Artificial Intelligence and the Uncertainty challenge in Fundamental Physics
27 Nov - 1 Dec 2023, SCAI Paris & Inst. Pascal

Sabine Kraml, LPSC Grenoble

Publication and reuse of ML models

in (from) LHC analyses

CNRS IN2P3 Les deux infinis
LPSC GRENOBLE | MODANE
AISSAI AI for science, science for AI

Publication and reuse - why care

Scientific work (data, code, analyses, results …) should be

- Findable
- Accessible
- Interoperable
- Reusable

Publication and reuse - why care

Scientific work (data, code, analyses, results …) should be

- **F**indable
- **A**ccessible
- **I**nteroperable
- **R**eusable

arXiv HEPData GitHub …

?

Publication and reuse - why care

FAIR-ification is a good idea …

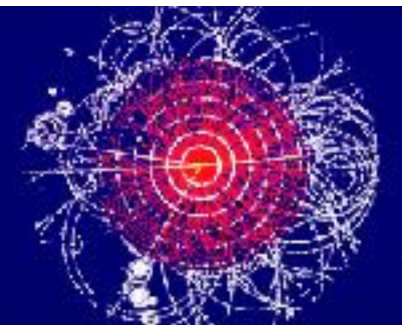
e.g., updating constraints,
testing new hypotheses,
performing combinations
and/or fits, etc.

→ new research based on existing data and analyses

longer shelf life & more scientific impact

Lightweight, public

Analysis preservation



Event selection

Signal selection
(signal/bkg discrimination)

Statistical eval.
(hypo test, interpretation)

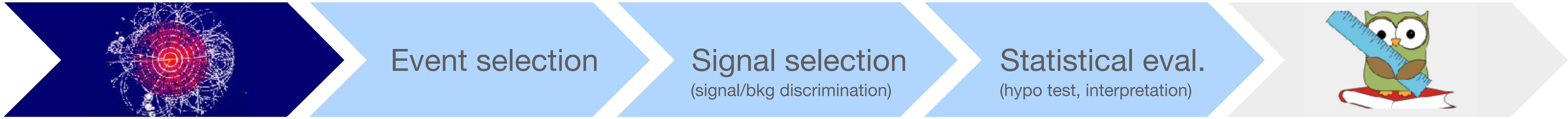
Preservation of **analysis logic and workflows** enables the reuse of the original analysis process and associated data products.

> “Ensure that **release of analysis preservation logic via public frameworks** for the community to use is integrated with experiment publication and data-release processes, **to maximise analysis impact**.”

Snowmass white paper on data and analysis preservation and reinterpretation
S. Bailey et al., arXiv:2203.10057

Lightweight, public

Analysis preservation



object definitions; identification, tagging, reconstruction efficiencies

detailed (pre)selection and signal/control region cuts

- Nowadays well established for `traditional' cut-based analyses
- Status and recommendations after Run2: arXiv:2003.07868
  [LHC Reinterpretation Forum]

pyhf

RIVET

ColliderBit

MadAnalysis 5

CheckMATE

ADL

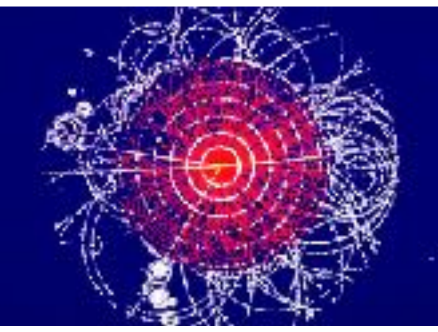
LLP Recasting

ATLAS SimpleAnalysis

....

Lightweight, public

Analysis preservation



Event selection

Signal selection
(signal/bkg discrimination)

Statistical eval.
(hypo test, interpretation)

object definitions;
identification, tagging,
reconstruction efficiencies

detailed (pre)selection and
signal/control region cuts

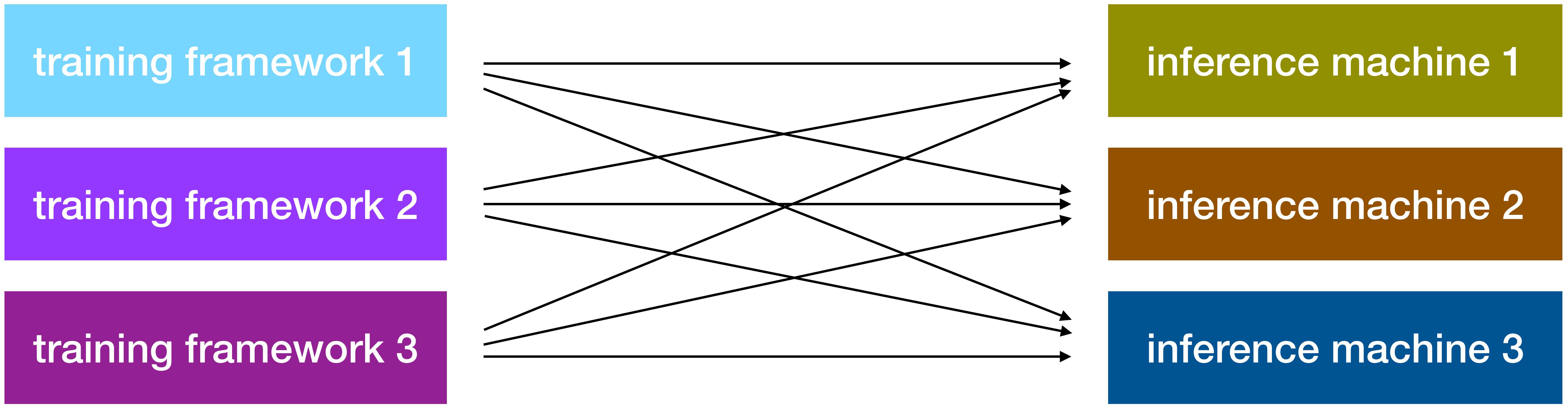
- Nowadays well established for `traditional' cut-based analyses
- Status and recommendations after Run2: arXiv:2003.07868
  [LHC Reinterpretation Forum]

Bottleneck: ML-based analyses

pyhf
RIVET
ColliderBit
MadAnalysis 5
CheckMATE
ADL
LLP Recasting
ATLAS SimpleAnalysis
....

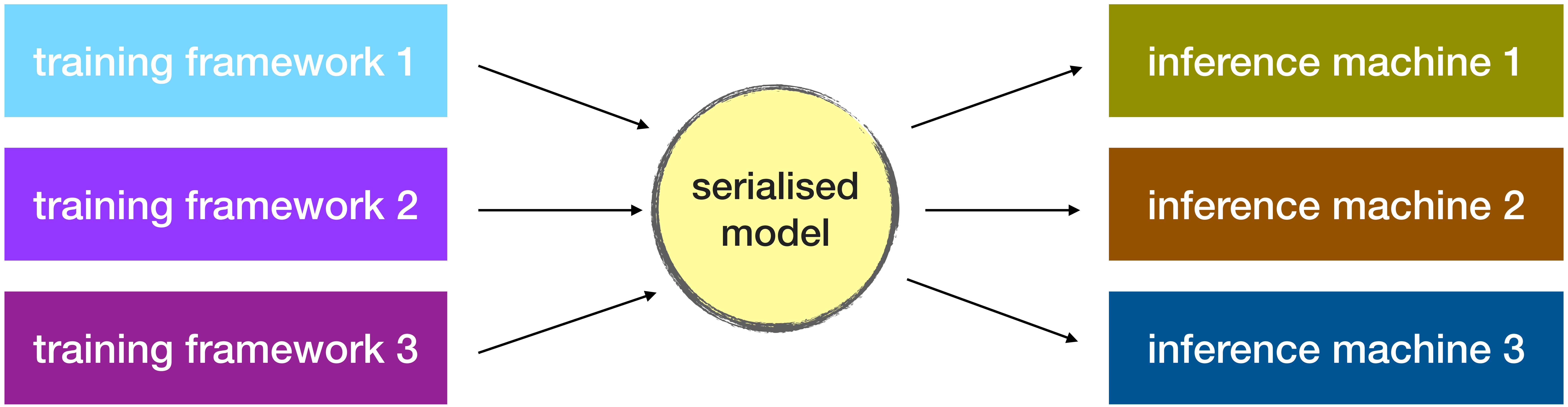
With machine learning:

one problematic is training and inferencing pipeline



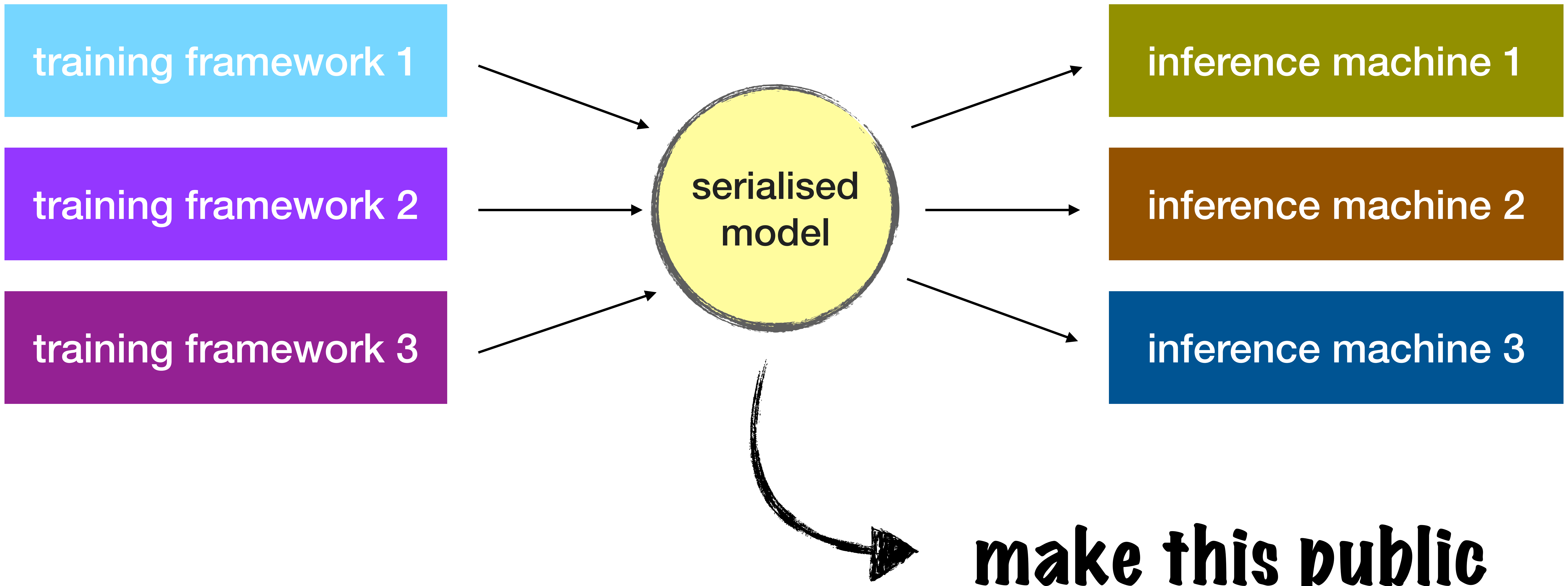
With machine learning:

better: training and inferencing interoperability;
save ML model in a stable exchange format



With machine learning:

better: training and inferencing interoperability;
save ML model in a stable exchange format



make this public

together with a clear description of inputs and output

Solutions for neural nets (e.g.)

Lightweight Trained Neural Network

- Designed to take tensorflow/sk-learn trained NNs and run them in C++
- Originally developed for ATLAS trigger; used internally in the collaboration
- Minimal dependencies: Eigen and Boost only; 20 operators.
- Human-readable JSON files

`lwtnn`

Open Neural Network Exchange

- Designed to allow NNs trained in one context to be run in a completely different one
- Industry standard, developed by Facebook and Microsoft
- Supports tensorflow, pytorch, sk-learn and more; almost 200 operators
- Binary ONNX files

ONNX

and more … see e.g. talk by Dan Guest at Reinterpretation Forum 2022

Existing examples from ATLAS

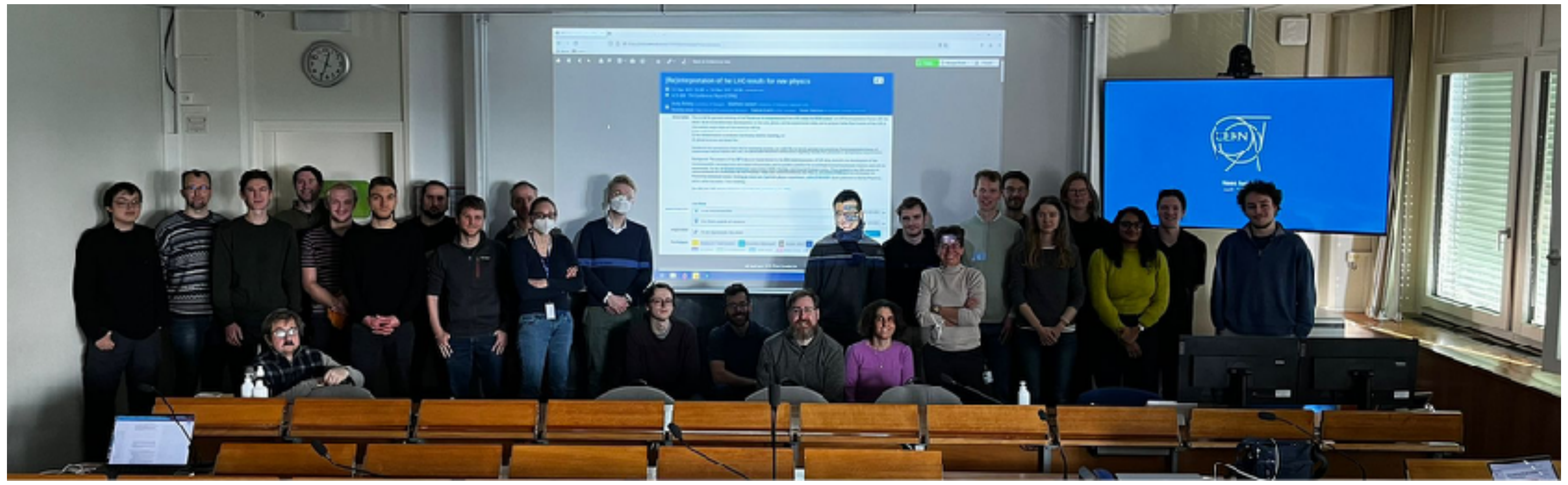
SUSY-2018-22	Search for squarks and gluinos: jets+MET BDT weights in XML format on HEPData + simpleAnalysis implementation	Nov 2020
SUSY-2019-04	RPV SUSY search, leptons + many jets ONNX files for 5 NNs (4-8 jets SRs) on HEPData + simpleAnalysis implementation	Sep 2021
SUSY-2018-30	SUSY search with MET and many b-jets simpleAnalysis implementation with ONNX-serialised NN model	Nov 2022
EXOT-2019-23	Search for neutral LLPs with displaced hadronic jets ("CalRatio LLP search") preserved NNs as ONNX, BDTs as executables with petrify-bdt; low level inputs; also 6d efficiency maps parametrising the BDT+NN selection + example code	June 2022
HDBS-2019-23	Anomaly detection search for new resonances Y → X+H in hadronic final states VRNN python code + post-training weights (PyTorch .pth file)	June 2023

In the Reinterpretation Forum

7th RiF workshop 12–15 Dec 2022 at CERN

Session on publication and reuse of ML models

Introduction	*Sabine Kraml*
30/7-018 - Kjell Johnsen Auditorium, CERN	16:30 - 16:35
Machine learning model serialization experiences	*Dan Guest*
30/7-018 - Kjell Johnsen Auditorium, CERN	16:40 - 16:55
Reusing Neural Networks: Lessons learned and Suggestions for the future	*Tomasz Procter*
30/7-018 - Kjell Johnsen Auditorium, CERN	17:00 - 17:15
Implementation of ML searches in CheckMATE	*Krzysztof Rolbiecki*
30/7-018 - Kjell Johnsen Auditorium, CERN	17:20 - 17:35
CMS inputs on ML models re-usability	*Jennifer Ngadiuba*
30/7-018 - Kjell Johnsen Auditorium, CERN	17:40 - 17:50
Publication and reuse of ML models for recasting - discussion	*All*
30/7-018 - Kjell Johnsen Auditorium, CERN	17:55 - 18:25



all the major frameworks (Checkmate, GAMBIT ColliderBit, MadAnalysis5, Rivet) have been developing interfaces for using the available ML models.

Krzysztof Rolbiecki on ATLAS-SUSY-2018-22 implementation in Checkmate

BDT validation

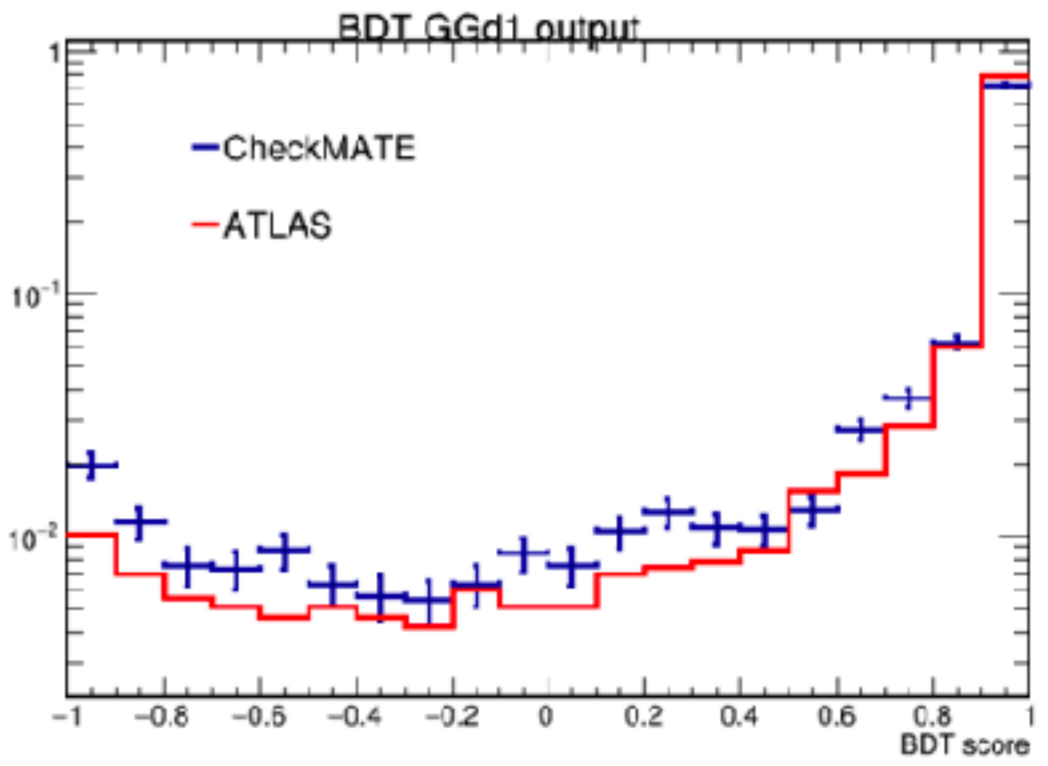
- Each SR targets direct gluino decays for specific range in $\Delta m = m_g - m_{neut}$
- GGd1: Δm = 1600-1900 GeV
- GGd2: Δm = 1000-1400 GeV
- GGd3: Δm = 600-1000 GeV
- GGd4: Δm = 200-600 GeV
- Overall, very good agreement

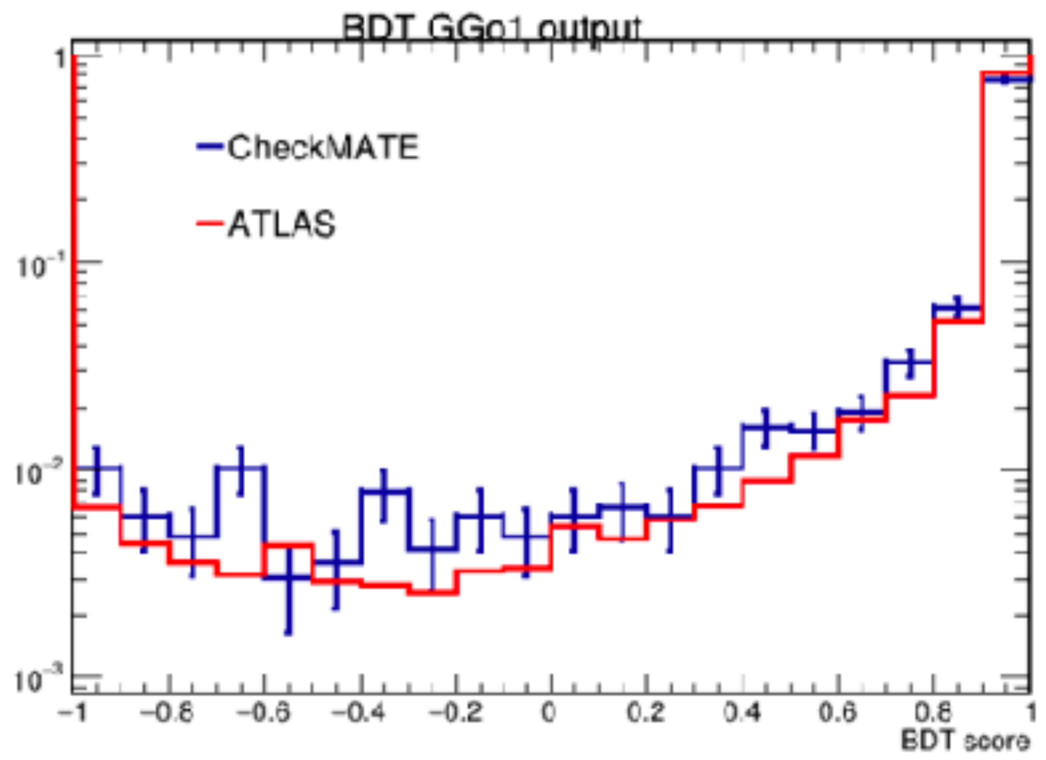
$m_{\tilde{g}}$ / $m_{\tilde{\chi}_1^0}$	2200 / 500		2200 / 1000		1800 / 1000		1400 / 1000	
	A	C	A	C	A	C	A	C
GGd1	14.1	12.5	7.04	5.5	5.5	4.2	3.0	3.0
GGd2	14.3	13.4	11.4	10.1	19.4	14.3	8.8	9.9
GGd3	14.4	14.1	14.4	13.8	71.7	62.0	49.1	43.7
GGd3	2.9	3.4	6.0	6.1	60.5	54.0	89.6	85.8

A = ATLAS; C = CheckMATE

"very good agreement"

BDT output comparison





- Both signal regions show good agreement
- GGd1 = direct decay; GGo1 = one step decay

BDT weights released as XML files for use with Root TMVA

10 -12 input variables: jet $p_T, \eta, E_T^{miss}, m_{eff}$, Aplanarity

…. and on ATLAS-SUSY-2018-30 implementation

SUSY search, gluinos, many b-jets + MET

- 8 NN signal regions: 4 for gluino decaying to top pair and 4 for gluino decaying to bottom pair (still it is one net)

ANA-SUSY-2018-30_config.json

ANA-SUSY-2018-30_model.onnx

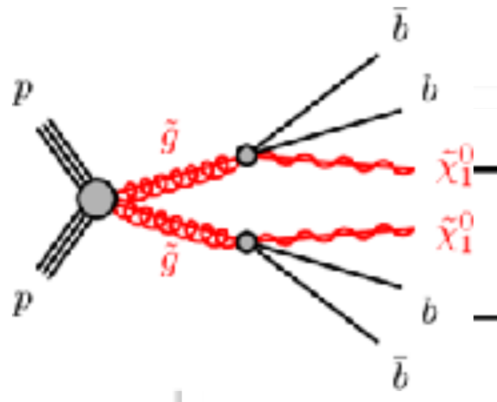
https://gitlab.cern.ch/atlas-sa/simple-analysis

- 87 input parameters: jet (small and large R) momenta, lepton momenta, MET and b-tag category (binary)

- output gives separate background and signal probabilities

→ very useable, but would be good to have plots for validation

Krzysztof Rolbiecki



	ATLAS	CheckMATE
Gtt selection		
Common rquirem.	7.66	7.30
SR-Gtt-2100-1	2.63	1.94
SR-Gtt-1800-1	2.80	2.11
SR-Gtt-2300-1200	2.95	2.62
SR-Gtt-1900-1400	0.19	0.27
Gbb selection		
Common rquirem.	80	65
SR-Gbb-2800-1400	22	14
SR-Gbb-2300-1000	21	14
SR-Gbb-2100-1600	6.20	6.80
SR-Gbb-2000-1800	0.19	0.58

"reasonable agreement across all channels"

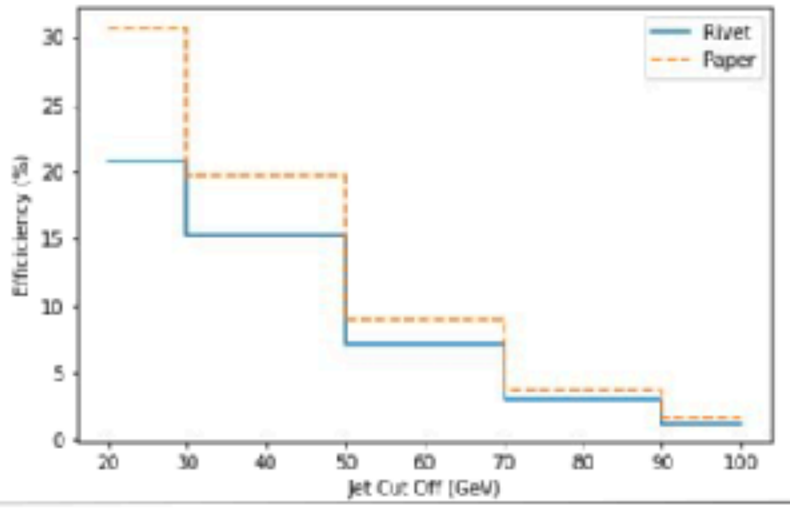
SUSY-2019-04 (RPV SUSY search) ML-model proved difficult

- One network for each case 4jets-8jets
- 65 input variables - mix of event information (H_T, similar), and specific jet/lepton information (e.g. p_T, η, φ, btag for lead 10 jets)
- Includes pseudo-continuous b-score for jets?!
  - Detector level.
  - simpleAnalysis suggests using 5, 1 or 0 for truth level data.
  - Paper notes this was the second most significant variable?!
- Paper describes three layer DNN:
  - But interrogating the file it seems a lot more complex - ONNX

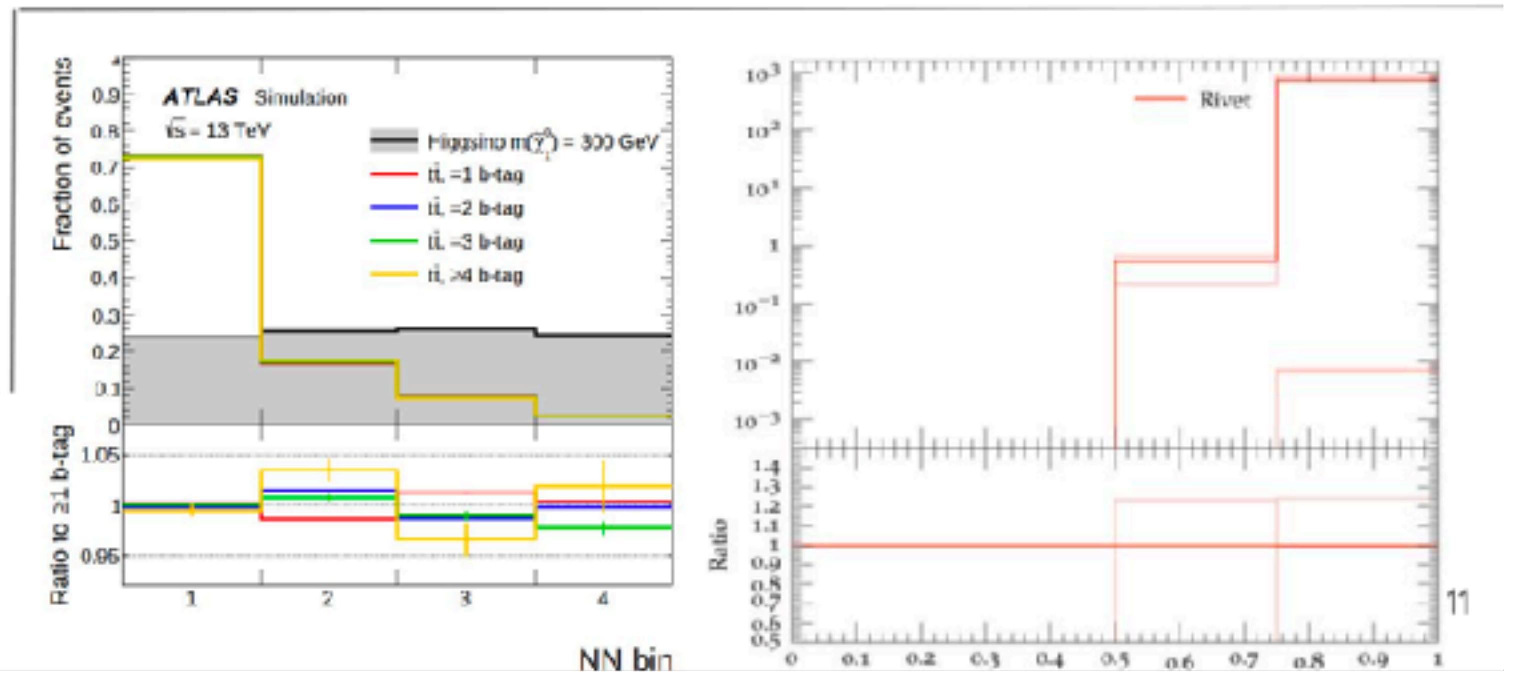
Tomasz Procter

Rivet Implementation – Validation

- Cutflows:
  - Not enough leptons - 22% vs 37% of events pass 1 lep > 27GeV.
  - Too many events passing NN cut.
  - But shapes consistent once you adjust for the leptons.



- Reproduction of Figure 2:



Tomasz Procter, RIF, December 2022

In the Reinterpretation Forum

7th RiF workshop 12–15 Dec 2022 at CERN

Session on publication and reuse of ML models

Introduction	*Sabine Kraml*	
30/7-018 - Kjell Johnsen Auditorium, CERN		16:30 - 16:35
Machine learning model serialization experiences	*Dan Guest*	
30/7-018 - Kjell Johnsen Auditorium, CERN		16:40 - 16:55
Reusing Neural Networks: Lessons learned and Suggestions for the future	*Tomasz Procter*	
30/7-018 - Kjell Johnsen Auditorium, CERN		17:00 - 17:15
Implementation of ML searches in CheckMATE	*Krzysztof Rolbiecki*	
30/7-018 - Kjell Johnsen Auditorium, CERN		17:20 - 17:35
CMS inputs on ML models re-usability	*Jennifer Ngadiuba*	
30/7-018 - Kjell Johnsen Auditorium, CERN		17:40 - 17:50
Publication and reuse of ML models for recasting - discussion	*All*	
30/7-018 - Kjell Johnsen Auditorium, CERN		17:55 - 18:25

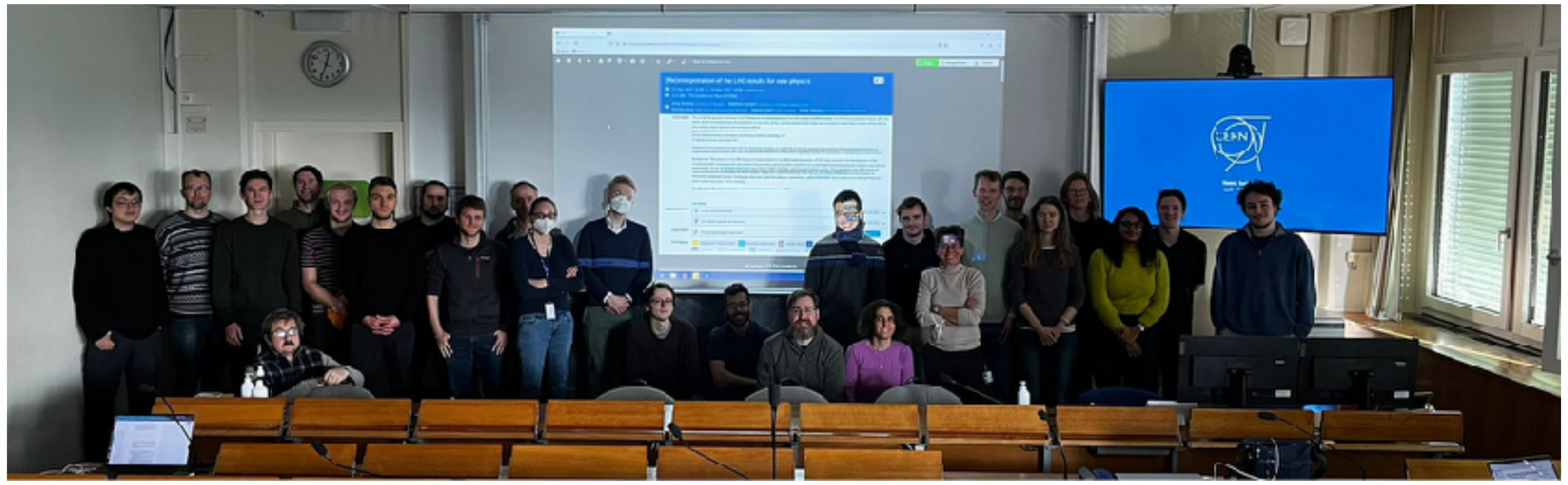
all the major frameworks (Checkmate, GAMBIT ColliderBit, MadAnalysis5, Rivet) have been developing interfaces for using the available ML models.

lessons learned

Les Houches PhysTeV workshop
June 2023

8th RiF workshop 29 Aug—1 Sep 2023 at Durham University

Recastable ML: guidelines, surrogate models and all that	*Tomasz Procter*	
PH8 (James Duff Lecture Theatre), Durham University		09:00 - 09:20



Guidelines for reusable ML models

Analysis Design	choice of framework, preservation format, architecture, input features
Documentation	clear definition of all input & output variables; code/framework version and dependencies
Validation	material enabling to verify performance (cut-flows, plots of in/out variables, runcards)
Surrogates	another ML model trained to approx. replicate the output of the original one (or simple parametrised efficiencies)

RiF & Les Houches 2023

Analysis Design Guidelines

- ☑ Use an **open-source framework** (tensorflow, pytorch, etc.)
  - Proprietary packages, such as NeuroBayes or Matlab-based packages, can make reuse difficult
- ☑ Ensure the network or tree can be saved in a **useful preservation format** for inference (e.g. ONNX or lwtnn).
  - Just leaving a `.h5` file or `.pkl` file is unlikely to be stable
- ☑ Be considerate with **choice of inputs** (can they be reproduced?)
  - Tomasz: "If a tagger depends entirely on detector level inputs, that's fine (but please provide detailed efficiencies – including misstags – or surrogates), but 10 truth-level quantities + pseudo-continuous b-score is frustrating."
- ☑ **Avoid over-complexity** in the network design - heavily customised layers or activation functions, e.g. TensorFlow lambdas, may not be well preserved (test!)

Documentation Guidelines

- ☑ Like for variables in any other analysis, we need **full definitions of all variables** that go into and come out of the ML model.

- ☑ Definitions include:
  - **Units** (GeV vs MeV, …)
  - **Normalisations**
  - **Phi conventions**: $[0, 2\pi]$ vs $[-\pi, \pi]$
  - **Input and output ordering**
  - …

- ☑ A validated **analysis code** (rivet, simpleAnalysis) automatically supplies much of this information.

- ☑ A short **explanatory note** uploaded alongside the ML model (e.g., in the form of a `README` file) is always a good idea; include all relevant **version info**!

nb. ONNX interpreter must match ONXX version

Validation Guidelines

- ☑ Where cuts depend on the ML model output, like for every other cut-based analysis, setp-by-step **cutflows** are a vital validation tool.
  - cut-flow information both before and after any ML-based selections
- ☑ **Plots of input and output variables** for validation samples (especially for most important features) are also useful.
- ☑ Full details of the **physics models** used to generate the information above are essential for any serious validation, e.g. **SLHA files and generator run cards**, or directly event samples
- ☑ Some understanding of feature importance is not only physically interesting, but can be essential in debugging.

Efficiencies and Surrogate Models

If an ML-model requires very experiment-specific inputs which cannot be reproduced outside the collaboration (low-level detector quantities, hits, tracks, …)

☑ If possible, provide **parametrised efficiencies** in terms of physics quantities accessible in simulation outside the collaboration

☑ **Train *another* network** approx. replicating the output of the original one

- can use truth-, parton- or reco-level inputs
- mimic output score of original model case by case
- need to determine level of accuracy of the surrogate

May or may not have access to the "true" answer (e.g. does the jet really contain a top quark?).

Same analysis design, documentation and validation guidelines as above apply

Conclusions

- Publishing ML models for reuse by others is possible and useful
- Successful examples exist from ATLAS analyses, ☚ thank you!
  already used in public reinterpretation frameworks
- Keep publication and reuse in mind early on in the analysis design — see guidelines

writeup in progress — comments & contributions welcome

Les Houches guide to reusable ML models in LHC analyses

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

https://www.overleaf.com/read/qvfyyyfpqbgn#09c3d0

Conclusions

- Publishing ML models for reuse by others is possible and useful
- Successful examples exist from ATLAS analyses
  already used in public reinterpretation frameworks
- Keep publication and r... ysis design — see guidelines...

Try it with the models you're developing in the HiggsML uncertainty challenge

...progress — comments & contributions welcome

Les Houches guide to reusable ML models in LHC analyses

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

https://www.overleaf.com/read/qvfyyyfpqbgn#09c3d0


Thanks for your attention

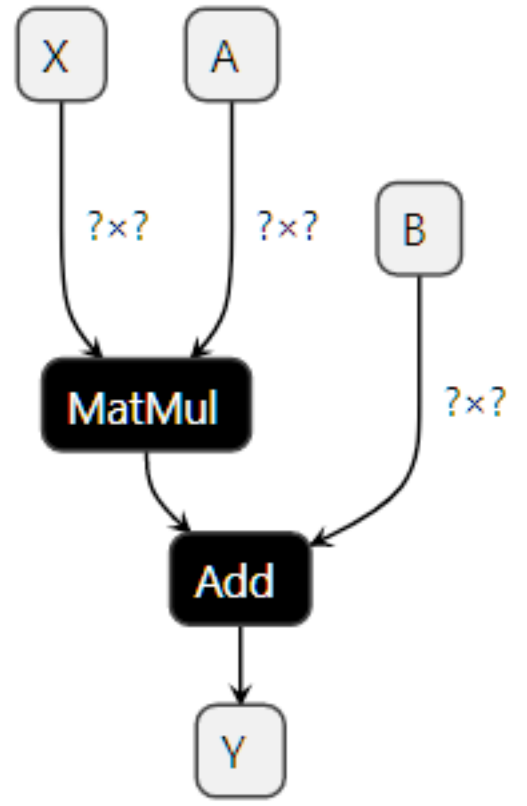
Open Neural Network Exchange

https://onnx.ai

"defines all the necessary operations a machine learning model needs to implement its inference function"

- ONNX is an open format built to represent ML models.
  - aims at providing a common language any ML framework can use to describe its models.
  - makes it possible to deploy a model **independent from the learning framework** used to build it.

  The deployment of a ML model usually requires replicating the entire ecosystem used to train the model, most of the time with a *docker*. Once a model is converted into ONNX, the production environment only needs a runtime (C, java, python, javascript, ….) to execute the graph defined with ONNX operators.



- Converters exist for scikit-learn, tensorflow, pytorch, and others

  NB must be updated every time ONNX or the library they support have a new released version.

- Beware of custom layers, experimental features, etc.!

  may be troublesome for converter and/or runtime (interpreter)

Runtime (interpreter) must match ONXX version
→ possible issue for preservation?

Ltwnn

Lightweight Trained Neural Network

David Hohn > lwtnn

build passing | coverity passed | DOI 10.5281/zenodo.597221

What is this?

The code comes in two parts:

1. A set of scripts to convert saved neural networks to a standard JSON format
2. A set of classes which reconstruct the neural network for application in a C++ production environment

The main design principles are:

- **Minimal dependencies:** The C++ code depends on C++11, Eigen, and boost PropertyTree. The converters have additional requirements (Python3 and h5py) but these can be run outside the C++ production environment.
- **Easy to extend:** Should cover 95% of deep network architectures we would realistically consider.
- **Hard to break:** The NN constructor checks the input NN for consistency and fails loudly if anything goes wrong.

We also include converters from several popular formats to the `lwtnn` JSON format. Currently the following formats are supported:

- Scikit Learn
- Keras (most popular, see below)

"Our underlying assumption is that *training* and *inference* happen in very different environments: we assume that the **training environment** is flexible enough to support modern and **frequently-changing libraries**, and that the **inference environment is much less flexible**."

see also talk by Dan Guest
at RiF 2022

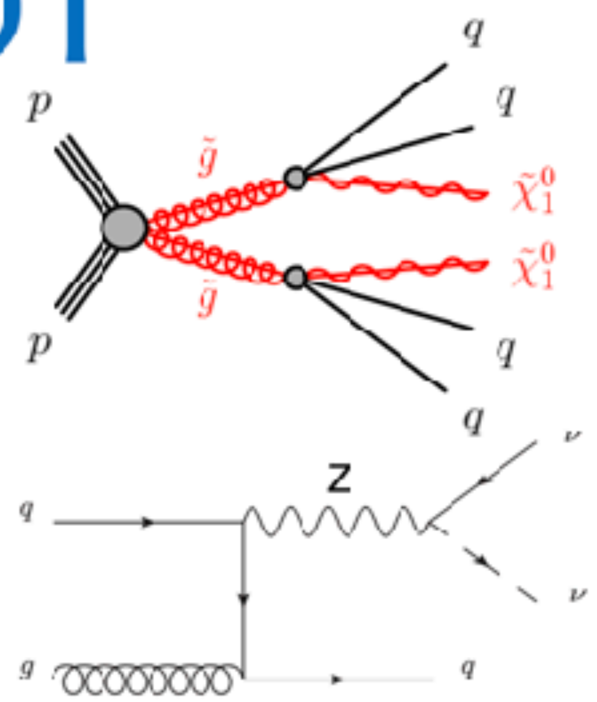
ATLAS-SUSY-2018-22

RAMP seminar by Kenta Uno, 9 April 2021

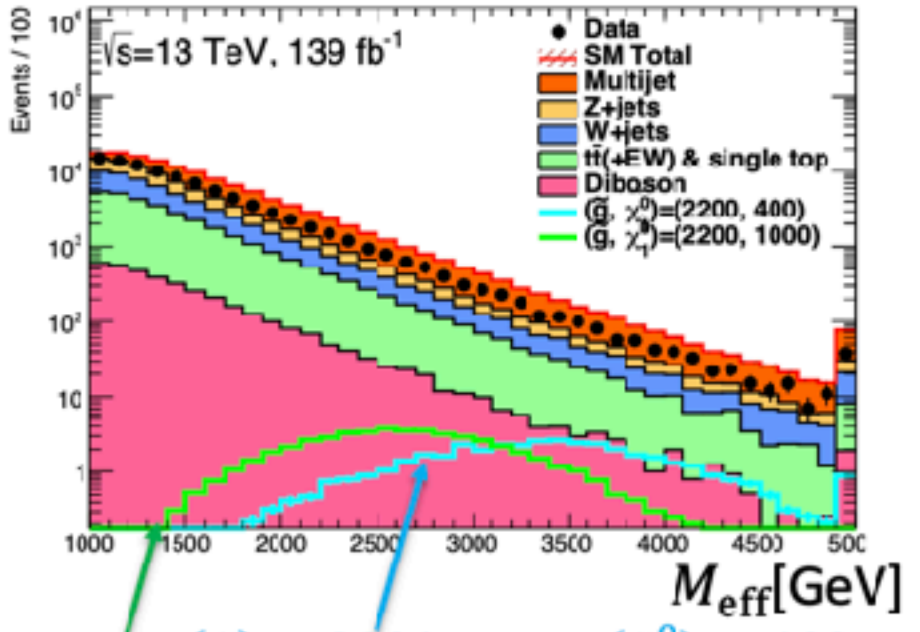
Analysis approach: BDT

Key variable in the previous analysis: M_{eff}

$$M_{\mathrm{eff}} = \sum_i |p_{\mathrm{T}}(i)| + E_{\mathrm{T}}^{\mathrm{miss}}$$



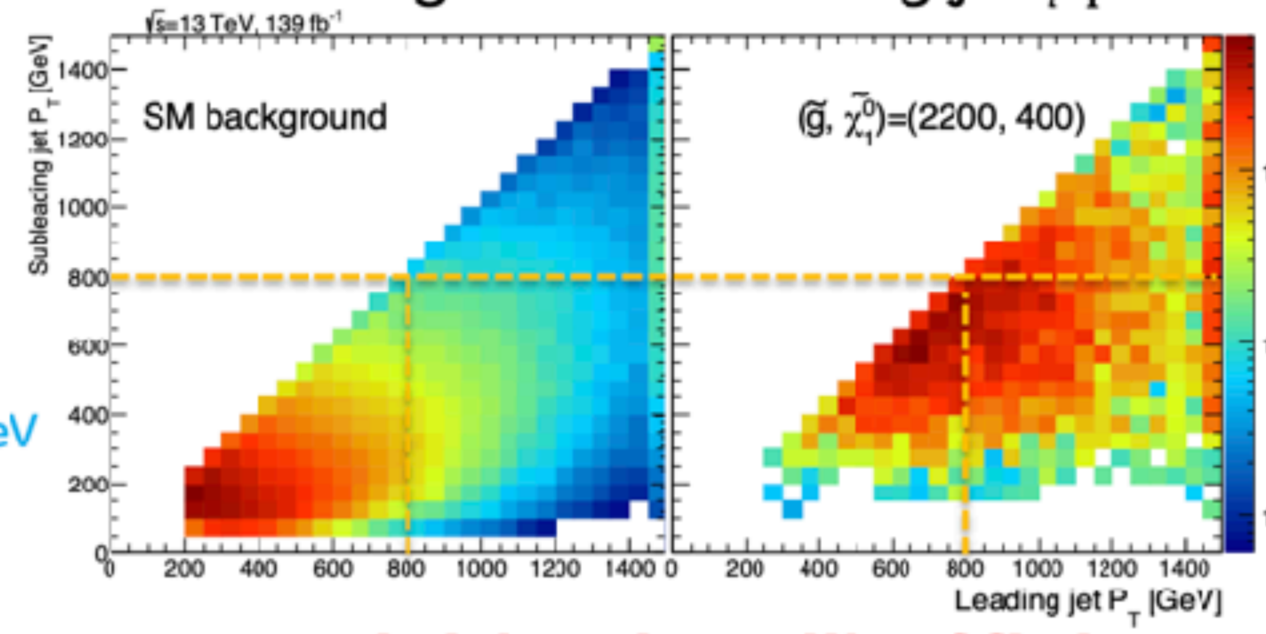
- However, it is not efficient as $\tilde{\chi}_1^0$ mass is large



$m(\tilde{g}) = 2200$ GeV, $m(\tilde{\chi}_1^0) = 400$ GeV

$m(\tilde{g}) = 2200$ GeV, $m(\tilde{\chi}_1^0) = 1000$ GeV

Leading and sub-leading jet p_{T}



Using correlation between variables is still efficient.

→ BDT analysis is useful for SUSY search

2021/4/9 5

10 -12 input variables: jet $p_{\mathrm{T}}, \eta, E_{\mathrm{T}}^{\mathrm{miss}}, m_{\mathrm{eff}}$, Aplanarity

Discussion point 1

- The auxiliary material provided for the BDT reinterpretation
  - We published HepData materials [Link]
    - Provided xml files of BDT classifiers
      → ZeroLepton2018-SRBDT-weight.tar.gz
    - Also provided analysis snipped code
      → ZeroLeptonBDT2018.cxx
      → By using this code, you can use same BDT classifiers.
    - Full likelihood is provided in the hepdata material

※ Signal events: fast sim (reco)
※ Background events: full reco

We provide data, SM total and signal yields in each BDT score bin and the acceptance and efficiencies of BDT SRs for simplified GG models.

→ I think it might help everyone..

2021/4/9 14

Reinterpretation material

Additional resources

C++ code snippet with the implementation of the analysis selection at truth-level

- Can be used with SimpleAnalysis framework

SLHA files for benchmark signals

ONNX files for the neural networks for the EWK analysis

- Not possible to use lwtnn because the architecture of one of the layers is not supported

Upcoming: **ROOT workspaces**, containing data and the fitted background model in all SRs

- not using pyhf as it doesn't support our parameterised background model