the "Likelihood available" analysis characteristics keyword.

Using published likelihoods

cabinetry

https://github.com/scikit-hep/cabinetry

```python
import json
import cabinetry
import pyhf
from cabinetry.model_utils import prediction
from pyhf.contrib.utils import download

# download the ATLAS bottom-squarks analysis probability models from HEPData
download("https://www.hepdata.net/record/resource/1935437?view=true", "bottom-squarks")

# construct a workspace from a background-only model and a signal hypothesis
bkg_only_workspace = pyhf.Workspace(json.load(open("bottom-squarks/RegionC/BkgOnly.json")))
patchset = pyhf.PatchSet(json.load(open("bottom-squarks/RegionC/patchset.json")))
workspace = patchset.apply(bkg_only_workspace, "sbottom_600_280_150")

# construct the probability model and observations
model, data = cabinetry.model_utils.model_and_data(workspace)

# produce visualizations of the pre-fit model and observed data
prefit_model = prediction(model)
cabinetry.visualize.data_mc(prefit_model, data)

# fit the model to the observed data
fit_results = cabinetry.fit.fit(model, data)

# produce visualizations of the post-fit model and observed data
postfit_model = prediction(model, fit_results=fit_results)
cabinetry.visualize.data_mc(postfit_model, data)
```

Using published likelihoods

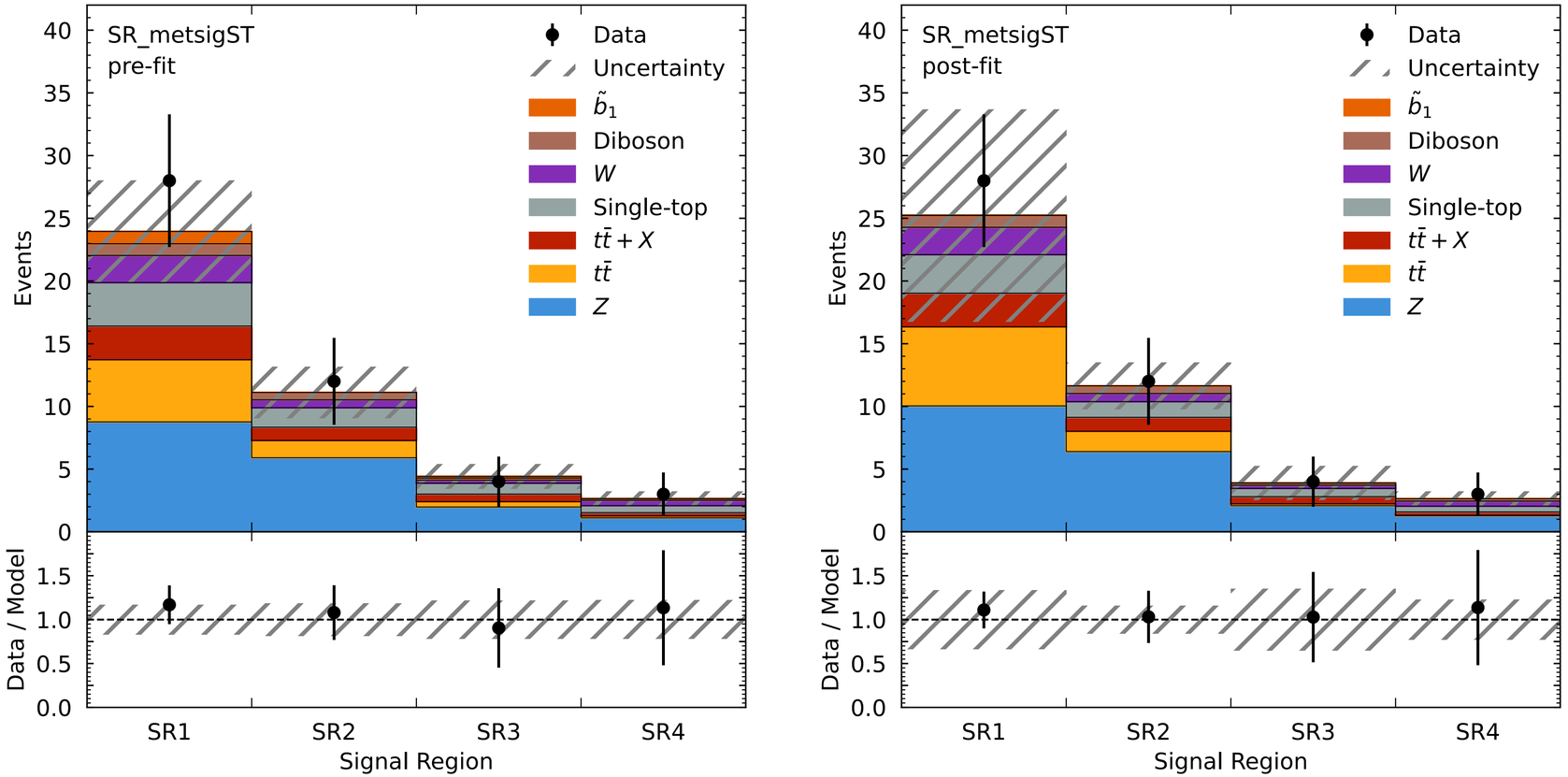
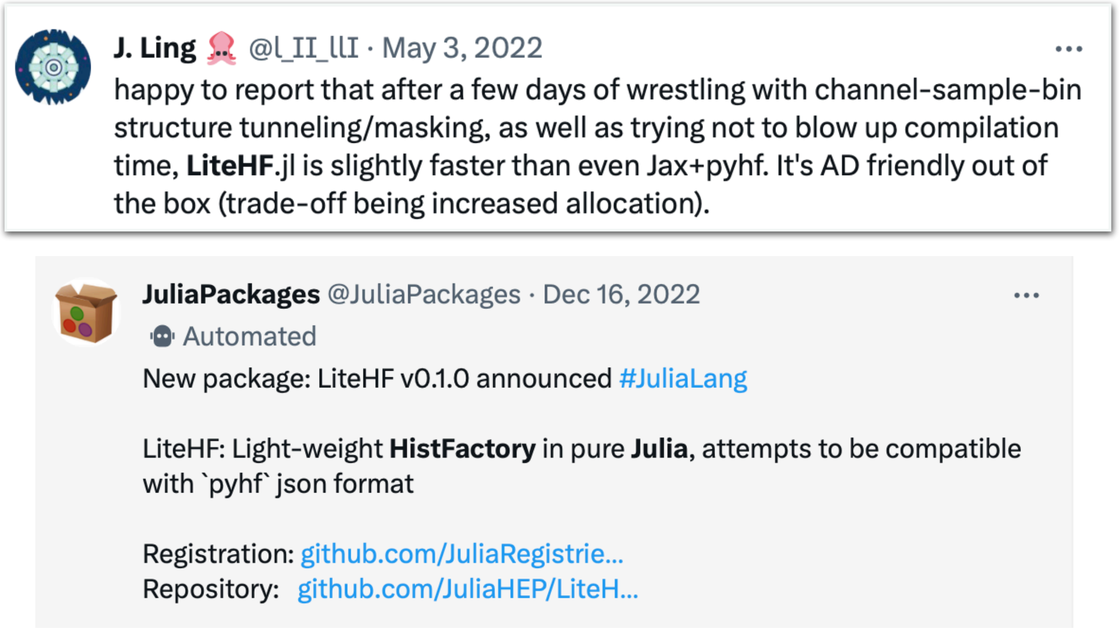


Figure 3: Pre-fit (left) and post-fit (right) visualizations of a selected signal hypothesis for four signal regions of the ATLAS search [41] of a bottom-squark of mass 600 GeV with a second-lightest neutralino of mass 280 GeV and lightest supersymmetric particle of mass 150 GeV generated from the full statistical models published in Ref. [20] using code from Ref. [40].

Declarative specifications FTW!

We are now seeing implementations in julia and an exciting initiative to expand this strategy to other statistical modeling tools



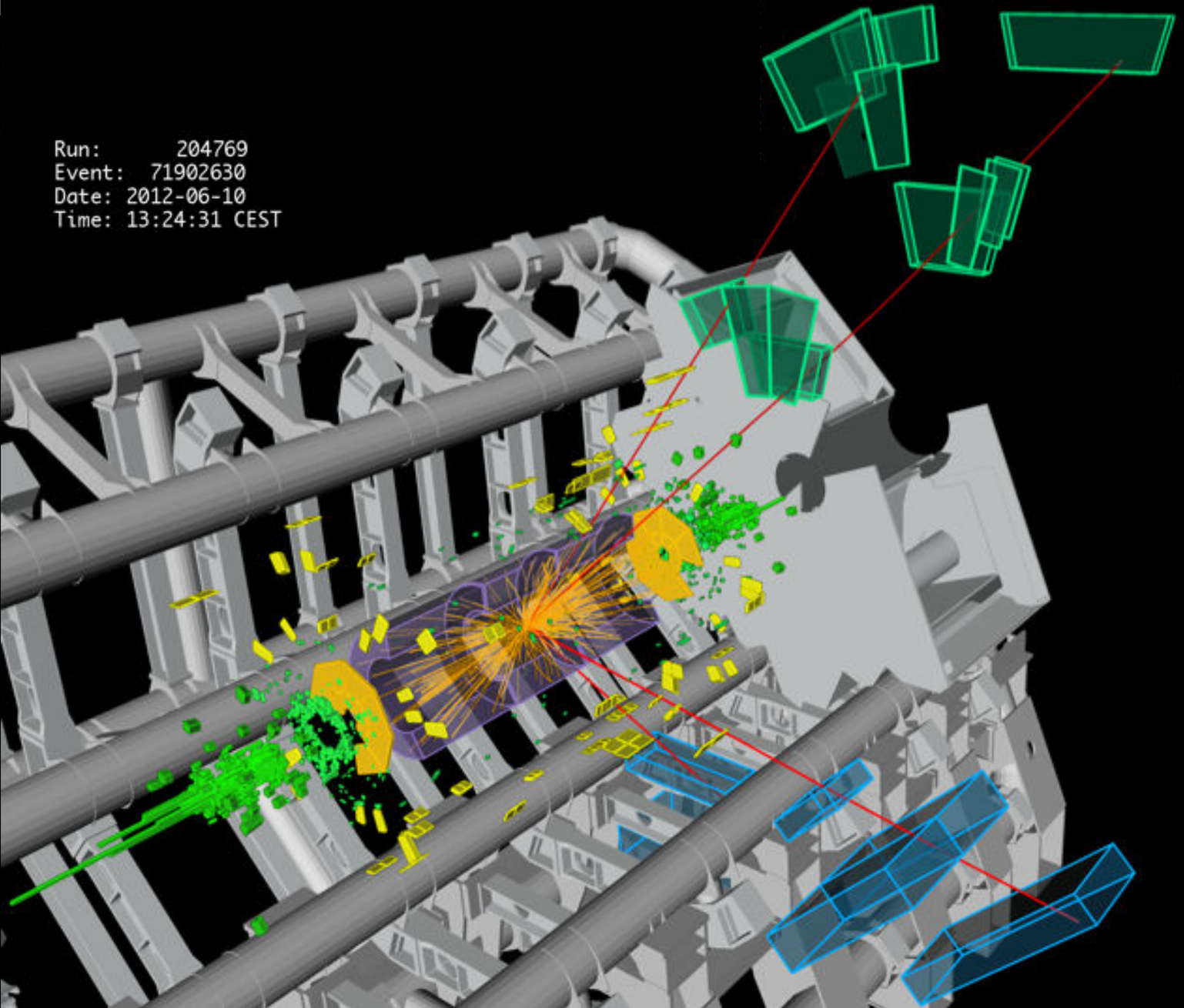
https://github.com/JuliaHEP/LiteHF.jl

https://github.com/hep-statistics-serialization-standard/

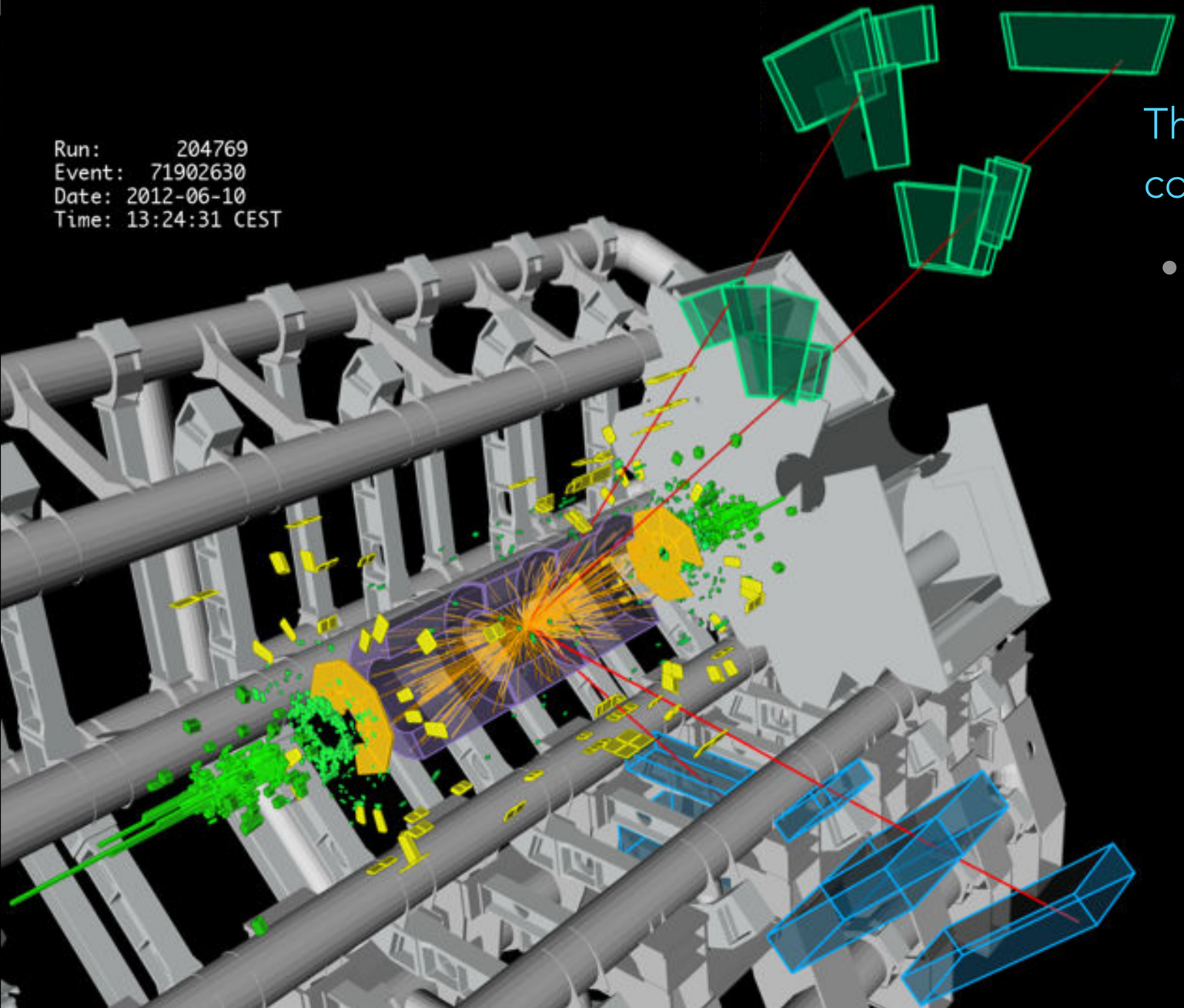
Thinking in Arrays:
the move to columnar data & array-oriented programming


Raw
Data
~1MB/event
Bulk
Data
Processing
Analysis
Data
~1KB/event

Bulk data analysis pipeline



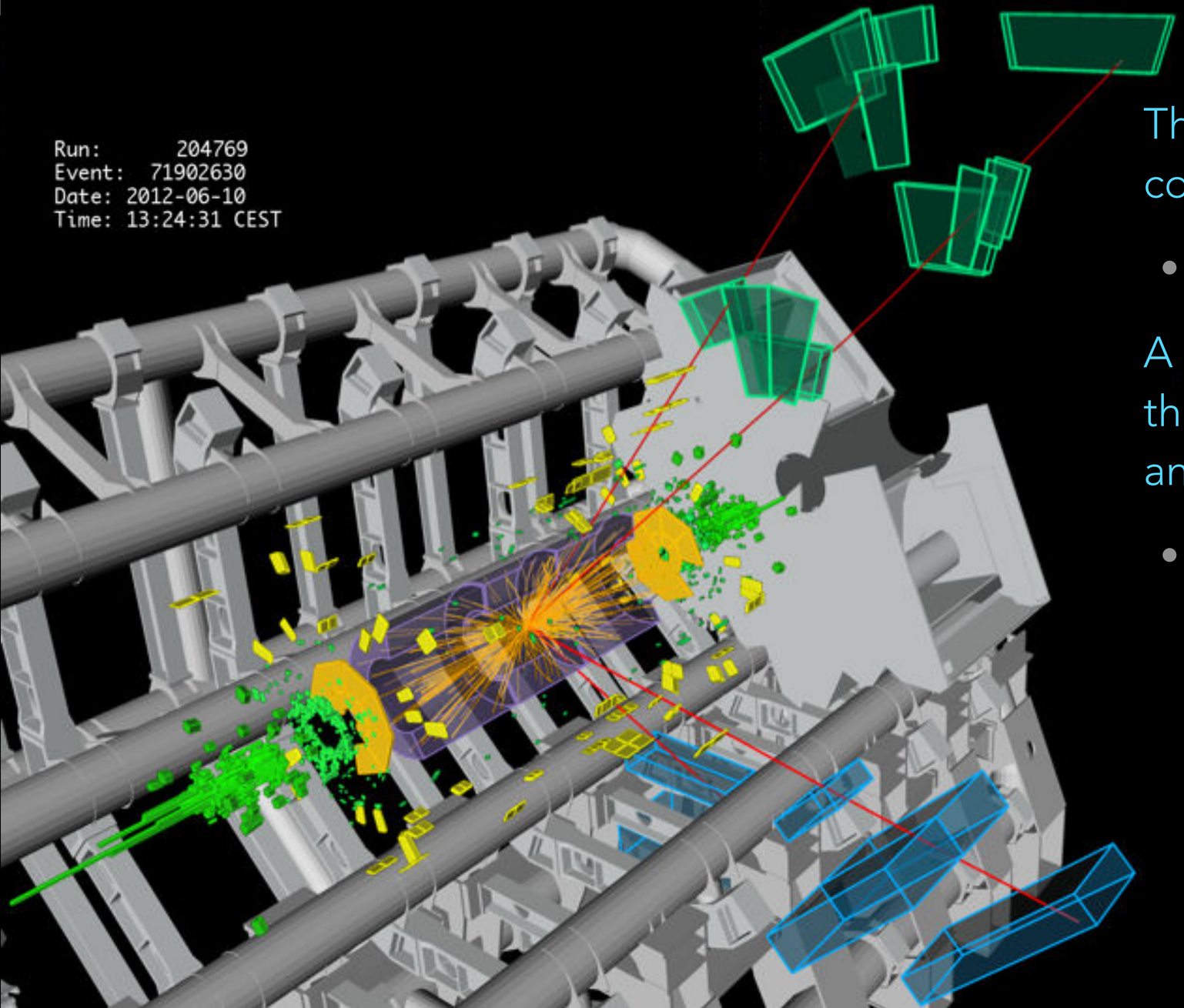
Bulk data analysis pipeline



The raw data associated to one collision is complex & not convenient to work with

- Roughly 100M sensors. Sparse.

Bulk data analysis pipeline



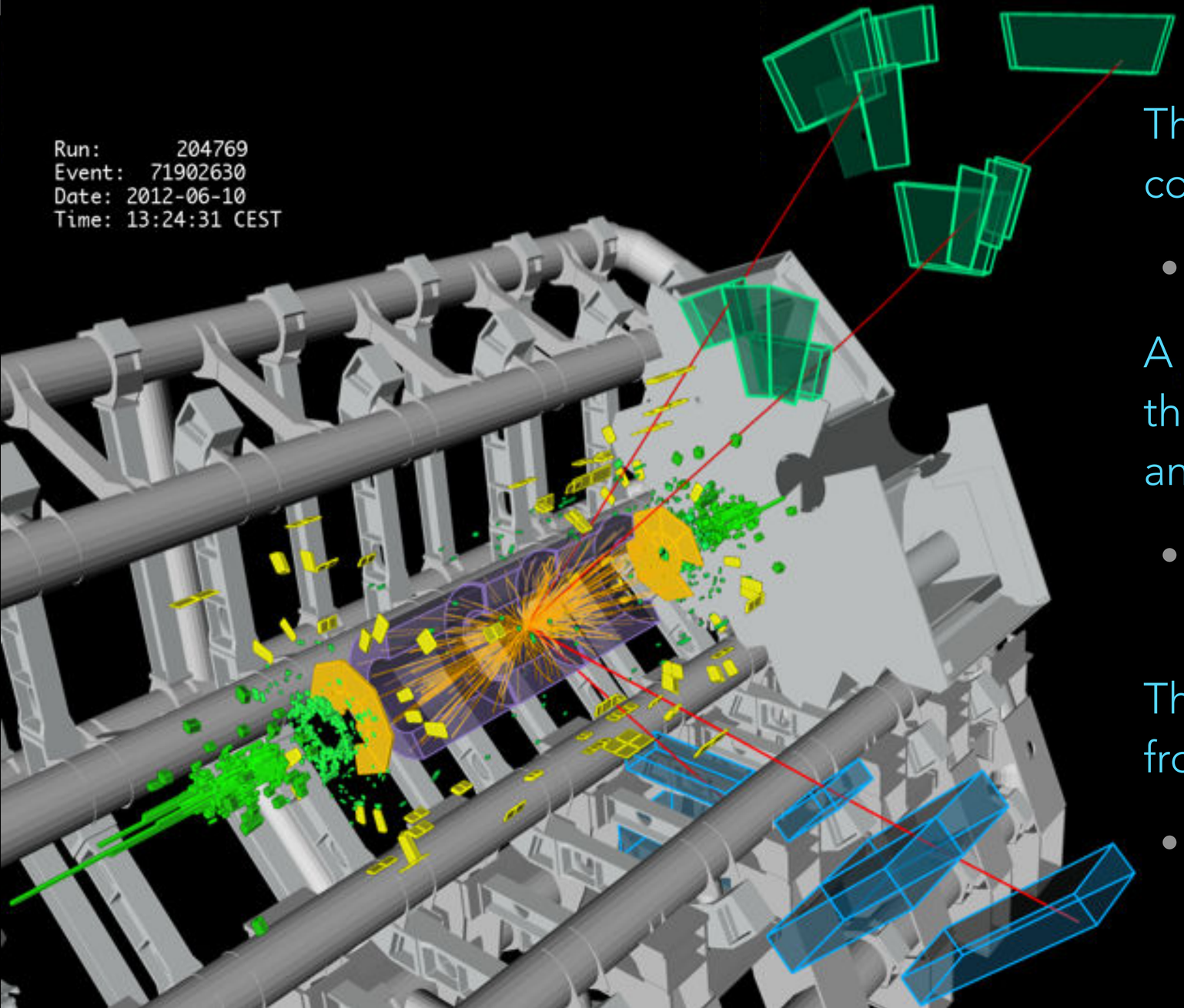
The raw data associated to one collision is complex & not convenient to work with

- Roughly 100M sensors. Sparse.

A **data analysis pipeline** runs algorithms that identify and characterize particles by analyzing energy deposits in detector

- e.g. the red lines & yellow lines are identified as particles of different types

Bulk data analysis pipeline



The raw data associated to one collision is complex & not convenient to work with

- Roughly 100M sensors. Sparse.

A **data analysis pipeline** runs algorithms that identify and characterize particles by analyzing energy deposits in detector

- e.g. the red lines & yellow lines are identified as particles of different types

The number of identified particles varies from collision to collision

- Output is **not tabular data**, but a nested, variable-length data structure

Awkward Array is a library for nested, variable-sized data, including arbitrary-length lists, records, mixed types, and missing data, **using NumPy-like idioms.**

Awkward Array

Awkward Array is a library for nested, variable-sized data, including arbitrary-length lists, records, mixed types, and missing data, **using NumPy-like idioms.**

Awkward Array

Awkward Array is a library for nested, variable-sized data, including arbitrary-length lists, records, mixed types, and missing data, **using NumPy-like idioms.**

```
import awkward as ak

array = ak.Array([
    [{"x": 1.1, "y": [1]}, {"x": 2.2, "y": [1, 2]}, {"x": 3.3, "y": [1, 2, 3]}],
    [],
    [{"x": 4.4, "y": [1, 2, 3, 4]}, {"x": 5.5, "y": [1, 2, 3, 4, 5]}]
])
```

```
array = ak.Array([
    [{"x": 1.1, "y": [1]}, {"x": 2.2, "y": [1, 2]}, {"x": 3.3, "y": [1, 2, 3]}],
    [],
    [{"x": 4.4, "y": [1, 2, 3, 4]}, {"x": 5.5, "y": [1, 2, 3, 4, 5]}]
])
```

```
array = ak.Array([
    [{"x": 1.1, "y": [1]}, {"x": 2.2, "y": [1, 2]}, {"x": 3.3, "y": [1, 2, 3]}],
    [],
    [{"x": 4.4, "y": [1, 2, 3, 4]}, {"x": 5.5, "y": [1, 2, 3, 4, 5]}]
])
```

the following slices out the `y` values, drops the first element from each inner list, and runs NumPy's `np.square` function on everything that is left:

```
output = np.square(array["y", ..., 1:])
```

```
array = ak.Array([
    [{"x": 1.1, "y": [1]}, {"x": 2.2, "y": [1, 2]}, {"x": 3.3, "y": [1, 2, 3]}],
    [],
    [{"x": 4.4, "y": [1, 2, 3, 4]}, {"x": 5.5, "y": [1, 2, 3, 4, 5]}]
])
```

the following slices out the `y` values, drops the first element from each inner list, and runs NumPy's `np.square` function on everything that is left:

```
output = np.square(array["y", ..., 1:])
```

The result is

```
[
    [[], [4], [4, 9]],
    [],
    [[4, 9, 16], [4, 9, 16, 25]]
]
```

The equivalent loop-based approach in pure Python is:

```
output = []
for sublist in array:
    tmp1 = []
    for record in sublist:
        tmp2 = []
        for number in record["y"][1:]:
            tmp2.append(np.square(number))
        tmp1.append(tmp2)
    output.append(tmp1)
```

- the Awkward Array one-liner takes 1.5 seconds to run and uses 2.1 GB of memory,
- the equivalent using Python lists and dicts takes 140 seconds to run and uses 22 GB of memory.

The equivalent loop-based approach in pure Python is:

```
output = []
for sublist in array:
    tmp1 = []
    for record in sublist:
        tmp2 = []
        for number in record["y"][1:]:
            tmp2.append(np.square(number))
        tmp1.append(tmp2)
    output.append(tmp1)
```

Note: we would traditionally write loop-based analysis code in C++

Move to array-oriented paradigm is a big cultural shift.

- the Awkward Array one-liner takes 1.5 seconds to run and uses 2.1 GB of memory,
- the equivalent using Python lists and dicts takes 140 seconds to run and uses 22 GB of memory.

Playing nice

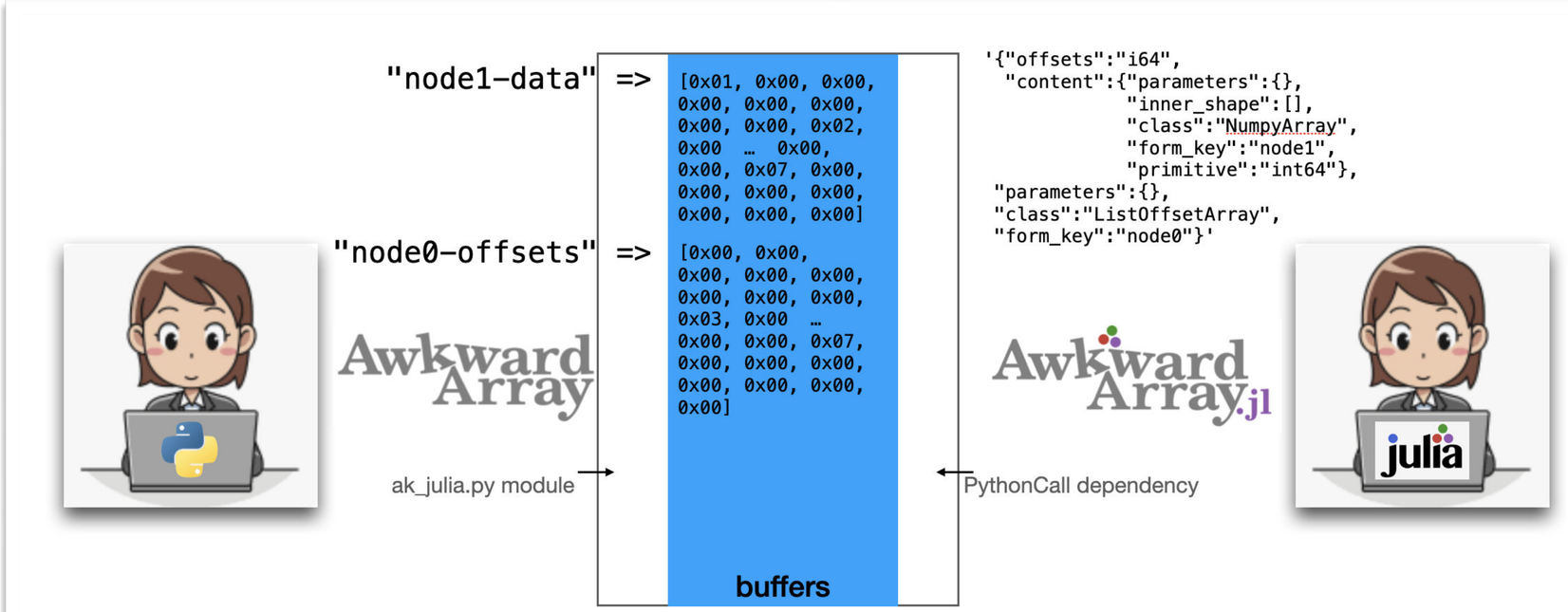
Computer Science > Programming Languages

[Submitted on 28 Apr 2024]

Bridging Worlds: Achieving Language Interoperability between Julia and Python in Scientific Computing

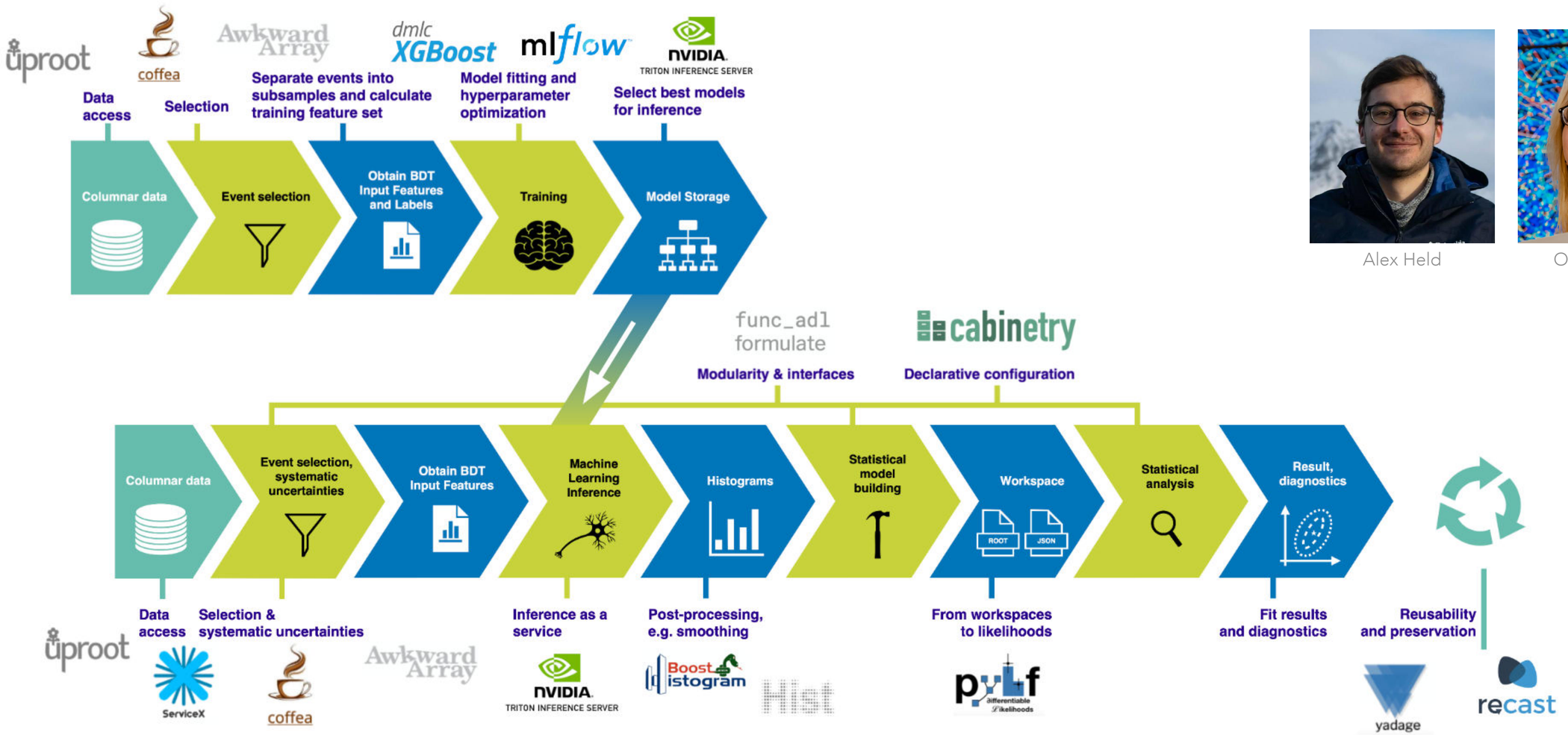
Ianna Osborne, Jim Pivarski, Jerry Ling

In the realm of scientific computing, both Julia and Python have established themselves as powerful tools. Within the context of High Energy Physics (HEP) data analysis, Python has been traditionally favored, yet there exists a compelling case for migrating legacy software to Julia. This article focuses on language interoperability, specifically exploring how Awkward Array data structures can bridge the gap between Julia and Python. The talk offers insights into key considerations such as memory management, data buffer copies, and dependency handling. It delves into the performance enhancements achieved by invoking Julia from Python and vice versa, particularly for intensive array-oriented calculations involving large-scale, though not excessively dimensional, arrays of HEP data. The advantages and challenges inherent in achieving interoperability between Julia and Python in the domain of scientific computing are discussed.



Grand Challenges

We have embraced the use of Grand Challenges as a mechanism to improve integration of the ecosystem of tools & test performance on analysis facilities.



Alex Held

Oksana Shadura

https://agc.readthedocs.io/en/latest/?badge=latest

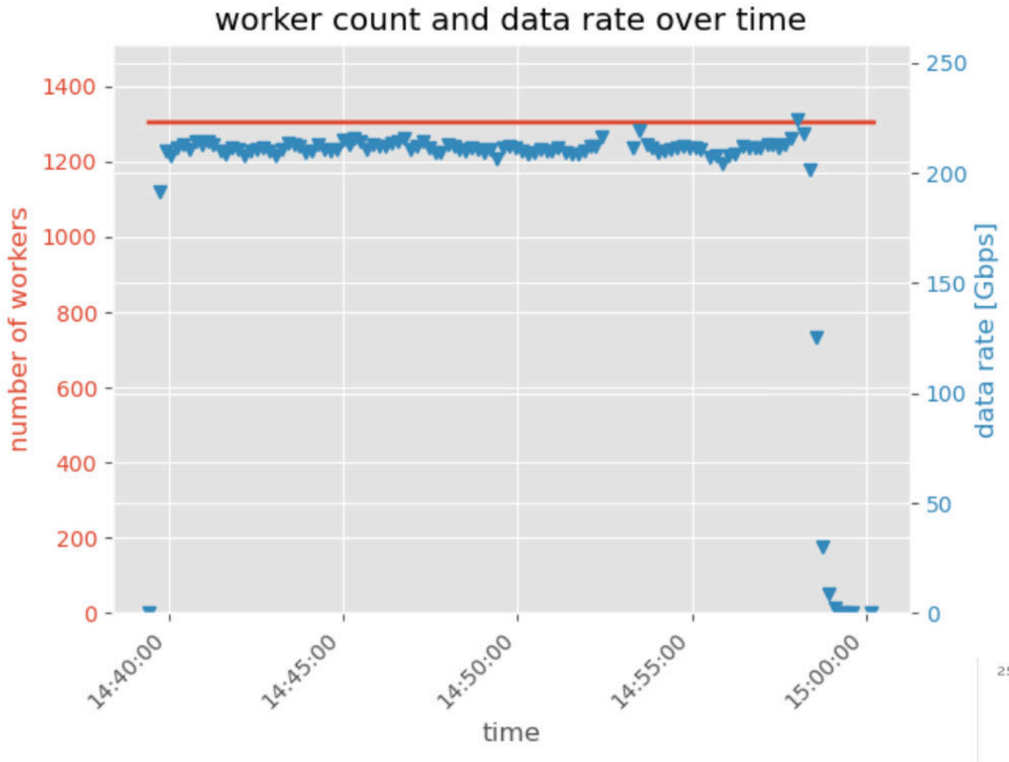
Grand Challenges

We have embraced the use of Grand Challenges as a mechanism to improve integration of the ecosystem of tools & test performance on analysis facilities.

Process 100B events in 30 min
(55MHz event processing rate)

2KB/event ⇒ 200TB (200 Gbps)

Analysis running on 2,200 cores

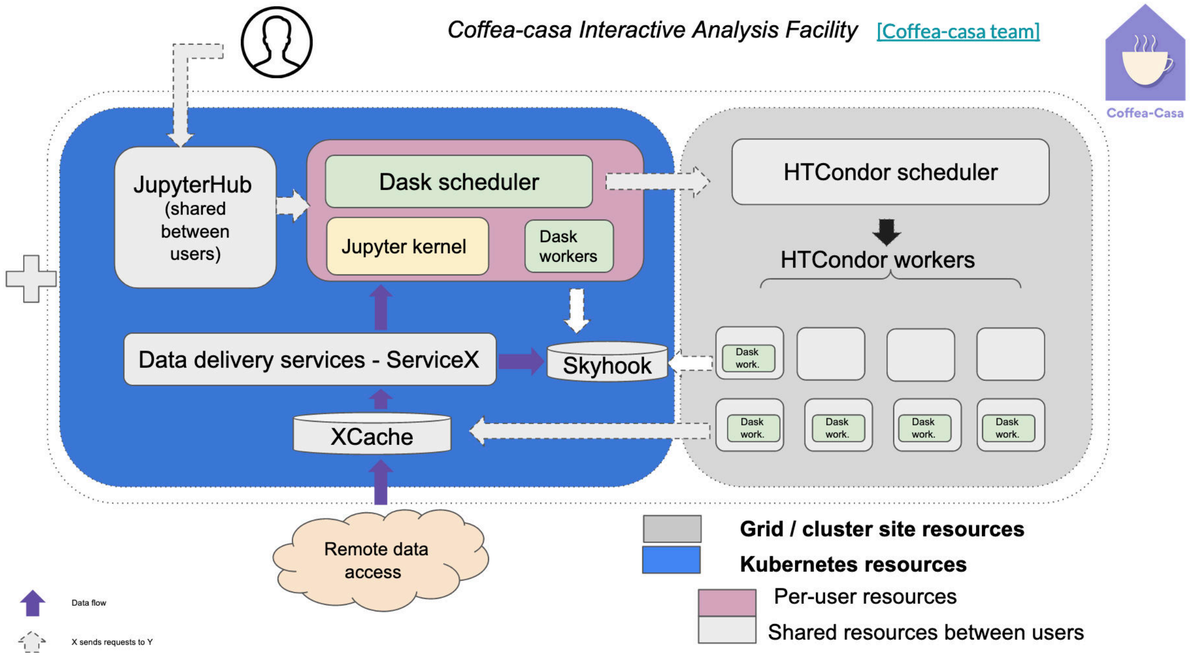


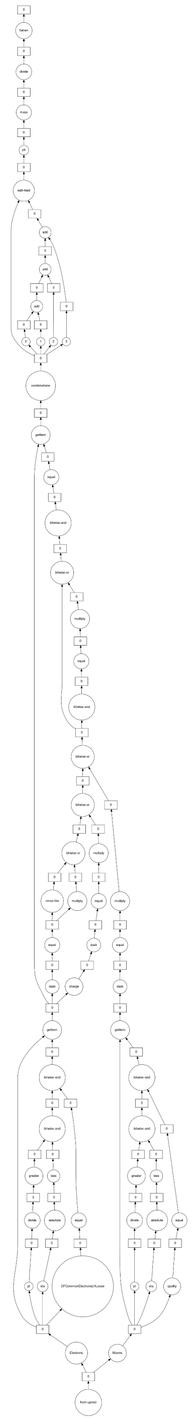
TL;DR We learned a lot about Dask

Quick shoutout

dask-awkward

Elements featured at SciPy this week by Vangelis Kourlitis in *How the Scientific Python ecosystem helps answering fundamental questions of the Universe*





Vangelis Kourlitis, How the Scientific Python ecosystem helps answering fundamental questions of the Universe github.com/ekourlit/scipy2024-ATLAS-demo

Analysis Preservation & Reuse

Reinterpretation as a Service

$$\mathcal{L}_{SM} = \underbrace{\frac{1}{4}\mathbf{W}_{\mu\nu}\cdot\mathbf{W}^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}G^{a}_{\mu\nu}G^{\mu\nu}_{a}}_{\text{kinetic energies and self-interactions of the gauge bosons}}$$

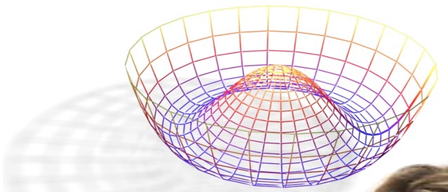
$$+ \underbrace{\bar{L}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})L + \bar{R}\gamma^{\mu}(i\partial_{\mu} - \frac{1}{2}g'YB_{\mu})R}_{\text{kinetic energies and electroweak interactions of fermions}}$$

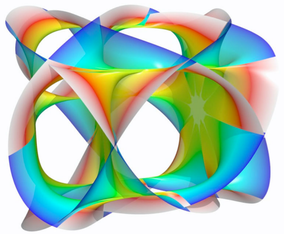
$$+ \underbrace{\frac{1}{2}\left|(i\partial_{\mu} - \frac{1}{2}g\tau\cdot\mathbf{W}_{\mu} - \frac{1}{2}g'YB_{\mu})\,\phi\right|^{2} - V(\phi)}_{W^{\pm},Z,\gamma,\text{and Higgs masses and couplings}}$$

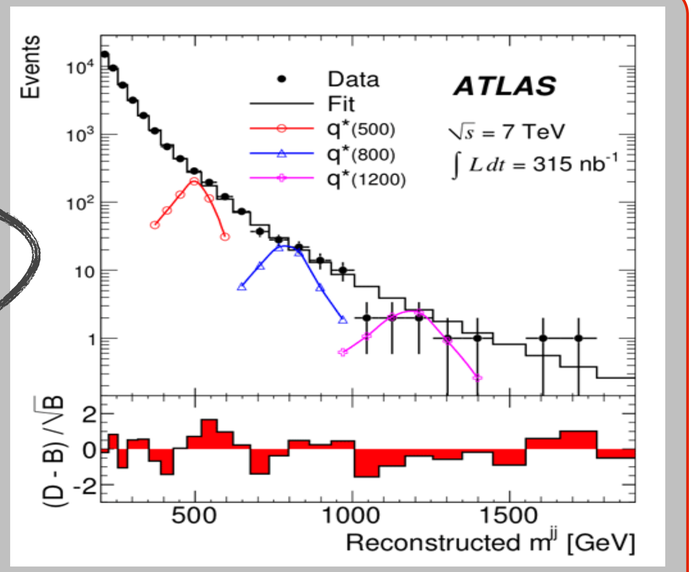
$$+ \underbrace{g''(\bar{q}\gamma^{\mu}T_{a}q)\,G^{a}_{\mu}}_{\text{interactions between quarks and gluons}} + \underbrace{(G_{1}\bar{L}\phi R + G_{2}\bar{L}\phi_{c}R + h.c.)}_{\text{fermion masses and couplings to Higgs}}$$

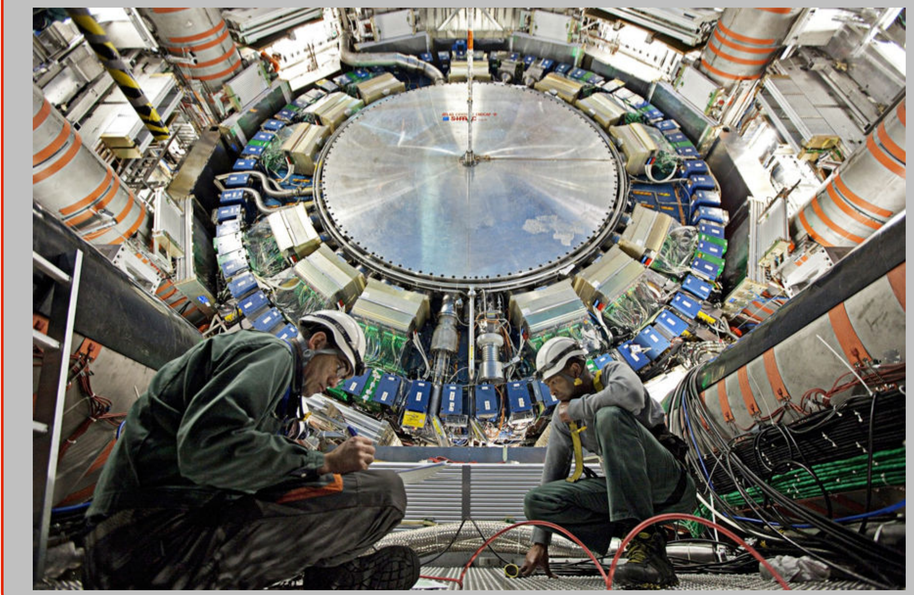
Q

A







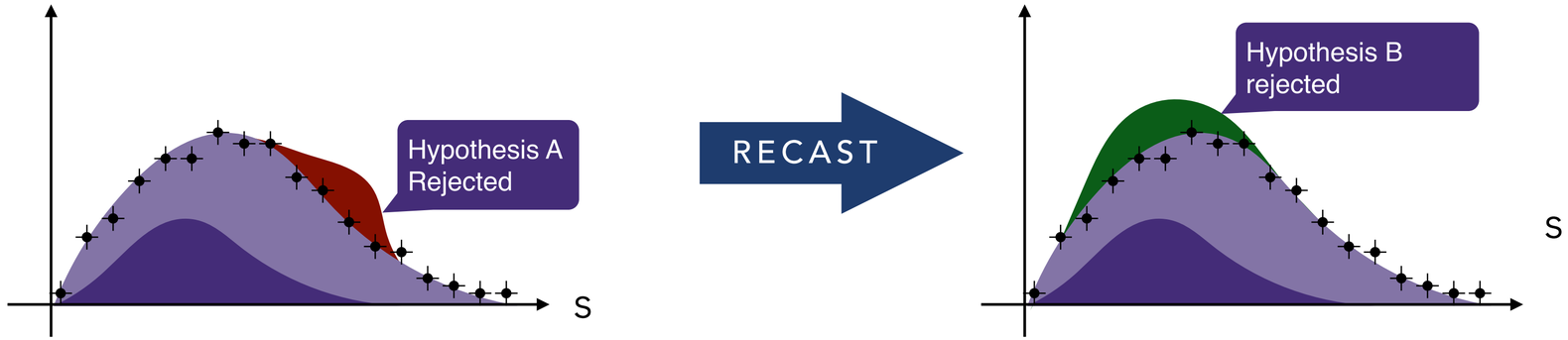


Reinterpretation infrastructure

The data analysis pipeline developed to test **Hypothesis A** can be reused

- If **Hypothesis B** predicts a similar signature in the data as **Hypothesis A**, then that same pipeline may be good enough to draw conclusions for **Hypothesis B**
  - Not optimal, but not wrong
- This depends on the details of pipeline and hypotheses. Need the code!

We have invested effort to preserve analysis pipeline so that it can be reused



observed **data +** **predicted distribution for the alternate in Hypothesis A → Hypothesis B**

recast

Open is not enough!

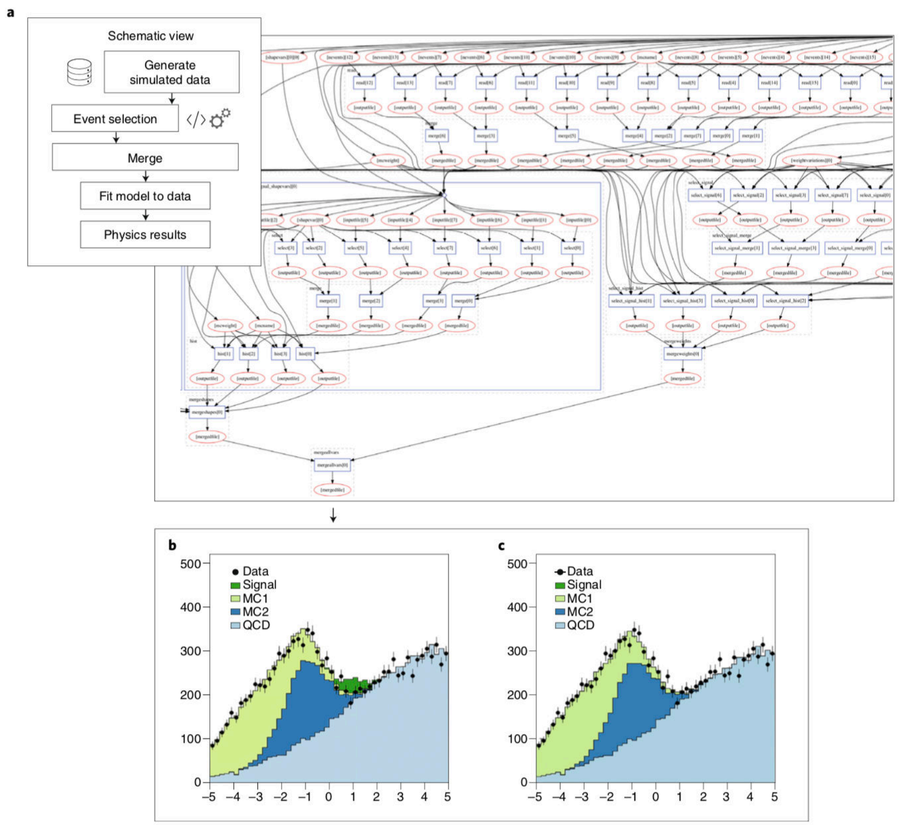


Fig. 2 | Example of a complex computational workflow on REANA mimicking a beyond the standard model (BSM) analysis . This figure shows an example where the experimental data is compared to the predictions of the standard model with an additional hypothesized signal component. The example permits one to study the complex computational workflows used in typical particle physics analyses. **a**–**c**, The computational workflow (**a**) may consist of several tens of thousands of computational steps that are massively parallelizable and run in a cascading 'map-reduce' style of computations on distributed compute clusters. The workflow definition is modelled using the Yadage workflow specification and produces an upper limit on the signal strength of the BSM process. A typical search for BSM physics consists of simulating a hypothetical signal process (**c**), as well as the background processes predicted by the standard model with properties consistent with the hypothetical signal (marked dark green in (**b**)). The background often consists of simulated background estimates (dark blue and light green histograms) and data-driven background estimates (light blue histogram). A statistical model involving both signal (dark green histogram) and background components is built and fit to the observed experimental data (black markers). **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 nominal 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 publicly available at ref. [35]. For icon credits, see Fig. 1.

reana

Reproducible research data analysis platform

Flexible

Run many computational workflow engines.

COMMON WORKFLOW LANGUAGE

Scalable

Support for remote compute clouds.

kubernetes HTCondor High Throughput Computing slurm workload manager

Reusable

Containerise once, reuse elsewhere. Cloud-native.

Free

Free Software. MIT licence. Made with ❤ at CERN.

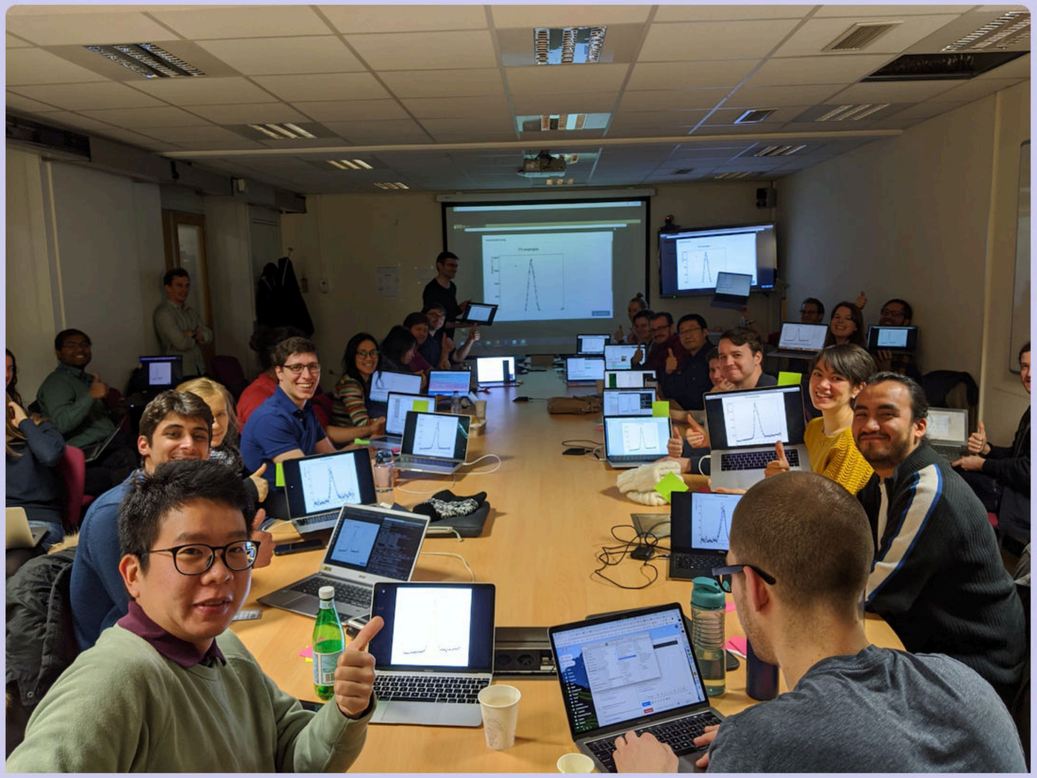
CERN

Chen, X., Dallmeier-Tiessen, S., Dasler, R. *et al.* Open is not enough. *Nature Phys* **15**, 113–119 (2019). https://doi.org/10.1038/s41567-018-0342-2

Training

Shifting the emphasis from reproducibility to reuse was key

- Train community to build containers & create reusable workflows



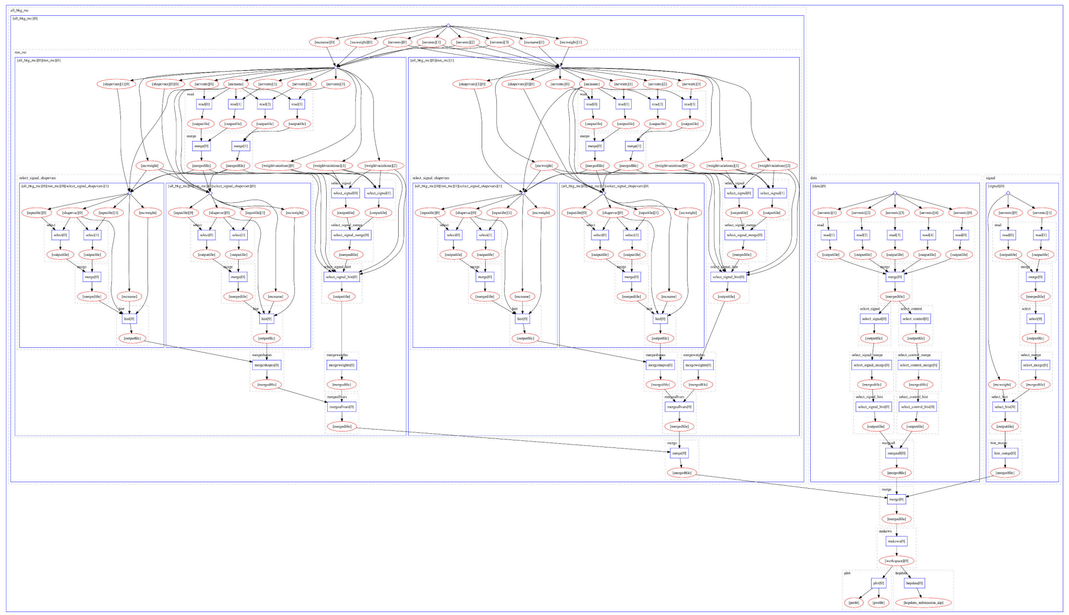
Participants in Analysis Preservation Bootcamp showing off their ability to reproduce an LHC analysis. Photo Credit: Samuel Meehan

Instructors Danika MacDonnel and Giordon Stark working with participants. Photo Credit: Samuel Meehan.

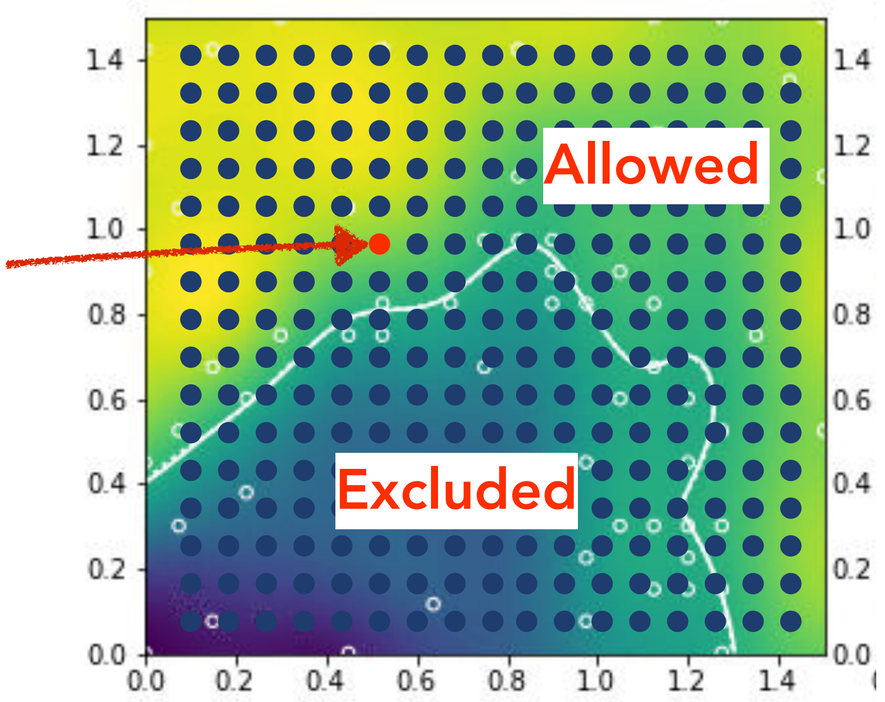
Reinterpretation with RECAST

Basic approach:

- scan parameter space of theory, simulate signal for each point
- execute complex workflow that implements analysis for each parameter point
- determine which regions of parameter space are excluded



Complex computational workflow

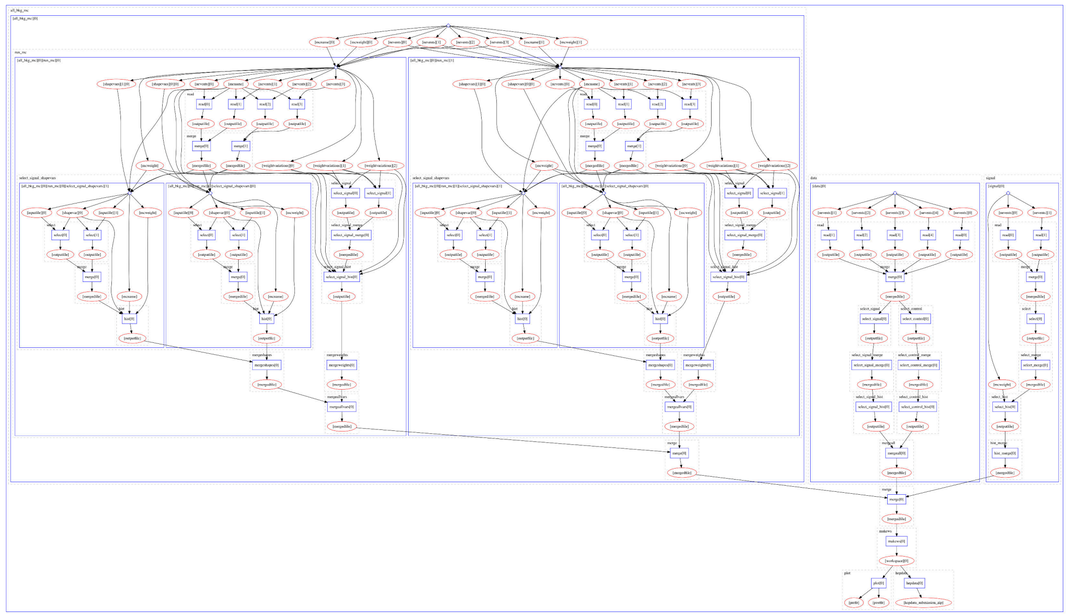


Parameter space of theory

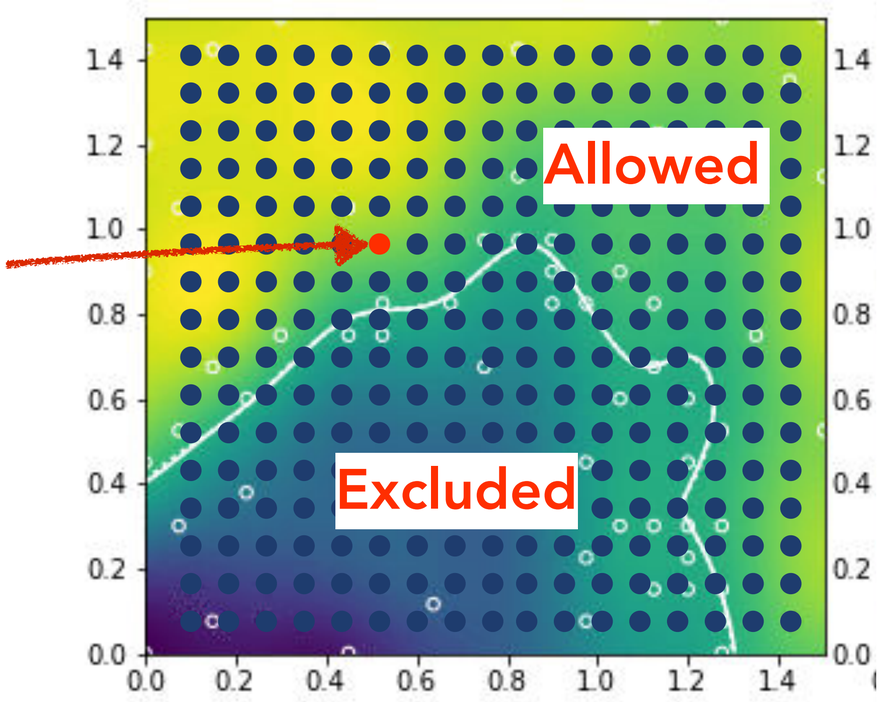
Reinterpretation with RECAST

Basic approach:

- scan parameter space of theory, simulate signal for each point
- execute complex workflow that implements analysis for each parameter point
- determine which regions of parameter space are excluded



Complex computational workflow



Parameter space of theory

Active Learning for Reinterpretation

336.Efficient search for new physics using Active Learning in the ATLAS Experiment

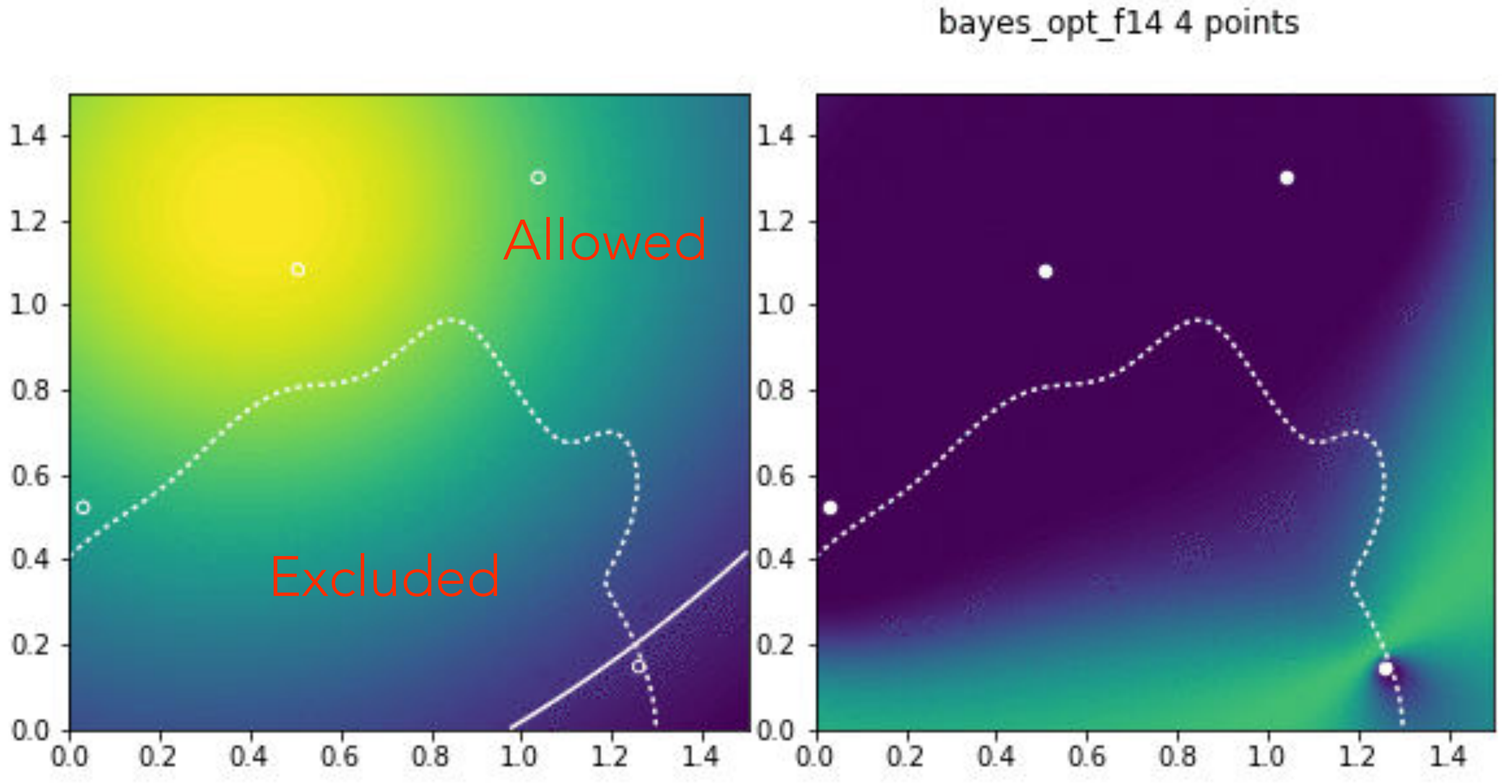
Zubair Bhatti, Kyle Cranmer, Irina Espejo, Lukas Heinrich, Phillip Gadow, Patrick Rieck and Janik von Ahne on behalf of ATLAS, ATL-PHYS-PUB-2022-045

Instead of simulating on a grid a priori, simulate on demand where it is relevant!

Drastically more efficient use of computing resources

Changes traditional relationship between production system & analysis

Active learning algorithm steers enormous computational effort



https://github.com/diana-hep/excursion

DOI 10.5281/zenodo.1634427

Active Learning for Reinterpretation

336.Efficient search for new physics using Active Learning in the ATLAS Experiment

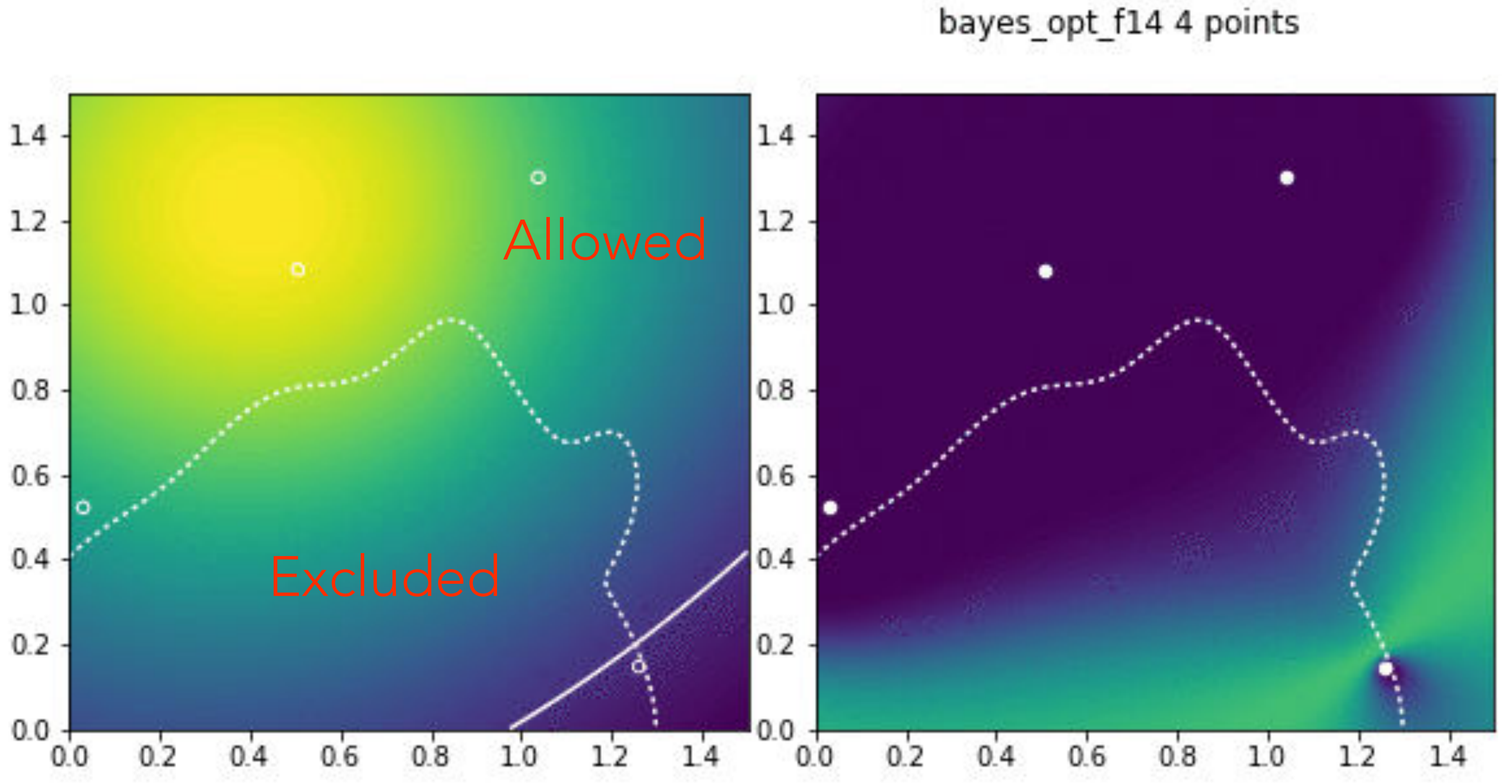
Zubair Bhatti, Kyle Cranmer, Irina Espejo, Lukas Heinrich, Phillip Gadow, Patrick Rieck and Janik von Ahne on behalf of ATLAS, ATL-PHYS-PUB-2022-045

Instead of simulating on a grid a priori, simulate on demand where it is relevant!

Drastically more efficient use of computing resources

Changes traditional relationship between production system & analysis

Active learning algorithm steers enormous computational effort



https://github.com/diana-hep/excursion

DOI 10.5281/zenodo.1634427

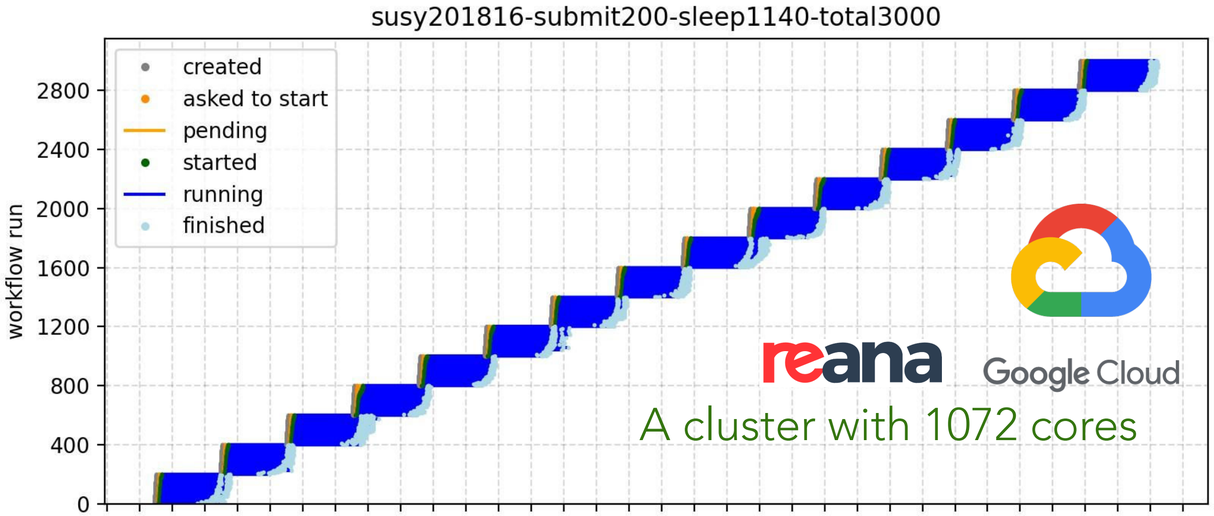
Reuse enables new science

JHEP 2024 (2024) 106
DOI: 10.1007/JHEP05(2024)106

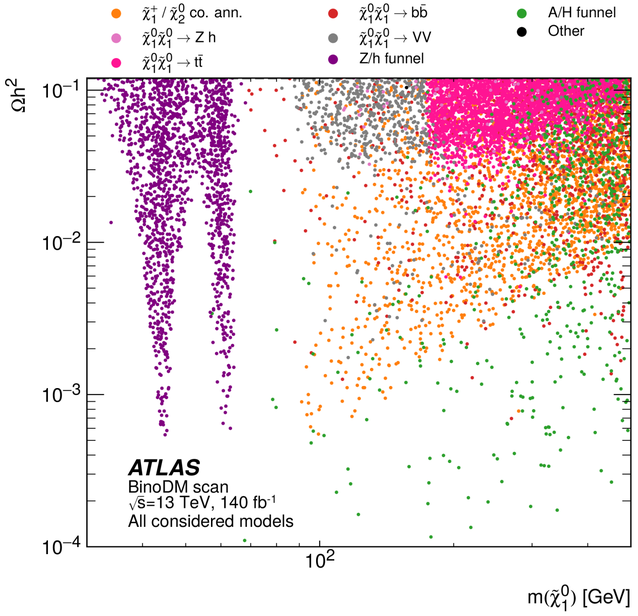
CERN-EP-2024-021
31st May 2024

Scalable ATLAS pMSSM computational workflows using containerised REANA reusable analysis platform

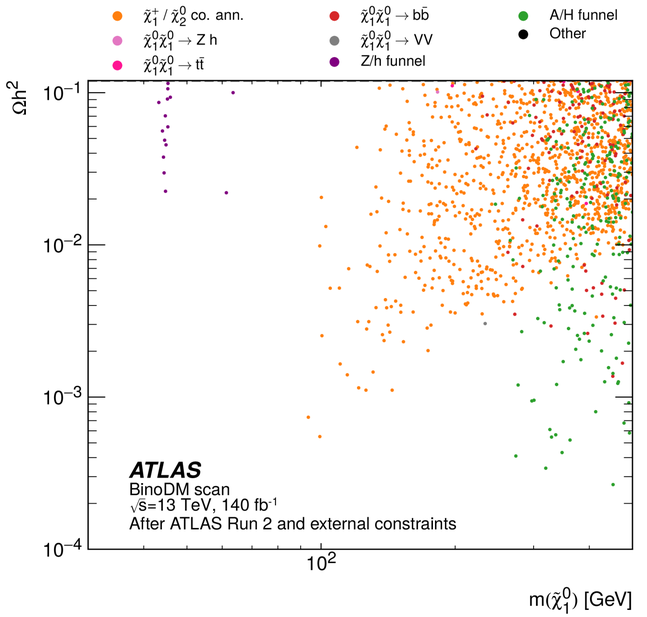
Marco Donadoni[1], *Matthew* Feickert[2], *Lukas* Heinrich[3], *Yang* Liu[4], *Audrius* Mečionis[1], *Vladyslav* Moisieienkov[1], *Tibor* Šimko[1,*], *Giordon* Stark[5], and *Marco* Vidal García[1]



ATLAS Run 2 searches for electroweak production of supersymmetric particles interpreted within the pMSSM



(a) All considered models



(d) Models not excluded by ATLAS or external constraints

*"This result was a tour de force from ATLAS's supersymmetry physics group, pulling together results from **eight separate ATLAS searches** using data collected during Run 2 of the LHC and evaluating **tens of thousands of supersymmetric models** in the **19-dimensional parameter space** to set **new, more restrictive constraints** on models with supersymmetric **dark matter** particles. " — Matthew Feickert*

arxiv:2403.03494 & arxiv:2402.01392

Do we really need another workflow language?

Probably not…

- We brought together experts from different workflow languages to understand the state of the art and needs of particle physics community

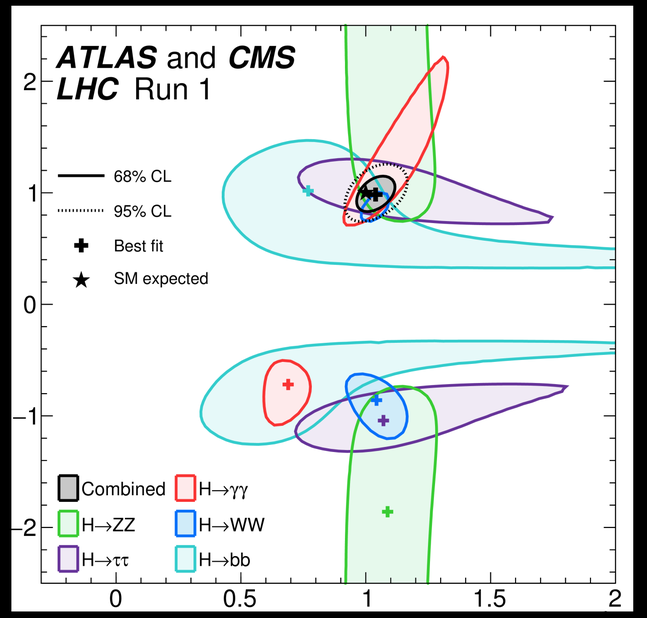
COMMON WORKFLOW LANGUAGE

Workshop on workflow languages for HEP analysis: https://indico.cern.ch/event/1380367/overview

Simulation-based Inference

High-dimensional event data

[CMS]



[ATLAS, CMS 1606.02266]

Precision constraints on new physics

Simulating particle physics processes

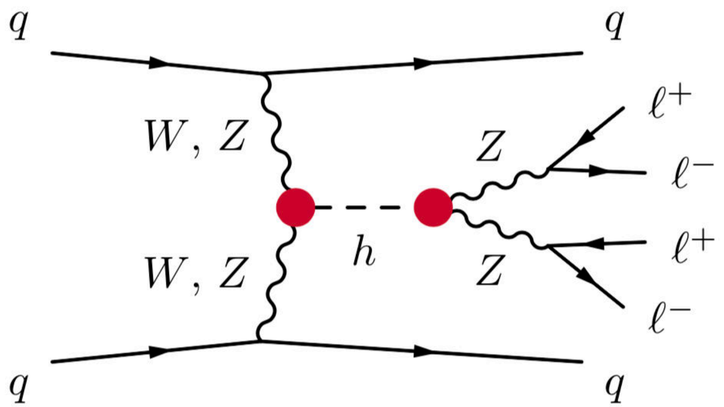
Theory
parameters
θ

Evolution

Simulating particle physics processes

Latent variables

Parton-level momenta z_p ← Theory parameters θ



Evolution

Simulating particle physics processes

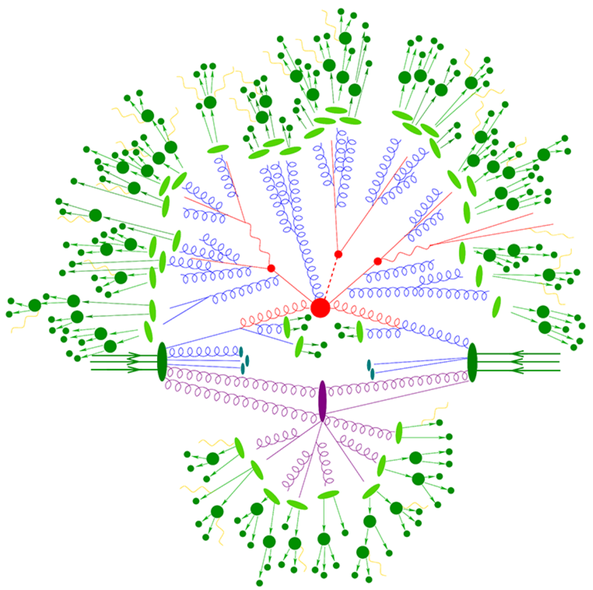
Latent variables

Shower splittings

Parton-level momenta

Theory parameters

z_s ← z_p ← θ



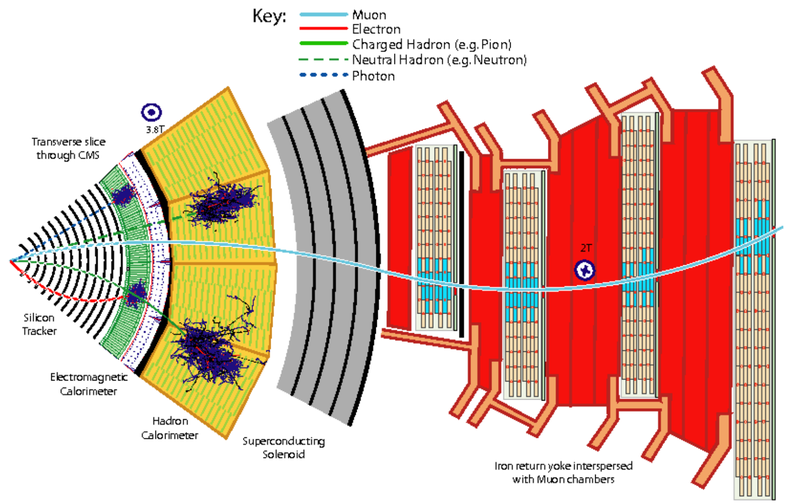
[F. Krauss]

Evolution

Simulating particle physics processes

Latent variables

Detector interactions	Shower splittings	Parton-level momenta	Theory parameters
z_d	z_s	z_p	θ



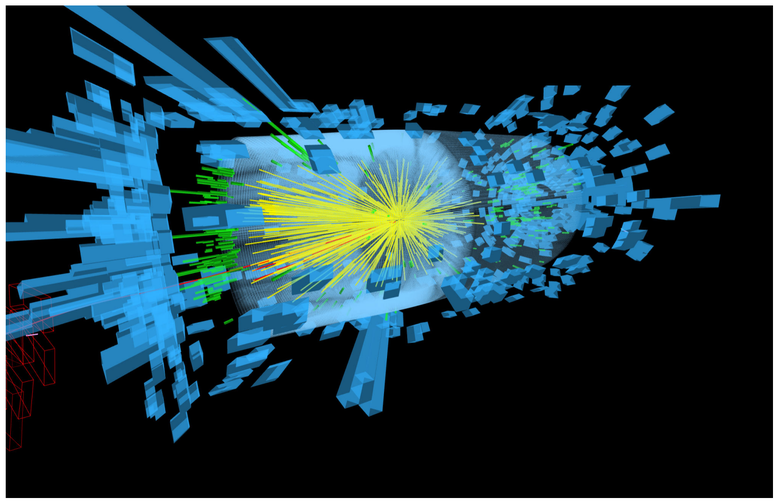
[CMS]

Evolution

Simulating particle physics processes

Latent variables

Observables	Detector interactions	Shower splittings	Parton-level momenta	Theory parameters
x	z_d	z_s	z_p	θ

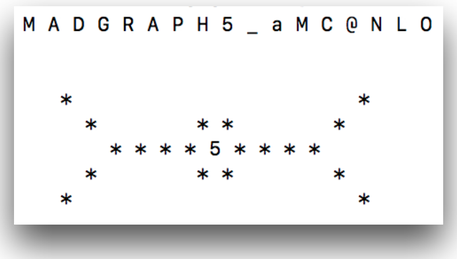


Evolution

Simulating particle physics processes

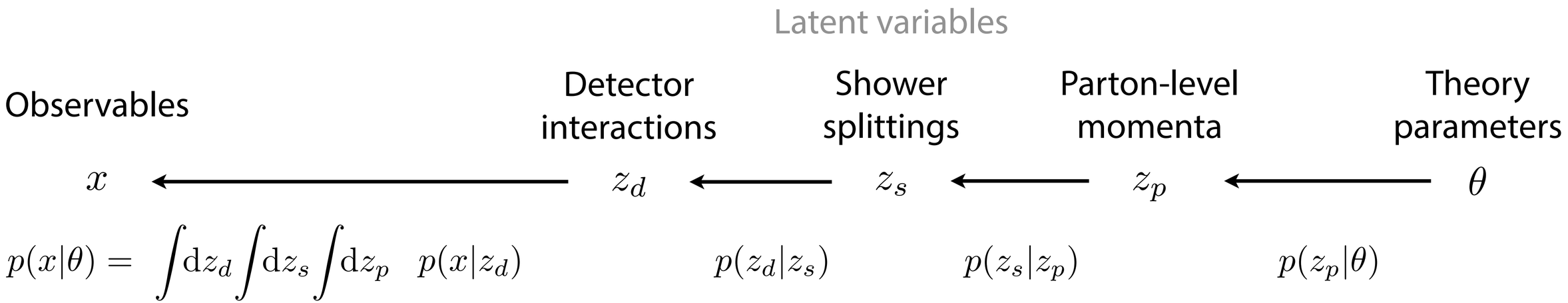
Latent variables

Observables	Detector interactions	Shower splittings	Parton-level momenta	Theory parameters
x	z_d	z_s	z_p	θ
Sample from	$p(x\|z_d)$	$p(z_d\|z_s)$	$p(z_s\|z_p)$	$p(z_p\|\theta)$



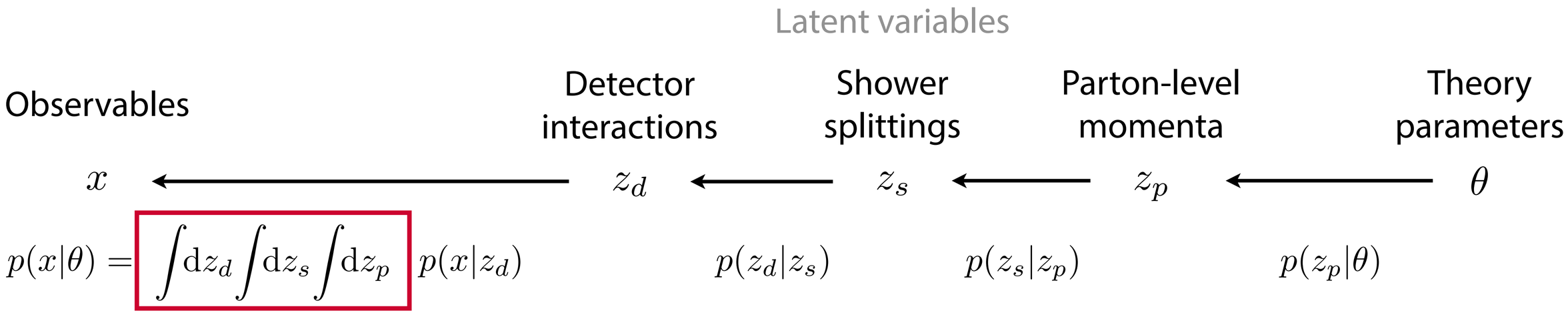
Prediction (simulation)

Simulating particle physics processes



Inference

Simulating particle physics processes



It's infeasible to calculate the integral over this enormous space!

Inference

Simulation-based Inference

This motivates a class of inference methods for a stochastic simulator where

- evaluating the **likelihood is intractable**, but
- it is **possible to sample** synthetic data $x \sim p(x \mid \theta)$

This setting is often referred to as **likelihood-free inference**, but I prefer the term **simulation-based inference** because usually one approximates the likelihood (or likelihood ratio) and then use established inference techniques

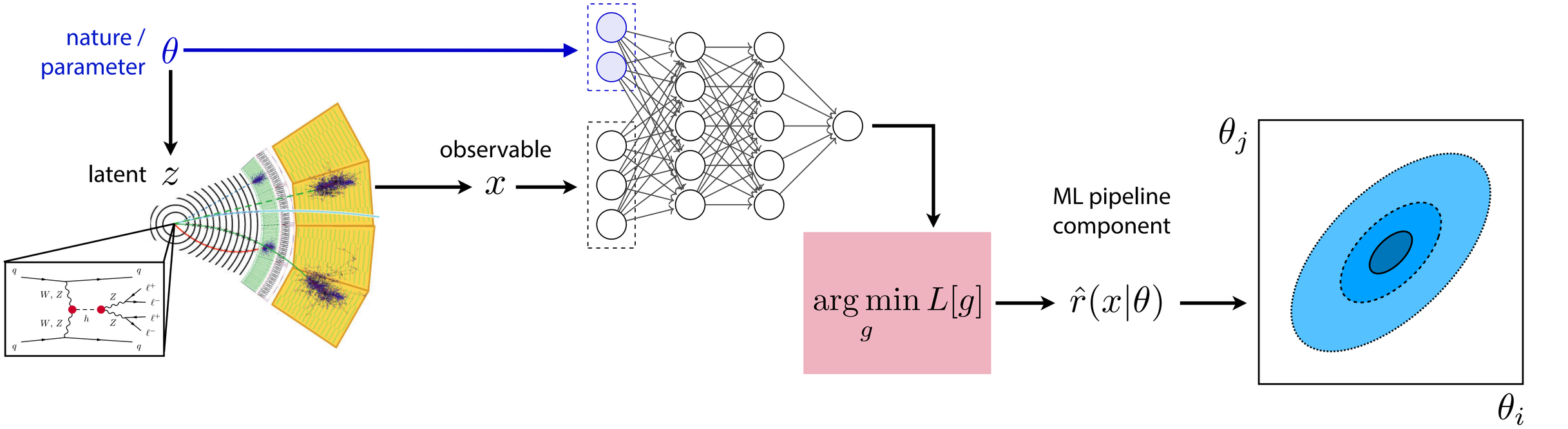
- applies to both Bayesian or Frequentist inference

[Cranmer, Brehmer, Louppe PNAS (2020), arXiv:1911.01429]

Simulation-Based Inference

Deep learning and neural density estimation are effective at learning approximate surrogates for the likelihood and posterior, **revolutionizing principled statistical inference in science!**

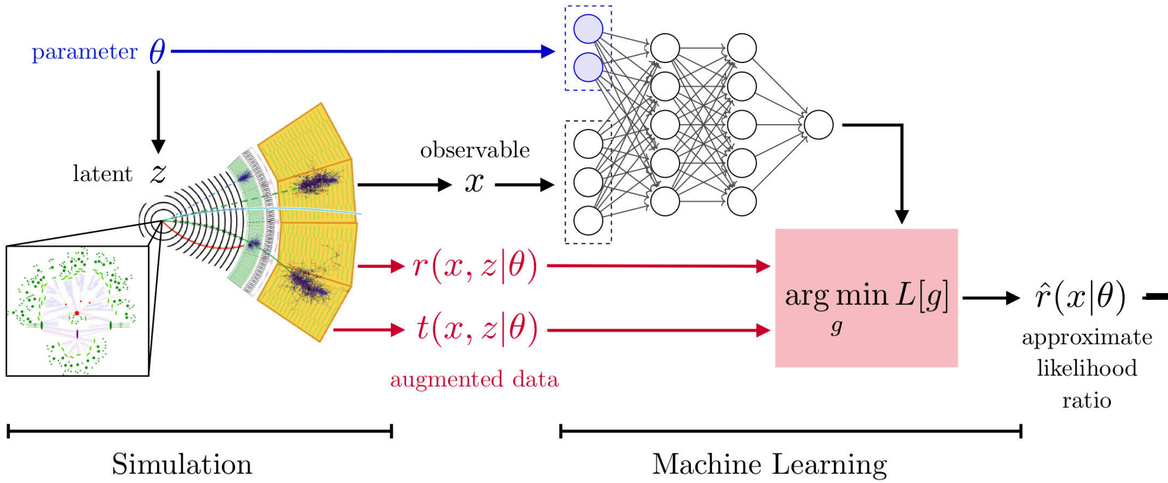
- Removes the need for hand-engineered summary statistics that sacrifice power
- Expert knowledge in simulator is transferred to surrogate via learning

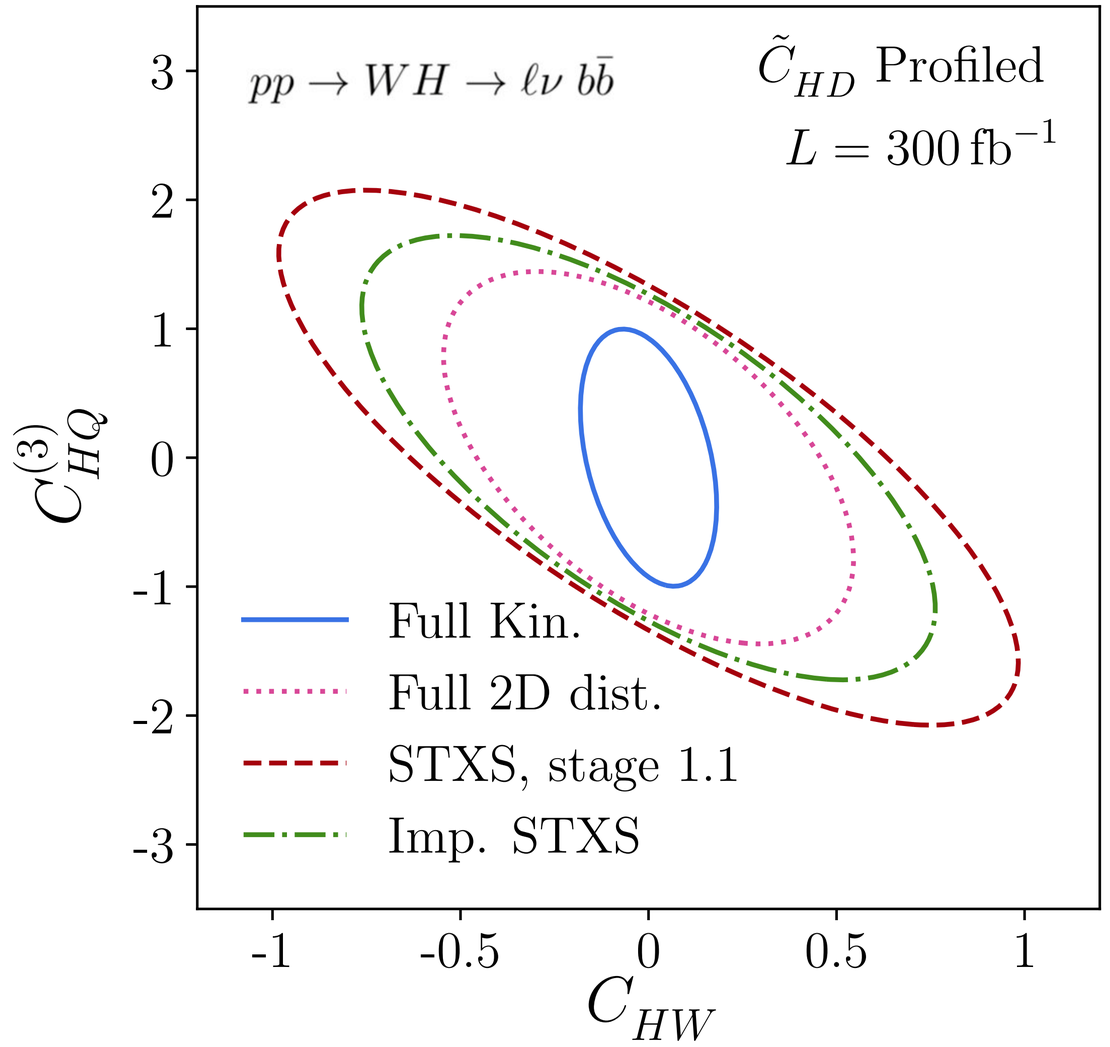


Impact on Studies of The Higgs Boson

Potential for massive gains in precision of a flagship measurement at the LHC !

Equivalent increasing data collected by LHC by several factors





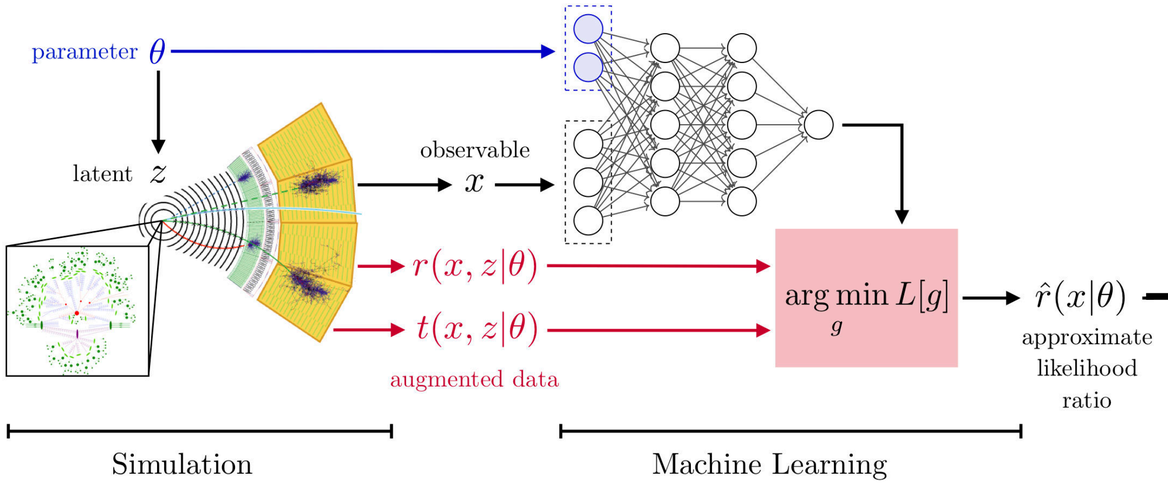
[J. Brehmer, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

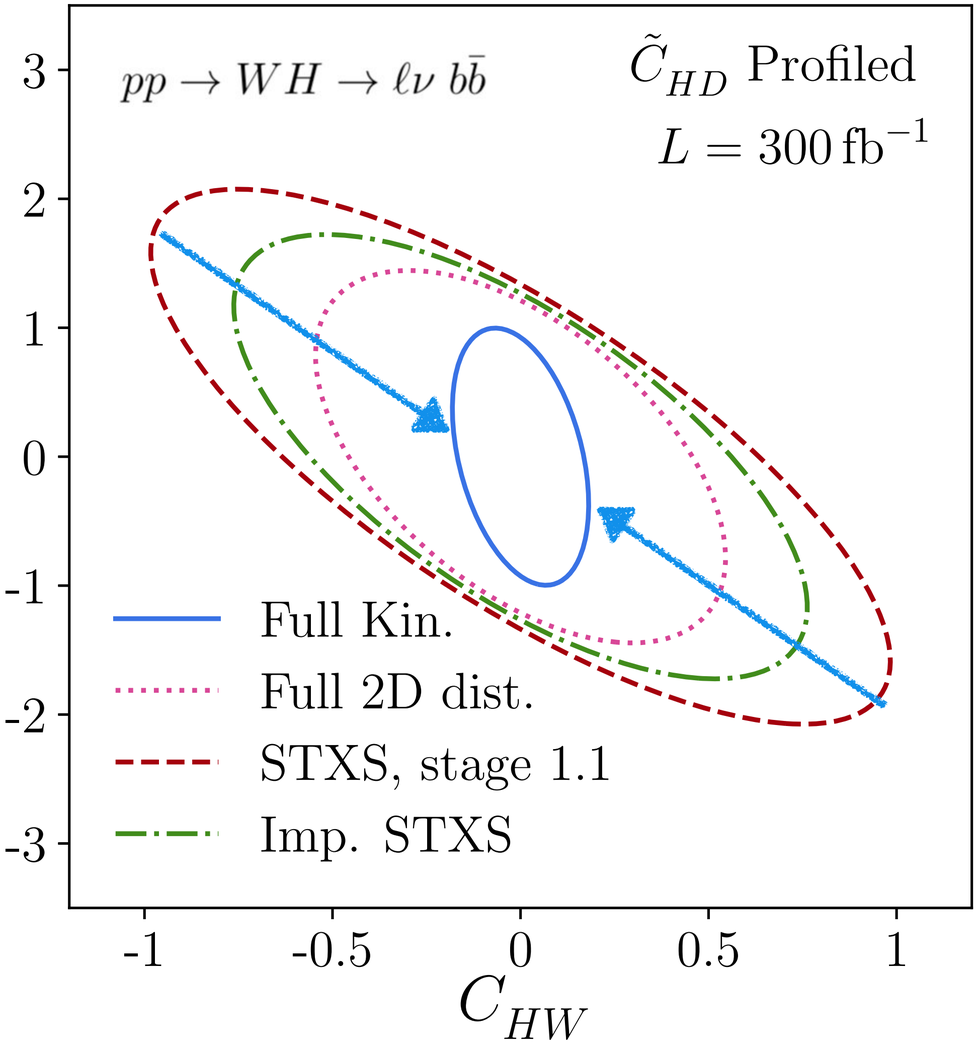
[J. Brehmer, F. Kling, I. Espejo, K. Cranmer 1907.10621]

Impact on Studies of The Higgs Boson

Potential for massive gains in precision of a flagship measurement at the LHC !

Equivalent increasing data collected by LHC by several factors



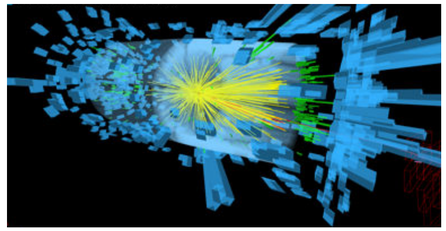


[J. Brehmer, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

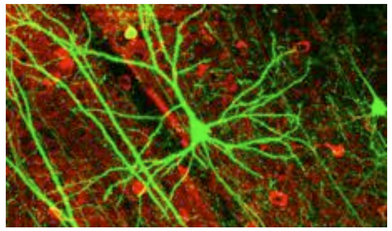
[J. Brehmer, F. Kling, I. Espejo, K. Cranmer 1907.10621]

Science is replete with high-fidelity simulators

Simulation-based inference is a major evolution in statistical inference for science!



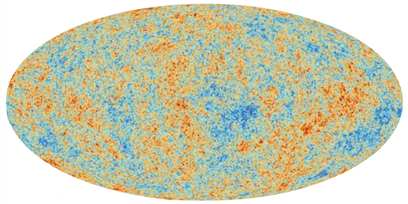
Particle colliders



Neuron activity

Epidemics

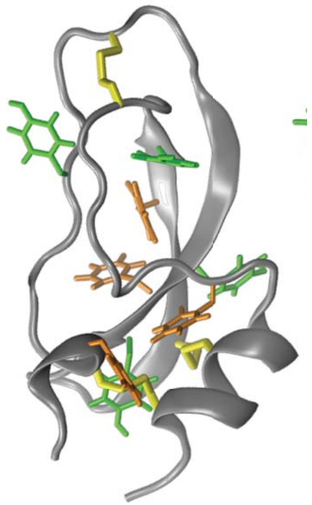
Gravitational lensing

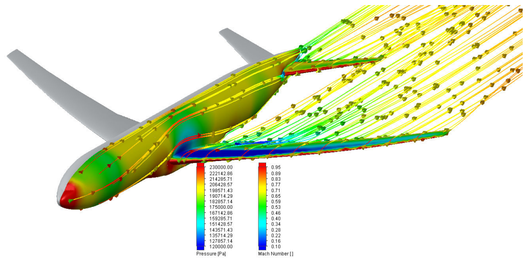


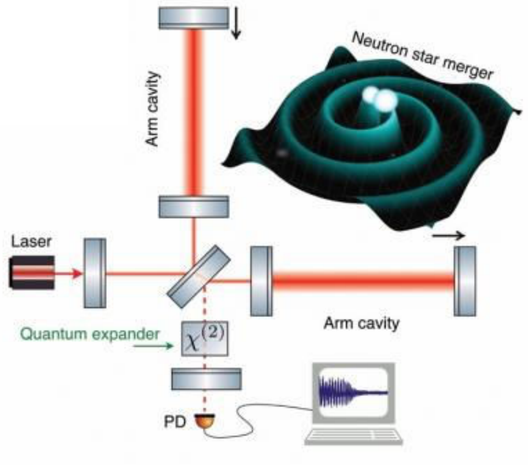
Evolution of the Universe

10^{-18} 10^{-15} 10^{-12} 10^{-9} 10^{-6} 10^{-3} 10^{0} 10^{3} 10^{6} 10^{9} 10^{12} 10^{15} 10^{18} 10^{21} 10^{24} 10^{27}

Length scale [m]







[Cranmer, Brehmer, Louppe PNAS (2020), arXiv:1911.01429]

simulation-based-inference.org

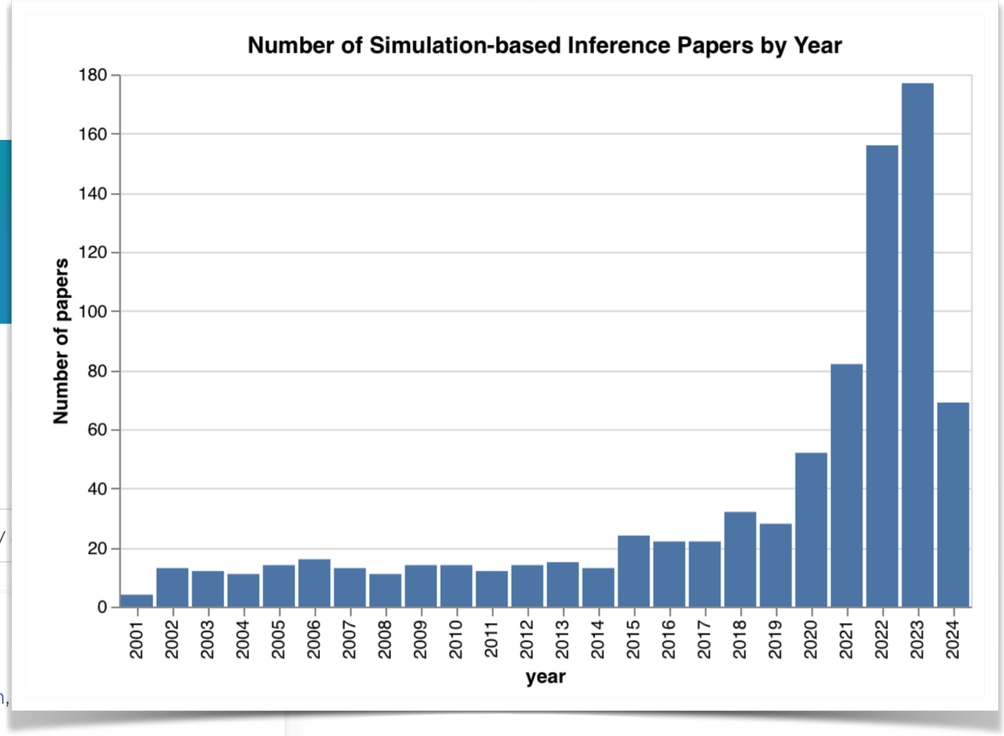
Papers

The list is automatically compiled each day. Should you observe any inaccuracies or concerns, kindly bring them to our attention.
Additionally, if you believe a new paper aligns with the topic, feel free to submit it.
Visualize the annual growth in the number of publications.

Sort by Category | Sort by Year | Sort by

Statistics

- Addressing Misspecification in Simulation-based Inference through Data-driven Calibration, Gamella, O Sener, J Behrmann... - arXiv preprint arXiv ..., 2024 - arxiv.org
- Modelling Sampling Distributions of Test Statistics with Autograd, AA Kadhim, HB Prosper - arXiv preprint arXiv:2405.02488, 2024 - arxiv.org
- Preconditioned Neural Posterior Estimation for Likelihood-free Inference, X Wang, RP Kelly, DJ Warne, C Drovandi - arXiv preprint arXiv ..., 2024 - arxiv.org
- A variational neural Bayes framework for inference on intractable posterior distributions, E Maceda, EC Hector, A Lenzi, BJ Reich - arXiv preprint arXiv:2404.10899, 2024 - arxiv.org
- Increased perceptual reliability reduces membrane potential variability in cortical neurons, B von Hünerbein, J Jordan, M Oude Lohuis... - bioRxiv, 2024 - biorxiv.org
- How much information can be extracted from galaxy clustering at the field level?, NM Nguyen, F Schmidt, B Tucci, M Reinecke... - arXiv preprint arXiv ..., 2024 - arxiv.org
- Evolution of Analysis Techniques and Statistical Treatment, A Held - Bulletin of the American Physical Society, 2024 - APS
- Simulation-Based Inference with Quantile Regression, H Jia - arXiv preprint arXiv:2401.02413, 2024 - arxiv.org
- Direct Amortized Likelihood Ratio Estimation, AD Cobb, B Matejek, D Elenius, A Roy... - arXiv preprint arXiv ..., 2023 - arxiv.org
- On simulation-based inference for implicitly defined models, J Park - arXiv preprint arXiv:2311.09446, 2023 - arxiv.org
- Machine Learning for Mechanistic Models of Metapopulation Dynamics, J Li, EL Ionides, AA King, M Pascual, N Ning - arXiv preprint arXiv ..., 2023 - arxiv.org
- Inference on spatiotemporal dynamics for networks of biological populations, J Li, EL Ionides, AA King, M Pascual, N Ning - arXiv preprint arXiv ..., 2023 - arxiv.org
- Optimal simulation-based Bayesian decisions, J Alsing, TDP Edwards, B Wandelt - arXiv preprint arXiv:2311.05742, 2023 - arxiv.org



Display a menu

Simulation-based Inference software packages

JOSS
The Journal of Open Source Software

sbi: A toolkit for simulation-based inference

Alvaro Tejero-Cantero[e, 1], **Jan Boelts**[e, 1], **Michael Deistler**[e, 1], **Jan-Matthis Lueckmann**[e, 1], **Conor Durkan**[e, 2], **Pedro J. Gonçalves**[1, 3], **David S. Greenberg**[1, 4], **and Jakob H. Macke**[1, 5, 6]

e Equally contributing authors **1** Computational Neuroengineering, Department of Electrical and Computer Engineering, Technical University of Munich **2** School of Informatics, University of Edinburgh **3** Neural Systems Analysis, Center of Advanced European Studies and Research (caesar), Bonn **4** Model-Driven Machine Learning, Centre for Materials and Coastal Research, Helmholtz-Zentrum Geesthacht **5** Machine Learning in Science, University of Tübingen **6** Empirical Inference, Max Planck Institute for Intelligent Systems, Tübingen

DOI: 10.21105/joss.02505

Software

- Review
- Repository
- Archive

Editor: Dan Foreman-Mackey
Reviewers:

- @EiffL
- @cranmer

Submitted: 14 July 2020
Published: 21 August 2020

Summary

Scientists and engineers employ stochastic numerical simulators to model empirically observed phenomena. In contrast to purely statistical models, simulators express scientific principles that provide powerful inductive biases, improve generalization to new data or scenarios and allow for fewer, more interpretable and domain-relevant parameters. Despite these advantages, tuning a simulator's parameters so that its outputs match data is challenging. Simulation-based inference (SBI) seeks to identify parameter sets that a) are compatible with prior knowledge and b) match empirical observations. Importantly, SBI does not seek to recover a single 'best' data-compatible parameter set, but rather to identify all high probability regions of parameter space that explain observed data, and thereby to quantify parameter uncertainty. In Bayesian terminology, SBI aims to retrieve the posterior distribution over the parameters of interest. In

MadMiner: Machine learning–based inference for particle physics

By Johann Brehmer, Felix Kling, Irina Espejo, and Kyle Cranmer

pypi package 0.6.3 | build passing | docs failing | chat on gitter | code style black | License MIT | DOI 10.5281/zenodo.1489147 | arXiv 1907.10621

BayesFlow

Tests passing | License MIT | JOSS 10.21105/joss.05702 | contributions welcome

Welcome to our BayesFlow library for efficient simulation-based Bayesian workflows! Our library enables users to create specialized neural networks for *amortized Bayesian inference*, which repay users with rapid statistical inference after a potentially longer simulation-based training phase.

Swyft

Swyft is a system for scientific simulation-based inference at scale.

pypi package 0.4.5 | codecov 26% | JOSS 10.21105/joss.04205 | DOI 10.5281/zenodo.5752734

Swyft is the official implementation of Truncated Marginal Neural Ratio Estimation (TMNRE), a hyper-efficient, simulation-based inference technique for complex data and expensive simulators.

Wrap it up

The next 10 years

https://www.usparticlephysics.org/

6.7

Software, Computing, and Data Science

Software and computing play a critical role in maximizing the science output of particle physics experiments. They are an integral component of experimental design, trigger and data acquisition, simulation, reconstruction, and data analysis. They also underlie simulation and design of particle accelerators. The complexity, computational needs and data volumes of particle physics experiments are expected to increase dramatically in the next decade or so. Advances in software and computing, including AI/ML, will be key for

6.8

Technology Innovations and Impact on Society

Particle physicists have a long history of recognizing, embracing, and fostering emerging technologies. The devices and tools that particle physics develops enable applications beyond fundamental physics in fields as different as medicine and aerospace. This section outlines the many ways particle physics spurs innovation, with example impacts.

6.8.1 – Computing

Particle physicists are the stewards of some of the world's largest datasets and therefore have a special connection to computing and data applications. Particle physicists were among the first to transition from analog to digitized data systems and were early adopters of the use of machine learning algorithms. The world wide web and the web browser were invented at CERN as an efficient information sharing system for particle physicists. To cope with enormous datasets, particle physicists played leading roles in the development of global-scale distributed computing technologies and high-performance networking. The ambitious portfolio of projects proposed for the next 10 years will drive innovation that addresses the scientific needs of the particle physics community and will broadly impact society.

6.8.2 – Artificial Intelligence and Machine Learning

The unique challenges encountered in particle physics have proven to be fertile ground for innovation of AI/ML. Importing new techniques into the particle physics context requires a process of translation, through which those techniques are stress-tested, generalized, and improved. Those improvements feed back into the wider AI/ML research community, completing a cycle of use-inspired research.

Thank you!

Questions?

“New directions in science are launched by new tools much more often than by new concepts. The effect of a concept-driven revolution is to explain old things in new ways. The effect of a tool-driven revolution is to discover new things that have to be explained.”

– FREEMAN DYSON