Summary on Deep Dive: ML Models Publication

Jennifer Ngadiuba (Fermilab), Javier Duarte (UCSD)

CMS Week Physics Plenaries
February 7, 2024
CERN

Where is (or will be) ML used in CMS?

Front-end electronics	◉ Fast ML on ASICs for data compression in Phase 2 HGCAL
Trigger	◉ Fast ML on FPGAs for Run 3 & Phase 2 L1 trigger and 40 MHz scouting
DQM	◉ Automated data certification ◉ Online anomaly detection (ECAL, HCAL, muon system)
Simulation	◉ Calorimeter simulation with generative models
Reconstruction	◉ Energy and mass regression (e.g., MET, photons, electrons, jets) ◉ PU mitigation ◉ Clustering (e.g., calorimeter, jets, vertexing) ◉ Particle flow
Analysis / object ID	◉ Tau leptons, heavy flavour / boosted / displace jets tagging ◉ Event classification ◉ Background estimation ◉ Uncertainties evaluation

A growing number of ML technical papers

Code	Name
MLG-23-001 » MC | show | CDS | CSBS	Portable Acceleration of CMS Mini-AOD Production with Coprocessor ...
MLG-23-002 » MC | show	ML techniques for identification of anomalous jets
MLG-23-003 » MC | show	ABCD background estimation method with ML for a Stealth and RPV t ...
MLG-23-004 » MC | show	Neural Autoregressive Flows for Data-driven background estimation ...
MLG-23-005 » MC | show	ML aspects of ML-based STXS measurements in Higgs to tau tau

Most are companion to a physics results paper indicating that the **complexity and innovation in the ML technical work** behind cannot be simply summarized anymore with a citation of external work

Why publish ML models?

To **preserve**, **reuse**, and **reinterpret** our results!

Executive summary. To achieve their full scientific impact, HEP experiments need to integrate extensive data and analysis preservation efforts into their publication processes, alongside the communication of results in reusable form and preservation of data products, and making event-level data publicly available. Without this, the influence of the hundreds of published analyses from the LHC, HL-LHC, EIC, and other future experiments will be limited mainly to the physics ideas in vogue at the time the collaboration collected their data. The public investment in experimental programs underscores the importance of going beyond the original paper publication and ensuring that analyses continue providing scientific value in perpetuity.

Snowmass '21: Data and Analysis Preservation,
Recasting, and Reinterpretation
arXiv:2203.10057

Reinterpretation: concept and workflow

Reinterpretation consists of recoding a published analysis from scratch for the purpose of interpreting it in terms of other physics models not interpreted in the original publication.

Usually done by phenomenologists to test their favorite physics model.

- An analysis code is rewritten outside the original experimental frameworks.
  - Analysis recoding is done less accurately compared to the original experimental code, since detector information does not exist in full detail in public simulation tools.
  - Public tools exist (by phenomenologists, experimentalists, also by ATLAS).
- Monte Carlo simulated events for the signal models are produced.
- Analysis code is run on the events to obtain predicted signal counts / efficiencies.
- Predicted signal counts are used together with observed data and background estimation results from the experimental publication to calculate limits.

IMPORTANT: Must validate the analysis by reproducing the experimental interpretation.
e.g. try to obtain cutflows, limits consistent with those given in the paper.

From Sezen's talk at Deep Dive

Challenges of publishing ML models

An analysis that uses ML heavily is much less accessible than a traditional cut & count analysis based on human-engineered features

- Providing sufficient model metadata (e.g. input preprocessing, etc.)
- Sharing models on an appropriate platform to enable discoverability (e.g. Huggingface, Zenodo, HEPData, etc.) + cross-referencing CMS analysis/publication
- Ensuring the model is in a persistent format (e.g. that can be read by future versions of software)
- Ensuring reusability of the model in simplified public simulation (e.g. Delphes)
- Adhering to Findable, Accessible, Interoperable, and Reusable (FAIR) Principles

Deep Dive on ML models publication

https://indico.cern.ch/event/1355548/

Morning (non CMS)

- 10:00 **Introduction**
  Speakers: Javier Mauricio Duarte (Univ. of California San
  GMT20240129-090... | Introduction
- 10:10 **Discussion**
- 10:20 **Les Houches recommendations summary**
  Speaker: Sezen Sekmen (Kyungpook National University (KI
  LesHouches.mp4 | SekmenLHML2301...
- 10:50 **Discussion**
- 11:00 **Surrogate model**
  Speaker: Sebastian Guido Bieringer (Hamburg University
  SurClass.pdf | SurrogateModel.mp4
- 11:20 **Discussion**
- 11:30 **ML model release in ATLAS**
  Speaker: Lukas Alexander Heinrich (Technische Universit
  ATLAS.mp4 | ML_Publish_Models...
- 11:50 **Discussion**

Afternoon (CMS)

- 13:00 **Flash Sim**
  Speaker: Andrea Rizzi (Universita & INFN Pisa (IT))
  FlashSim.mp4 | slides
- 13:20 **Discussion**
- 13:30 **Inputs from BTV**
  Speaker: Ming-Yan Lee (Rheinisch Westfaelische Tech. Hoch. (DE)
  BTV ML.pdf | BTV.mp4
- 13:40 **Discussion**
- 13:50 **Inputs from JME**
  Speaker: Anna Benecke (Universite Catholique de Louvain (UCL) (E
  20240129_JME_Inp... | JME.mp4
- 14:00 **Discussion**
- 14:10 **Inputs from cross-POG**
  Speaker: Dr Jean-Roch Vlimant (California Institute of Technolog
  xPOG_MLDeepDive_... | XPOG.mp4
- 14:20 **Discussion**
- 14:30 **Statistical results publication in CMS**
  Speaker: Piergiulio Lenzi (Universita e INFN, Firenze (IT))
  CAT_DeepDive_MLp... | CAT.mp4
- 14:50 **Discussion**

Afternoon (CMS)

- 15:30 **EXO-22-026: Searching for new physics detecting anomalies in jets**
  Speaker: Oz Amram (Fermi National Accelerator Lab. (US))
  CASE_reinterpretati... | GMT20240129-143...
- 15:45 **Discussion**
- 15:55 **EXO-22-015: Search for Emerging Jets with full Run 2 data**
  Speaker: Yi-Mu Chen (University of Maryland (US))
  EMJ_YiMu_2024-Ja... | GMT20240129-150...
- 16:10 **Discussion**
- 16:20 **SUS-23-001: Search for Stealth/RPV stops using Double DisCo neural network method**
  Speaker: Joshua Hiltbrand (Baylor University (US))
  GMT20240129-151... | MLG23003_202401...
- 16:35 **Discussion**
- 16:45 **EXO-22-020: Search for new physics with at least one displaced vertex and missing energy**
  Speaker: Ang Li (Austrian Academy of Sciences (AT))
  DeepDive_AngLi.pdf | GMT20240129-152...
- 17:00 **Discussion**
- 17:10 **FAIR AI Models and FAIR4HEP**
  Speaker: Dr Eliu Huerta
  FAIR_CERN.pdf | FAIR_CERN.pptx | GMT20240129-154...
- 17:30 **Wrap up**

Recommendations from Les Houches

Les Houches guide to reusable ML models in LHC analyses

Jack Y. Araz[1], *Andy Buckley*[2], *Gregor Kasieczka*[3], *Jan Kieseler*[4], *Sabine Kraml*[5], *Anders Kvellestad*[6], *Andre Lessa*[7], *Tomasz Procter*[2], *Are Raklev*[6], *Humberto Reyes-Gonzalez*[8,9,10], *Krzysztof Rolbiecki*[11], *Sezen Sekmen*[12], *Gokhan Unel*[13]

Abstract
With the increasing usage of machine-learning in high-energy physics analyses, the publication of the trained models in a reusable form has become a crucial question for analysis preservation and reuse. The complexity of these models creates practical issues for both reporting them accurately and for ensuring the stability of their behaviours in different environments and over extended timescales. In this note we discuss the current state of affairs, highlighting specific practical issues and focusing on the most promising technical and strategic approaches to ensure trustworthy analysis-preservation. This material originated from discussions in the LHC Reinterpretation Forum and the 2023 PhysTeV workshop at Les Houches.

Keywords
BSM; Tools; Machine-learning; Reinterpretation.

https://arxiv.org/abs/2312.14575

ML model design: checklist

For BDTs: petrify-bdt is a effort from within HEP to provide a more dependency-free way of preserving and executing BDTs:
read framework-specific BDT models (e.g. TMVA XML) and output standard-library C++ and/or Python code

- *Use machine-learning software that can be easily converted to a stable interchange format supported by open-source tools.*
  - *The ONNX and LWTNN JSON formats are the current most stable options for NNs.*
- *Alternatively, if possible, export the ML model to executable code without dependencies beyond standard libraries.*
- *Preserved networks should be runnable with as few dependencies as possible from an API to a compiled language (e.g. C++), not just from Python.*
- *Avoid over-complexity in network design, e.g. not using customised layers or custom activation functions if the application does not require them. Ensure the chosen architecture has sufficient preservation-format support, particularly with ONNX.*
- *Where possible, and especially if the model is dominated by simple kinematic inputs, avoid input features that are heavily dependent on detector and reconstruction details.*
- *Where inputs are heavily detector-based, in addition to preserving the ML model itself, provide detailed efficiency maps (including mistag rates) or an equivalent surrogate network using less detector-sensitive input features (see Section 5).*

16

From Sezen's talk at Deep Dive

Implementation & Validation: checklist

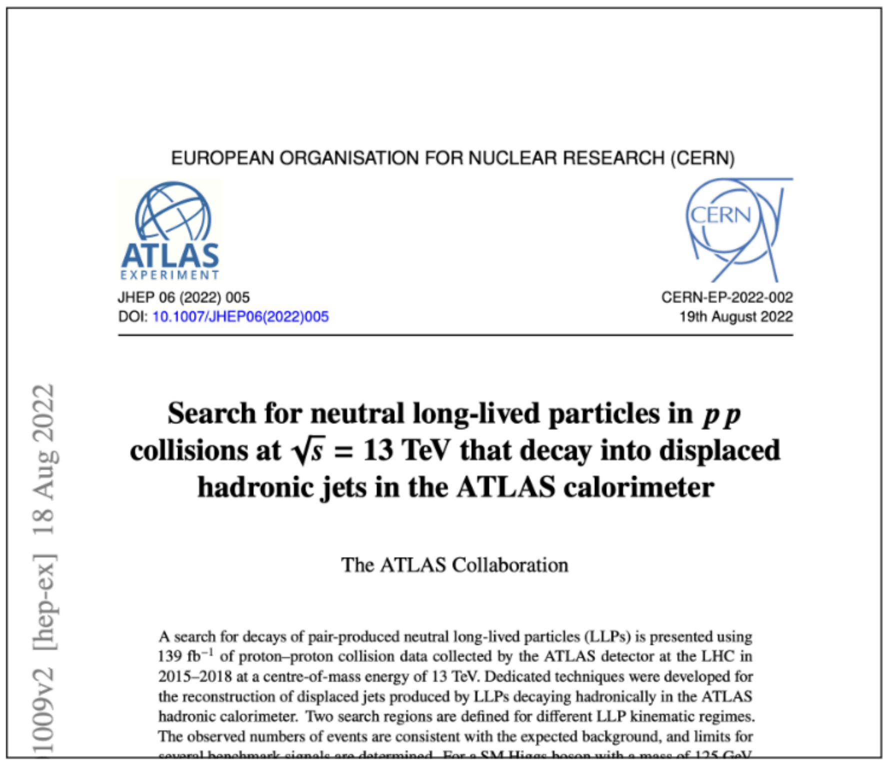
From Sezen's talk at Deep Dive

- Provide *exact definitions (including units, ordering and conventions) of network input features*, either as code examples or documentation.
- Provide a *sample of input features and output values* for technical validation.
- Provide *plots of input and output variables* for validation samples, if possible, with some indication of feature importance, as analysis supporting data.
- Provide *cut-flow information*, both before and after any ML-based selections.
- *Validated and runnable published analysis code* can be the clearest expression of both the general analysis logic and the specific interfacing with the ML functions.
- *Full descriptions of the physics models* used to generate the information above, e.g. SLHA files and generator run cards, are essential inputs to validating any serious reinterpretation

- ML training/evaluation code might not be essential in most cases but it is certainly encouraged (to be seen case by case)
- These guidelines go together with usual analysis preservations ones (e.g. analysis logic and analysis code/snippets, likelihoods cutflows, distributions, etc…)

Some examples from ATLAS

ATLAS published three Analyses with ONNX
(+ 1 Analyses that store PyTorch specific serialization)



EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

JHEP 06 (2022) 005
DOI: 10.1007/JHEP06(2022)005

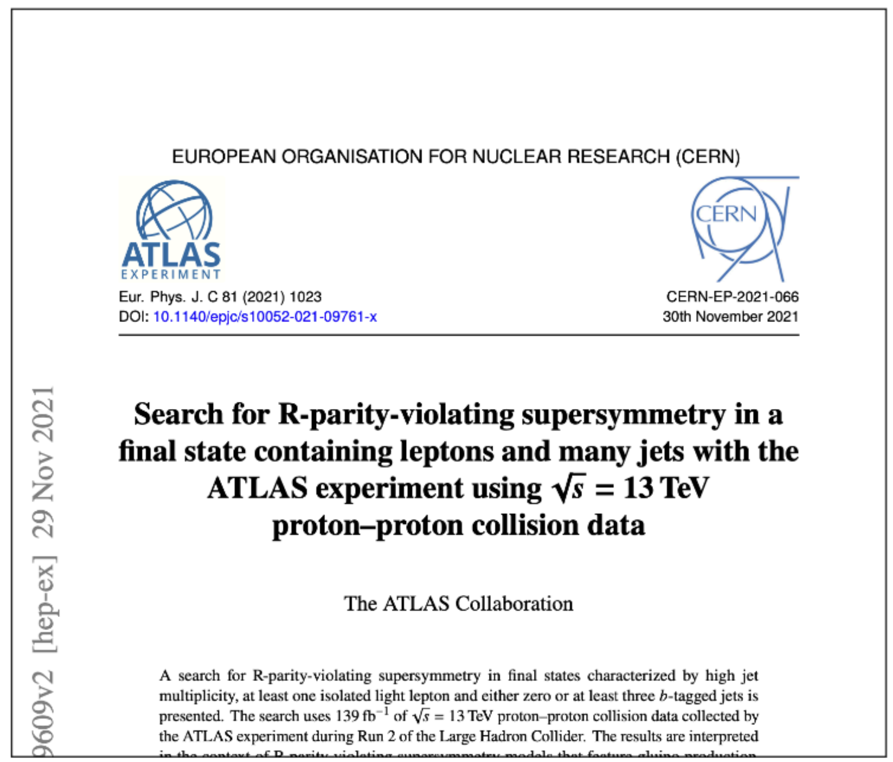
CERN-EP-2022-002
19th August 2022

Search for neutral long-lived particles in pp collisions at $\sqrt{s}$ = 13 TeV that decay into displaced hadronic jets in the ATLAS calorimeter

The ATLAS Collaboration

A search for decays of pair-produced neutral long-lived particles (LLPs) is presented using 139 fb^{-1} of proton–proton collision data collected by the ATLAS detector at the LHC in 2015–2018 at a centre-of-mass energy of 13 TeV. Dedicated techniques were developed for the reconstruction of displaced jets produced by LLPs decaying hadronically in the ATLAS hadronic calorimeter. Two search regions are defined for different LLP kinematic regimes. The observed numbers of events are consistent with the expected background, and limits for

https://www.hepdata.net/record/ins2043503



EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

Eur. Phys. J. C 81 (2021) 1023
DOI: 10.1140/epjc/s10052-021-09761-x

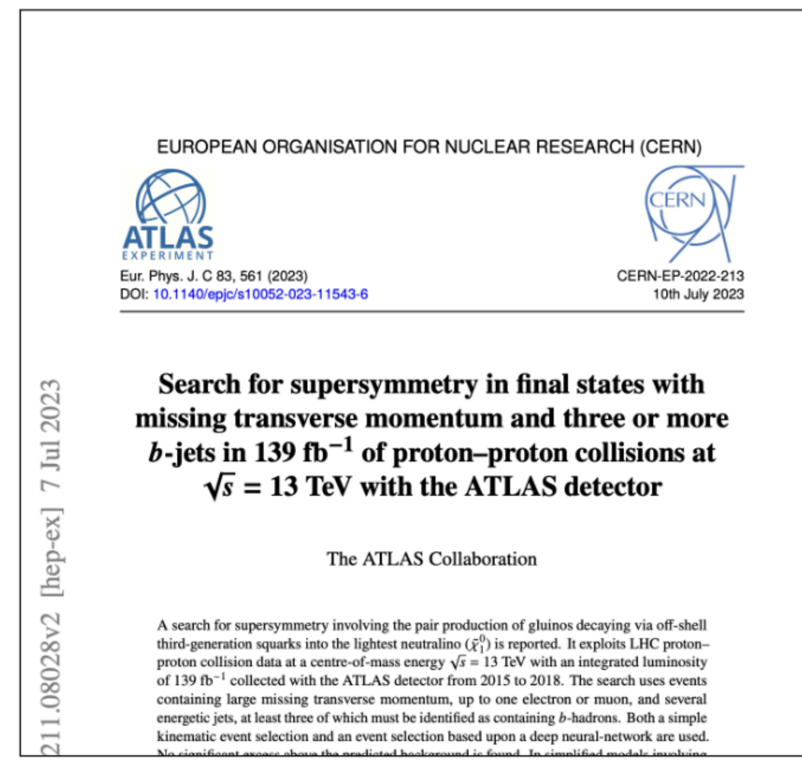
CERN-EP-2021-066
30th November 2021

Search for R-parity-violating supersymmetry in a final state containing leptons and many jets with the ATLAS experiment using $\sqrt{s}$ = 13 TeV proton–proton collision data

The ATLAS Collaboration

A search for R-parity-violating supersymmetry in final states characterized by high jet multiplicity, at least one isolated light lepton and either zero or at least three b-tagged jets is presented. The search uses 139 fb^{-1} of $\sqrt{s}$ = 13 TeV proton–proton collision data collected by the ATLAS experiment during Run 2 of the Large Hadron Collider. The results are interpreted

https://www.hepdata.net/record/ins1869040



EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

Eur. Phys. J. C 83, 561 (2023)
DOI: 10.1140/epjc/s10052-023-11543-6

CERN-EP-2022-213
10th July 2023

Search for supersymmetry in final states with missing transverse momentum and three or more b-jets in 139 fb^{-1} of proton–proton collisions at $\sqrt{s}$ = 13 TeV with the ATLAS detector

The ATLAS Collaboration

A search for supersymmetry involving the pair production of gluinos decaying via off-shell third-generation squarks into the lightest neutralino ($\tilde{\chi}^0_1$) is reported. It exploits LHC proton–proton collision data at a centre-of-mass energy $\sqrt{s}$ = 13 TeV with an integrated luminosity of 139 fb^{-1} collected with the ATLAS detector from 2015 to 2018. The search uses events containing large missing transverse momentum, up to one electron or muon, and several energetic jets, at least three of which must be identified as containing b-hadrons. Both a simple kinematic event selection and an event selection based upon a deep neural-network are used.

https://www.hepdata.net/record/ins2182381

[From Lukas talk at Deep Dive]

Some examples from ATLAS

ATLAS started with publishing TMVA XML (which in turn can be used with `petrify-bdt`) or `petrify-bdt` standalone code directly

EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

ATLAS EXPERIMENT

CERN

JHEP 06 (2022) 005
DOI: 10.1007/JHEP06(2022)005

CERN-EP-2022-002
19th August 2022

1009v2 [hep-ex] 18 Aug 2022

Search for neutral long-lived particles in pp collisions at $\sqrt{s}$ = 13 TeV that decay into displaced hadronic jets in the ATLAS calorimeter

The ATLAS Collaboration

A search for decays of pair-produced neutral long-lived particles (LLPs) is presented using 139 fb^{-1} of proton–proton collision data collected by the ATLAS detector at the LHC in 2015–2018 at a centre-of-mass energy of 13 TeV. Dedicated techniques were developed for the reconstruction of displaced jets produced by LLPs decaying hadronically in the ATLAS hadronic calorimeter. Two search regions are defined for different LLP kinematic regimes. The observed numbers of events are consistent with the expected background, and limits for

https://www.hepdata.net/record/ins2043503

EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

ATLAS EXPERIMENT

CERN

JHEP 02 (2021) 143
DOI: 10.1007/JHEP02(2021)143

CERN-EP-2020-166
14th January 2023

4293v2 [hep-ex] 1 Mar 2021

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

The ATLAS Collaboration

A search for the supersymmetric partners of quarks and gluons (squarks and gluinos) in final states containing jets and missing transverse momentum, but no electrons or muons, is presented. The data used in this search were recorded by the ATLAS experiment in proton–proton collisions at a centre-of-mass energy of $\sqrt{s}$ = 13 TeV during Run 2 of the

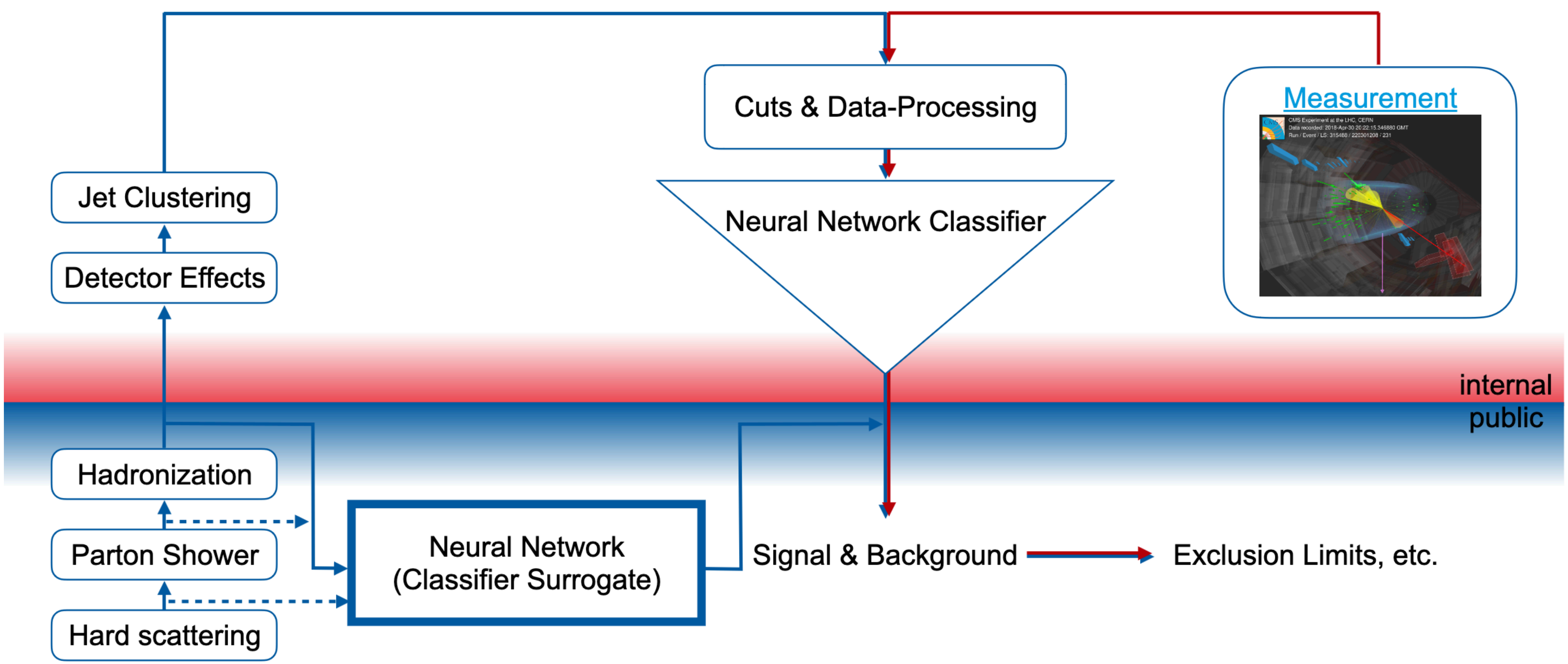
https://www.hepdata.net/record/ins1827025

[From Lukas talk at Deep Dive]

Surrogate models

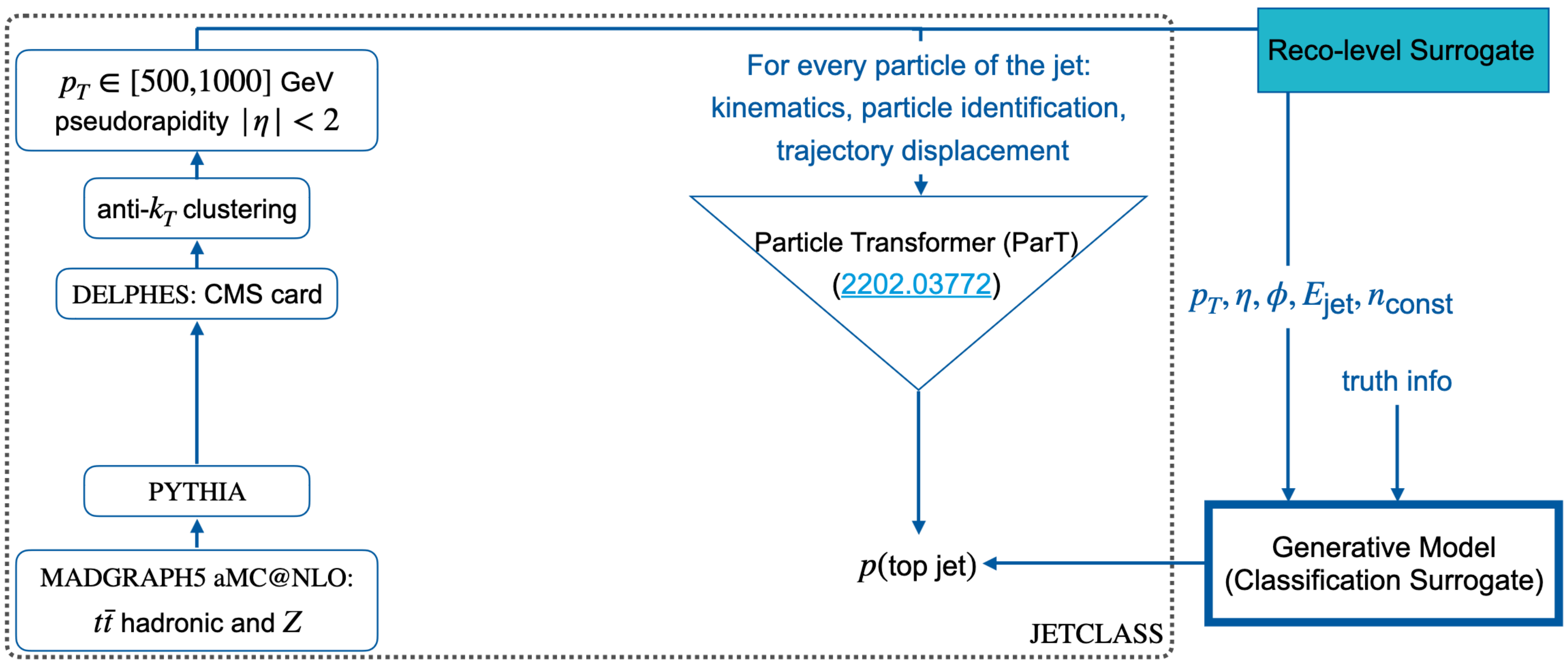
- **Main problem:** the theorist might use a Delphes-simulated signal which differs from full CMS simulation → wrong response of the ML model

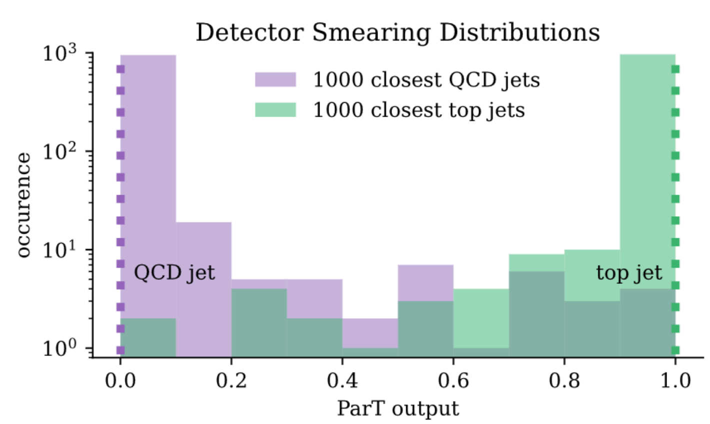
- **A surrogate model as solution:** neural network trained to replicate the output of the original ML model but using input events with a simpler set of attributes

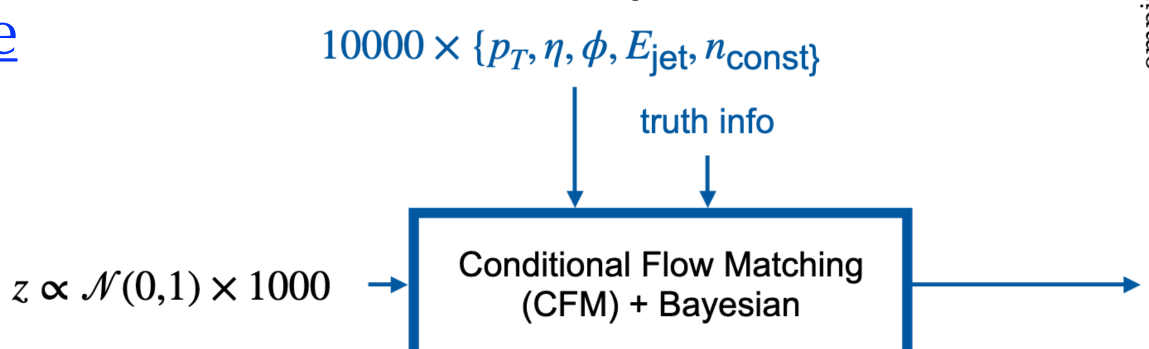


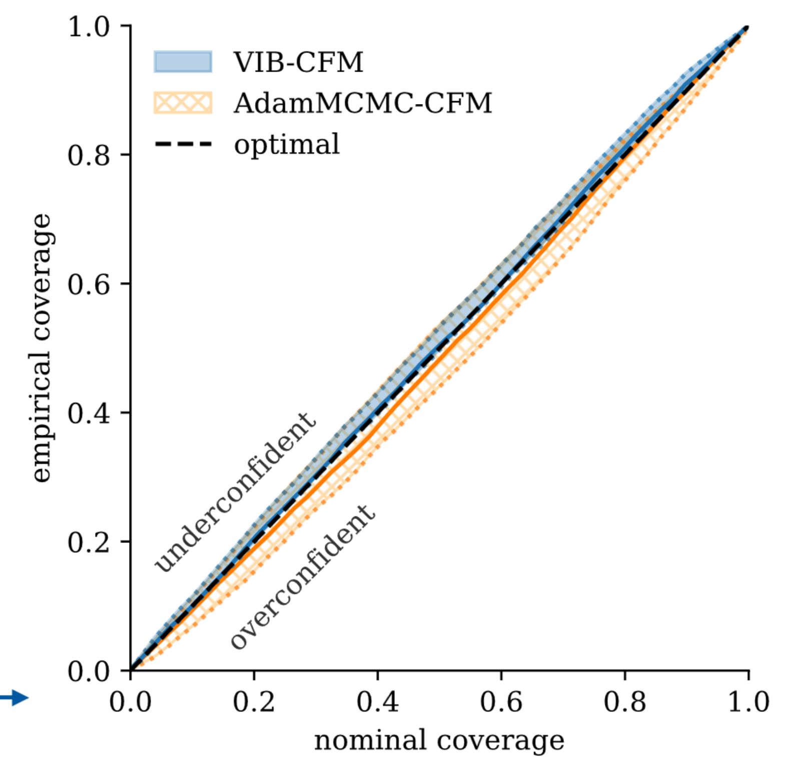
From Sebastian talk at Deep Dive

Surrogate model: toy setup







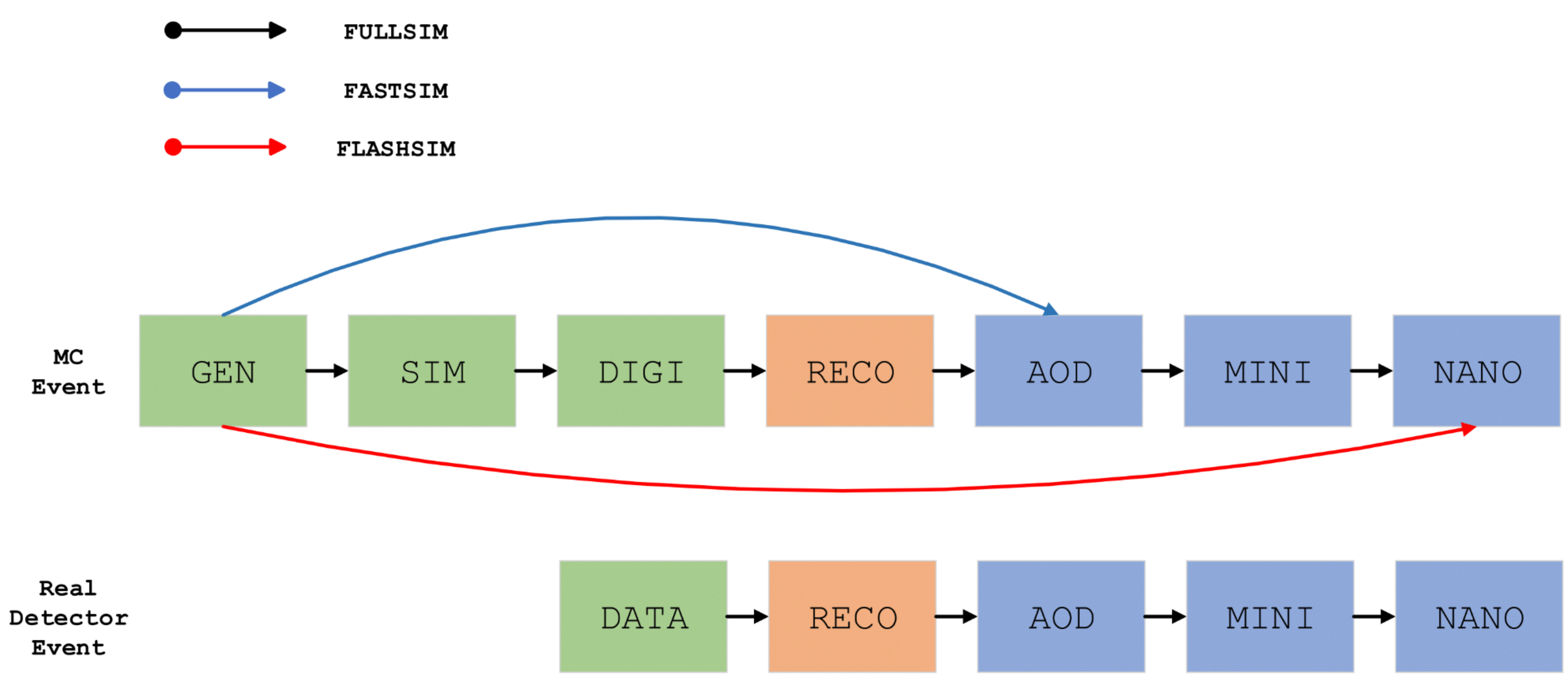


Both architectures deliver **well calibrated** Surrogates

[From Sebastian talk at Deep Dive]

FlashSim

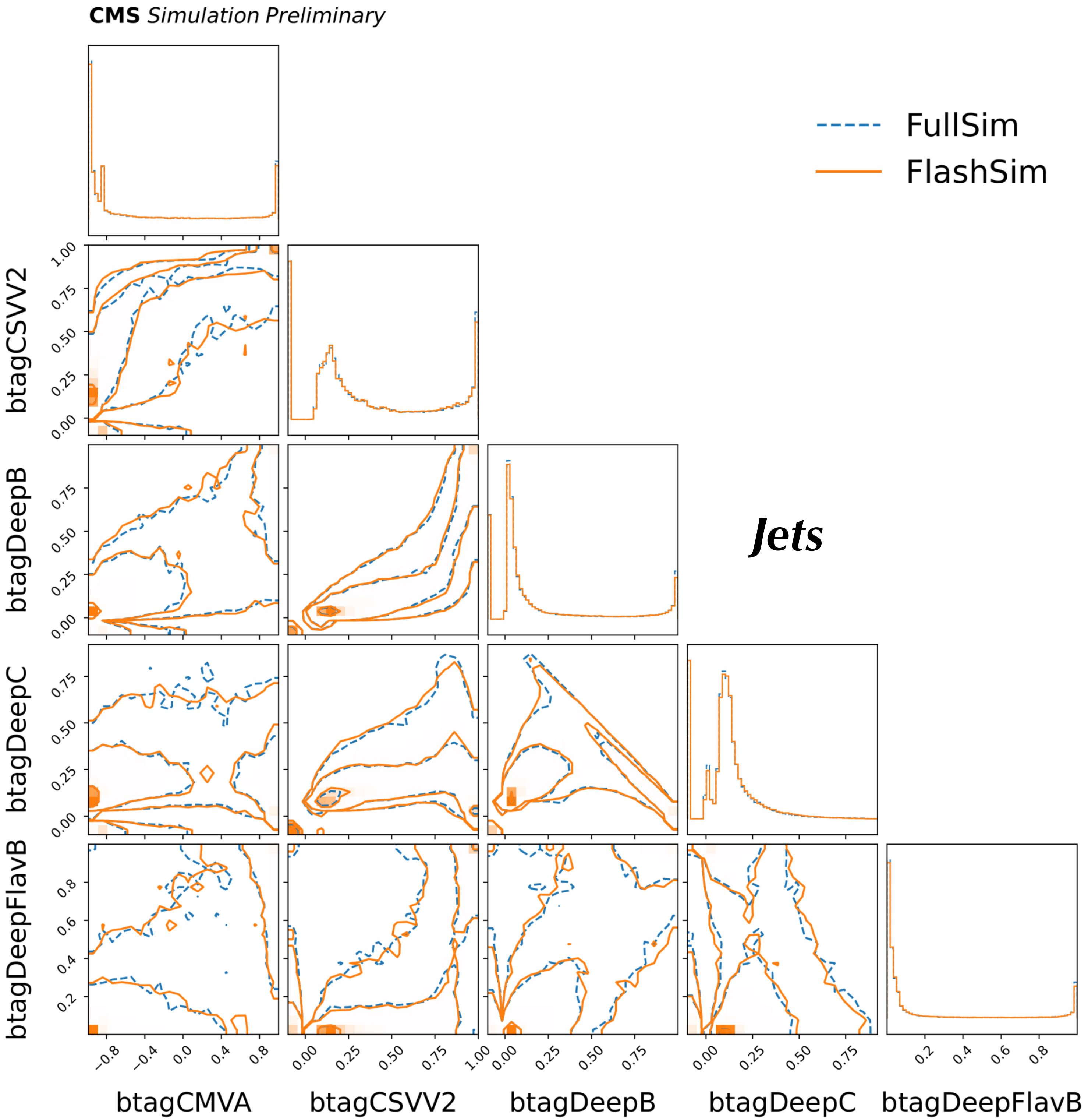
- Universal, fast ML-based end-to-end simulation
- Targets: quickly retrainable, fast as Delphi's, accuracy between FastSim and FullSim



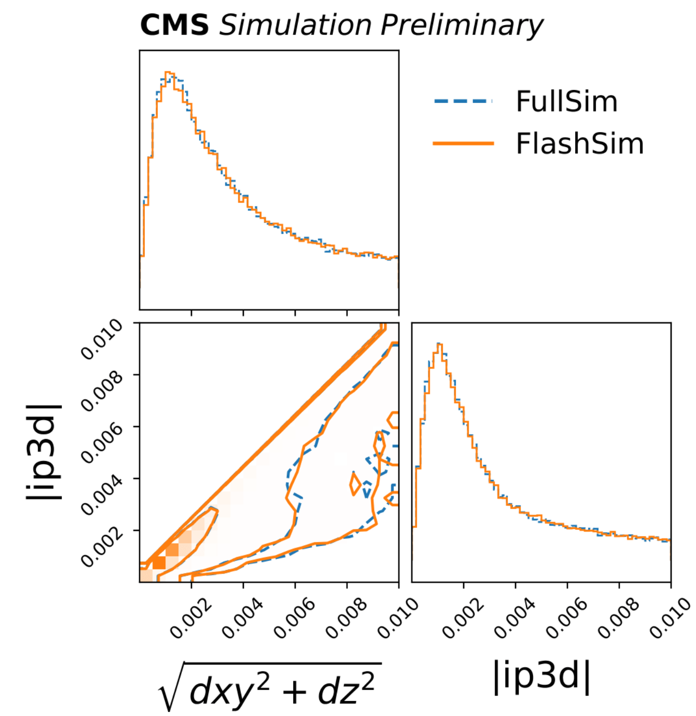
From Andrea's talk at Deep Dive

FlashSim Performance

- Promising results reproducing distributions of features of **jets**, **muons**, electrons, and more



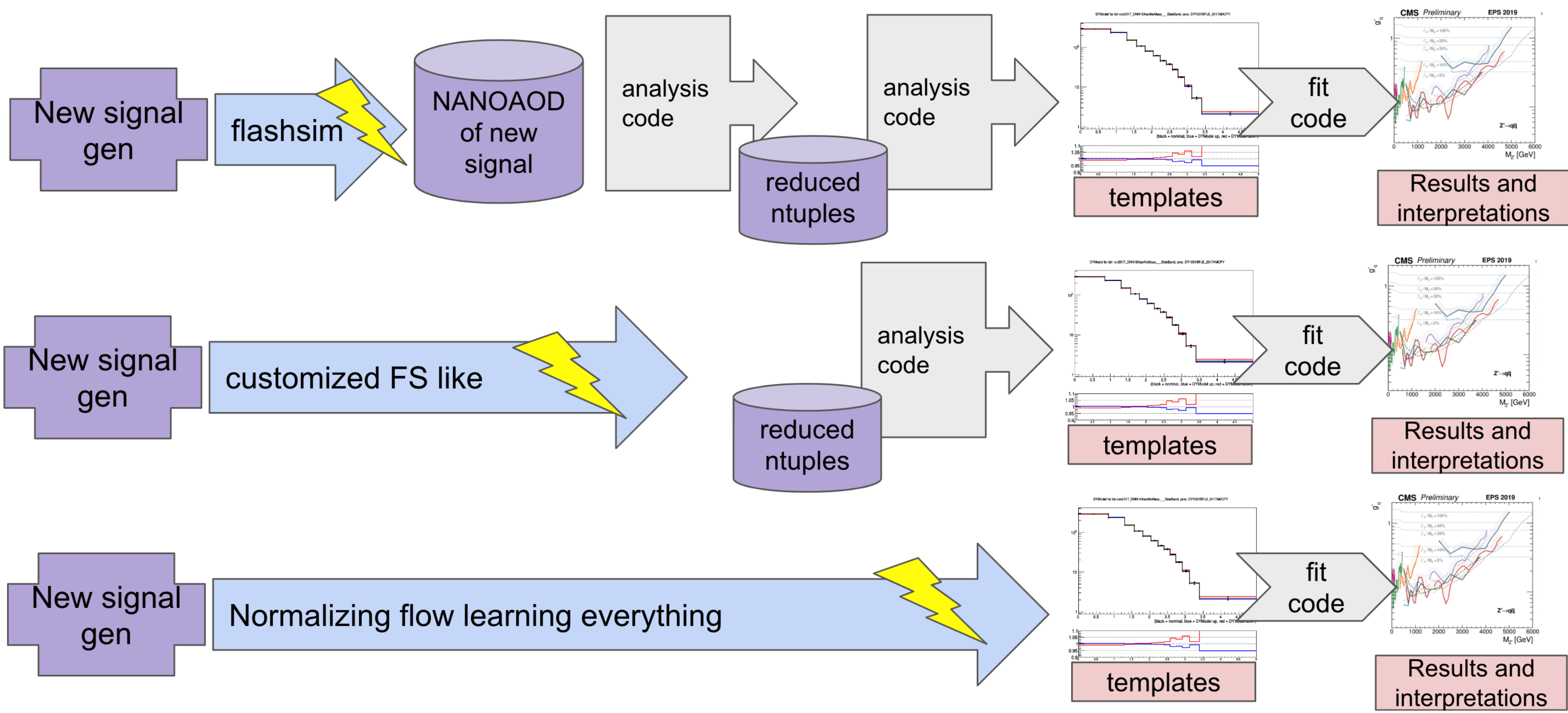
Jets



Muons

FlashSim Scenarios

- Publish FlashSim model(s) to enable reinterpretation on new generated signal
- Several possible scenarios envisioned: (1) fully general NANOAOD, (2) analysis-specific ntuples, (3) final analysis observables



Where do we start?

- In general, any analysis using simple NN (or BDT) should at least publish in HEPData the ONNX model (petrify-bdt/xml for BDTs) similarly to what ATLAS did
  - Important to document all aspects of the model, e.g. expected use/performance, input preprocessing, etc. along the lines of a Hugging Face model card: https://huggingface.co/docs/hub/en/model-cards
- Other analyses using less reusable models and inputs will require a case by case discussion
  - e.g., publish training code with toy dataset and/or train surrogate model
- We considered a few analyses and a few POGs papers where we can start applying recommendations

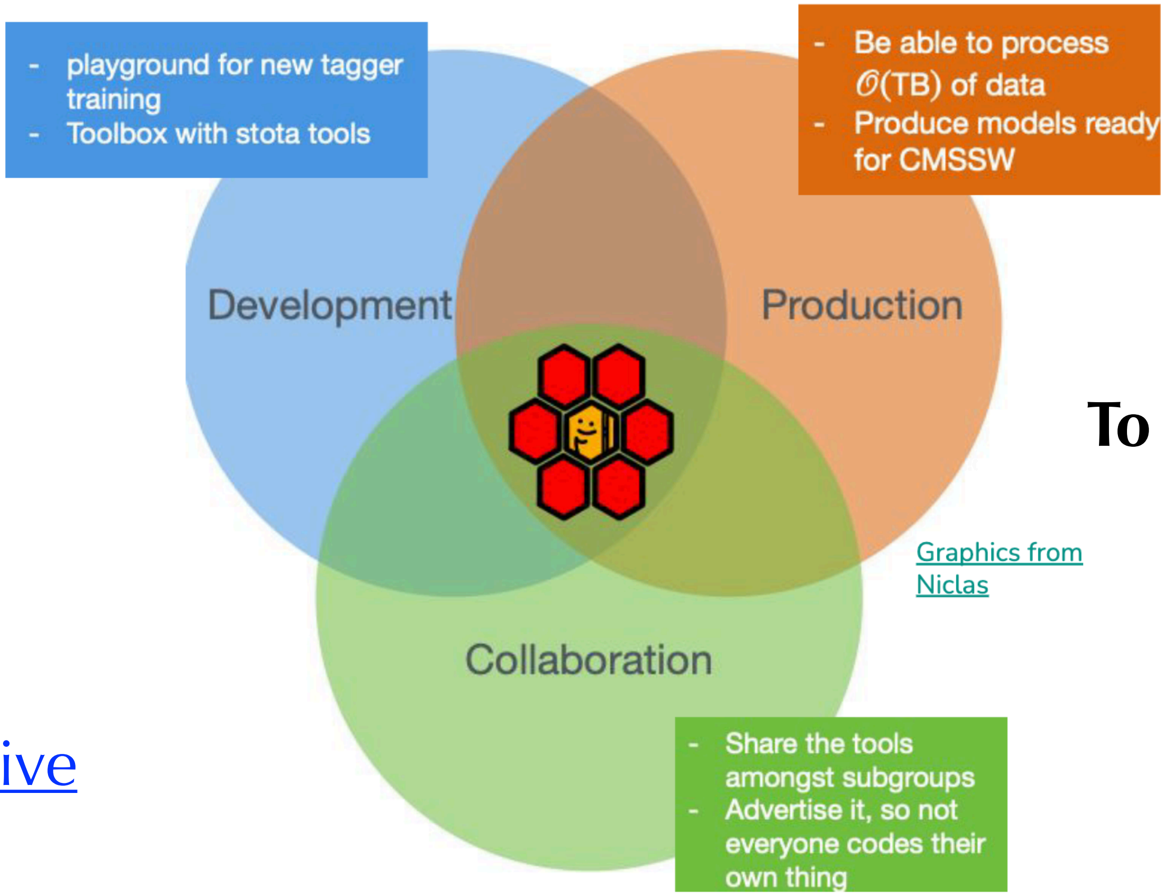
PAG Feedback

- Several analyses shared feedback on challenges and opportunities
- **EXO-22-026:** Searching for new physics detecting anomalies in jets (CASE)
    - Weakly supervised models require retraining for new signals => Publish training code alongside example trained models
- **SUS-23-001:** Search for Stealth/RPV stops using Double DisCo neural network method
    - Several analyses use CMS-specific information not available in Delphes, e.g. b-tagger discriminants => Need FlashSim or similar surrogate model solution to enable reinterpretation
- **EXO-22-015:** Search for Emerging Jets with full Run 2 data
- **EXO-22-020:** Search for new physics with at least one displaced vertex and missing energy

POG Feedback

b-hive : An object-tagging and ML framework

- Modular framework for NN training pipelines
- Working on NanoAOD-like inputs
- Bookkeeping, versioning, reproducibility



Tutorial, gitlab, mattermost
Presentations:
CAT in CMS week, MLG Forum

Graphics from Niclas

To be adopted also by JME

From BTV talk at Deep Dive

MLG+POG publications currently under discussion
→ candidates for ML model release:

- AK8 Particle Transformer
- AK4 Particle Transformer
- DeepMET
- ABCNet for PU mitigation

FAIR AI Models

- What is the right way to share training code?
- Can leverage “cookie cutter” project structure https://github.com/FAIR4HEP/cookiecutter4fair

IOP Publishing Mach. Learn.: Sci. Technol. **4** (2023) 045062 https://doi.org/10.1088/2632-2153/ad12e

MACHINE LEARNING Science and Technology

PAPER

CrossMark

OPEN ACCESS

RECEIVED
21 December 2022

REVISED
27 October 2023

ACCEPTED FOR PUBLICATION
6 December 2023

PUBLISHED
29 December 2023

FAIR AI models in high energy physics

Javier Duarte[1,*], Haoyang Li[1], Avik Roy[2], Ruike Zhu[2,3], E A Huerta[3,4], Daniel Diaz[1], Philip Harris[5], Raghav Kansal[1], Daniel S Katz[2], Ishaan H Kavoori[1], Volodymyr V Kindratenko[2], Farouk Mokhtar[1,6], Mark S Neubauer[2], Sang Eon Park[5], Melissa Quinnan[1], Roger Rusack[7] and Zhizhen Zhao[2]

1 University of California San Diego, La Jolla, CA 92093, United States of America
2 University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States of America
3 Argonne National Laboratory, Lemont, IL 60439, United States of America
4 The University of Chicago, Chicago, IL 60637, United States of America
5 Massachusetts Institute of Technology, Cambridge, MA 02139, United States of America
6 Halıcıoğlu Data Science Institute, La Jolla, CA 92093, United States of America
7 The University of Minnesota, Minneapolis, MN 55405, United States of America
* Author to whom any correspondence should be addressed.

E-mail: jduarte@ucsd.edu

Keywords: FAIR, AI, high energy physics, Higgs boson, ML

Abstract

The findable, accessible, interoperable, and reusable (FAIR) data principles provide a framework for examining, evaluating, and improving how data is shared to facilitate scientific discovery. Generalizing these principles to research software and other digital products is an active area of research. Machine learning models—algorithms that have been trained on data without being explicitly programmed—and more generally, artificial intelligence (AI) models, are an important target for this because of the ever-increasing pace with which AI is transforming scientific domains such as experimental high energy physics (HEP). In this paper, we propose a practical definition o FAIR principles for AI models in HEP and describe a template for the application of these principles. We demonstrate the template's use with an example AI model applied to HEP, in which a graph neural network is used to identify Higgs bosons decaying to two bottom quarks. We repor on the robustness of this FAIR AI model, its portability across hardware architectures and softwar frameworks, and its interpretability.

```
├── LICENSE            <- License for reusing code
├── Makefile           <- Makefile with commands like `make data` or `make train`
├── CITATION.cff       <- Standardized citation metadata
├── README.md          <- The top-level README for developers using this project
├── data
│   ├── processed      <- The final, canonical data sets for modeling
│   └── raw            <- The original, FAIR, and immutable data dump
│
├── Dockerfile         <- For building a containerized environment
│
├── docs               <- A default Sphinx project for documentation; see sphinx-doc.org for details
│
├── models             <- Trained and serialized models, model predictions, or model summaries
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
│                         the creator's initials, and a short `-` delimited description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
│
├── references         <- Data dictionaries, manuals, and all other explanatory materials
│
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── setup.py           <- Makes project pip installable (`pip install -e .`) so src can be imported
├── src                <- Source code for use in this project
│   ├── __init__.py    <- Makes `src` a Python module
│   │
│   ├── data           <- Scripts to download or generate data
│   │   └── make_dataset.py
│   │
│   ├── features       <- Scripts to turn raw data into features for modeling
│   │   └── build_features.py
│   │
│   ├── models         <- Scripts to train models and then use trained models to make
│   │   │                 predictions
│   │   ├── predict_model.py
│   │   └── train_model.py
│   │
│   └── visualization  <- Scripts to create exploratory and results oriented visualizations
│       └── visualize.py
│
└── tox.ini            <- Tox file with settings for running `tox`; see tox.readthedocs.io
```

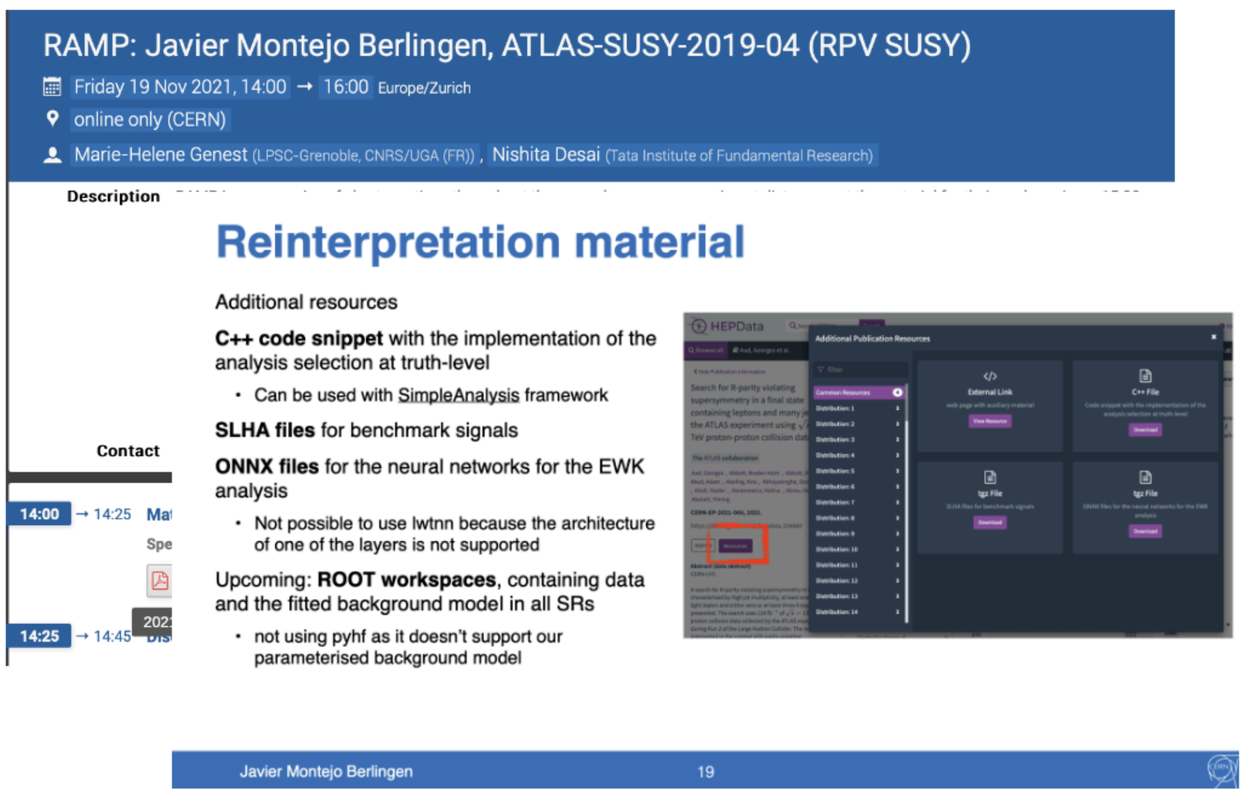
Conclusions

- Questions considered:
  - what are the best practices for sharing ML models in industry? in other sciences? in HEP?
  - how do we make published CMS ML models easy to find / linked to the paper?
  - how do we make published CMS ML models the most useful for reinterpretation?
  - what formats are CMS ML models currently stored in?
  - what types of inputs do CMS ML models use (low-level CMS-specific inputs? high-level particle inputs?)
  - how do we determine when a CMS ML model should be published (or just an efficiency map?)
  - should we release training code as well?
- Goal: **draft internal CMS recommendations to publish CMS ML models** for physics/ML papers as part of the publication pipeline
  - similar to HEPData requirements
- First attempts will not be the optimal but we need to start from somewhere…

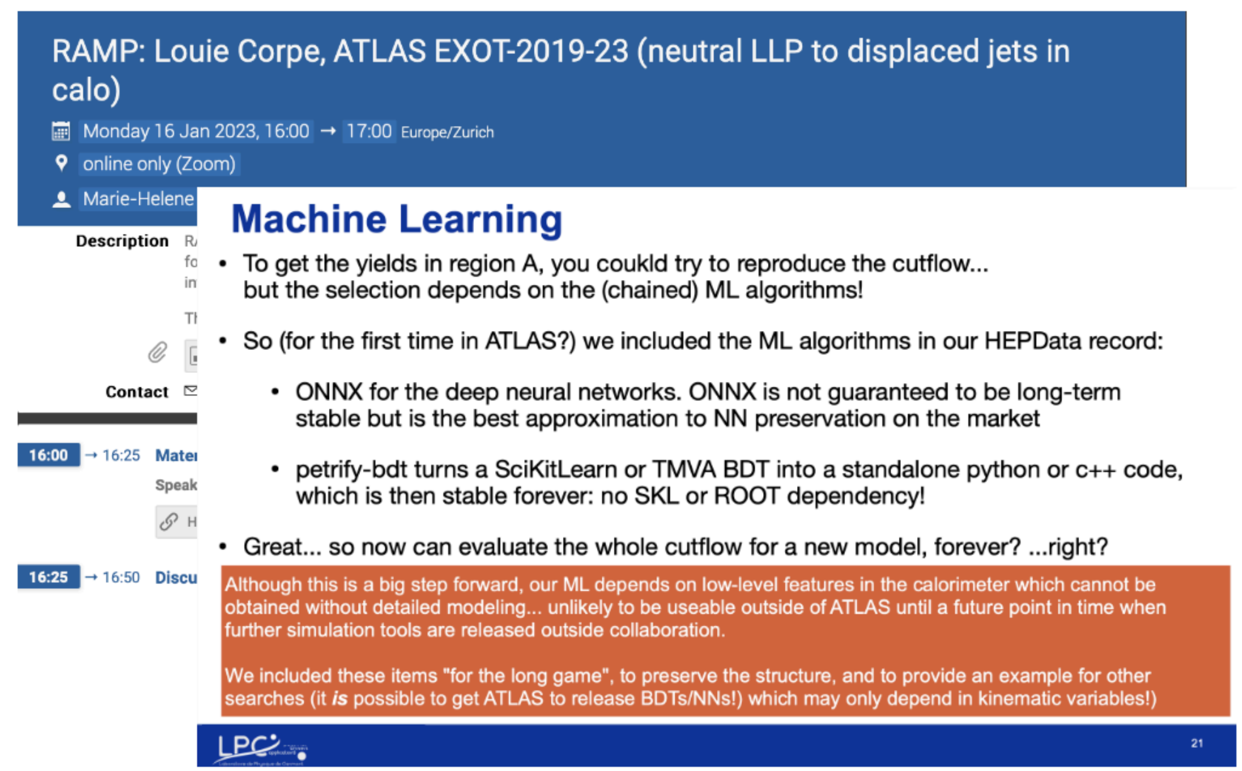
[From Lukas talk at Deep Dive]

RAMP Seminars

Preserving the just the pure model is not enough. Important Forum to bring together those who publish and those who re-use to check e.g. limitations, option questions, etc



https://indico.cern.ch/event/1083851/



https://indico.cern.ch/event/1233294/

Backup

FAIR Data Principles

https://www.go-fair.org/fair-principles/

Image: book.fosteropenscience.eu

FAIR Data Principles

https://www.go-fair.org/fair-principles/

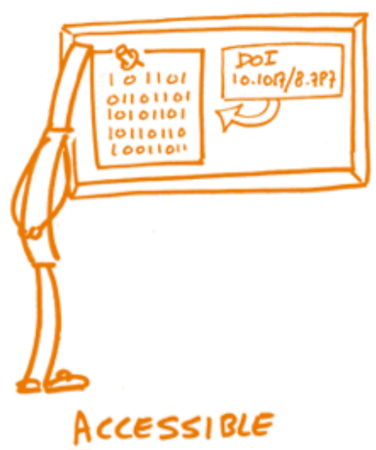
F1. (meta)data have **unique** and **persistent** identifier
F2. data are described with rich metadata
F3. metadata specify the data identifier
F4. (meta)data are registered or indexed in a searchable resource

Image: book.fosteropenscience.eu

FAIR Data Principles

https://www.go-fair.org/fair-principles/

F1. (meta)data have **unique** and **persistent** identifier
F2. data are described with rich metadata
F3. metadata specify the data identifier
F4. (meta)data are registered or indexed in a searchable resource

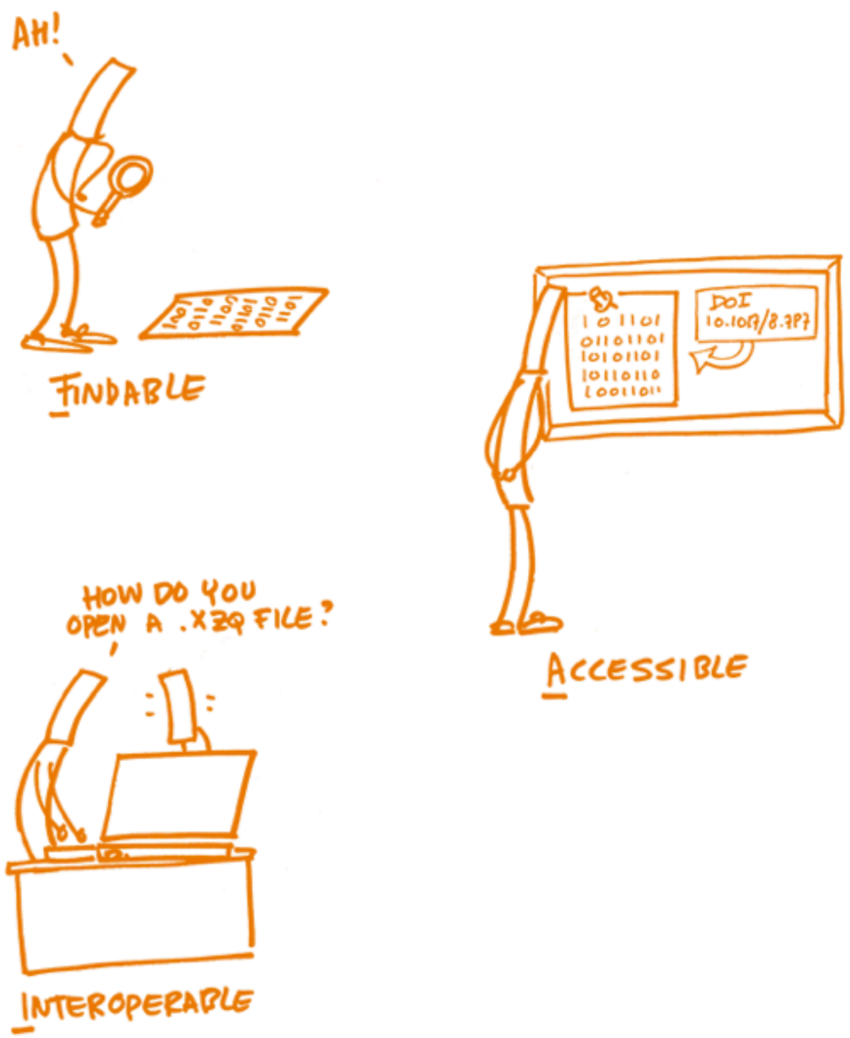


A1. (meta)data are retrievable using standardized protocol
A1.1 protocol is open, free, and universally implementable
A1.2 protocol allows for authentication and authorization
A2. metadata are accessible, even when the data is not

Image: book.fosteropenscience.eu

FAIR Data Principles

https://www.go-fair.org/fair-principles/



F1. (meta)data have **unique** and **persistent** identifier
F2. data are described with rich metadata
F3. metadata specify the data identifier
F4. (meta)data are registered or indexed in a searchable resource

A1. (meta)data are retrievable using standardized protocol
A1.1 protocol is open, free, and universally implementable
A1.2 protocol allows for authentication and authorization
A2. metadata are accessible, even when the data is not

I1. (meta)data use a formal, shared, and broadly applicable language for knowledge representation
I2. (meta)data use **vocabularies** that follow FAIR principles
I3. (meta)data include qualified references to other (meta)data

Image: book.fosteropenscience.eu

FAIR Data Principles

https://www.go-fair.org/fair-principles/

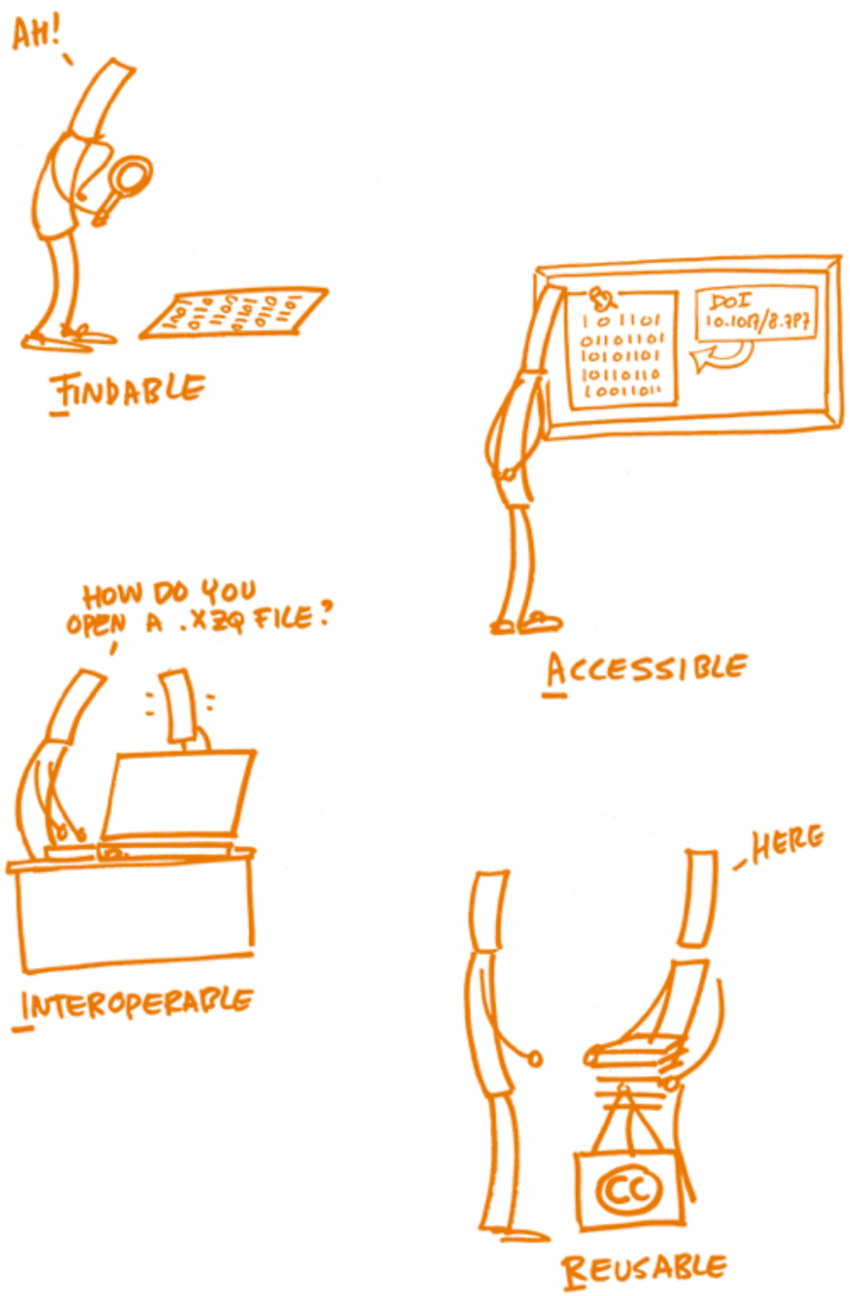


Image: book.fosteropenscience.eu

F1. (meta)data have **unique** and **persistent** identifier
F2. data are described with rich metadata
F3. metadata specify the data identifier
F4. (meta)data are registered or indexed in a searchable resource

A1. (meta)data are retrievable using standardized protocol
A1.1 protocol is open, free, and universally implementable
A1.2 protocol allows for authentication and authorization
A2. metadata are accessible, even when the data is not

I1. (meta)data use a formal, shared, and broadly applicable language for knowledge representation
I2. (meta)data use **vocabularies** that follow FAIR principles
I3. (meta)data include qualified references to other (meta)data

R1. (meta)data have a plurality of accurate and relevant attributes
R1.1. (meta)data have clear and accessible data usage license
R1.2. (meta)data are associated with their provenance
R1.3. (meta)data meet domain-relevant community standards