University of Glasgow

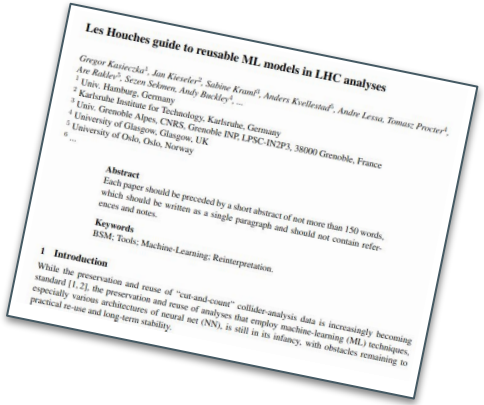
Reusing Neural Networks: Experiences and suggestions for EFT cases

Tomasz Procter,
LHC EFT WG Area 3,
October 2023

My background

- **NOT** an EFT specialist
- ATLAS experimentalist with a hand in some reinterpretation tools (Rivet, Gambit)
  - Have successfully reinterpreted a couple of ML-dependent analyses.
- Also some internal ATLAS experience (NNs in an exotics context)

- Les Houches guidelines on reinterpretable NN
  - These are really close to being done! Check them out here.
- Doing some work on surrogate taggers.



Les Houches guide to reusable ML models in LHC analyses

Gregor Kasieczka[1], Jan Kieseler[2], Sabine Kraml[3], Anders Kvellestad[4], Andre Lessa, Tomasz Procter[4], Are Raklev[5], Sezen Sekmen, Andy Buckley[4], ...

[1] Univ. Hamburg, Germany
[2] Karlsruhe Institute for Technology, Karlsruhe, Germany
[3] Univ. Grenoble Alpes, CNRS, Grenoble INP, LPSC-IN2P3, 38000 Grenoble, France
[4] University of Glasgow, Glasgow, UK
[5] University of Oslo, Oslo, Norway
[6] ...

Abstract

Each paper should be preceded by a short abstract of not more than 150 words, which should be written as a single paragraph and should not contain references and notes.

Keywords

BSM; Tools; Machine-Learning; Reinterpretation.

1 Introduction

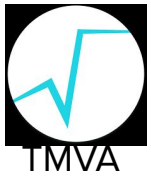
While the preservation and reuse of "cut-and-count" collider-analysis data is increasingly becoming standard [1, 2], the preservation and reuse of analyses that employ machine-learning (ML) techniques, especially various architectures of neural net (NN), is still in its infancy, with obstacles remaining to practical re-use and long-term stability.

Outline

- Why I’m talking to you
- Preserving NNs
- Reinterpreting NNs
  - What’s been tried
  - Successes
  - Future problems & solutions
- Direct application to EFT

Before reinterpretation: Preserving ML





- Many software tools, many output formats, many version dependencies…
- Raw .h5 files, pickle files etc not very stable
  - May not run the same in just a couple of years.
- ONNX/ONNXruntime - industry developed tool for sharing neural nets across architectures
  - Easy to produce from tf/keras and pytorch
- LWTNN - ATLAS trigger developed tool - good, but fewer active developers.
- Fully define inputs: ordering, units, padding etc. - it’s not preserved if this isn’t known!!!
- Probably quite broadly relevant - e.g. likelihoods stored as DNNs etc.

Initial experiences with reinterpretation

- So far: two publicly available LHC analysis NN networks - both from ATLAS SUSY:
  - ANA-SUSY-2019-04 (RPV SUSY search in lep+jets final state)
  - ANA-SUSY-2018-30 (gluino pair production in multi-b final states)
- This is a new type of experimental output - the experiments are still feeling out how to publicise this.
- ANA-SUSY-2018-30 has worked well in multiple frameworks (rivet, gambit, checkmate, ...).
- Several key features made this work:
  - Lots of extra info (ordering, units, usage example) - would have been impossible without SimpleAnalysis [1].
  - All inputs are easily accessible to reinterpretation tools
    - Lepton/jet kinematics, MET, btag yes/no
    - No detector-level variables (including continuous btag score)

Cut	Paper	Rivet
0-lep	80.0	83.7
$\Delta\phi^{4j}_{min}\geq 0.6$	52.5	54.6
2800-1400 NN Cut	21.7	23.9
$\Delta\phi^{4j}_{min}\geq 0.6$	52.5	54.6
2300-1000 NN Cut	21.3	23.3
$\Delta\phi^{4j}_{min}\geq 0.4$	61.1	63.8
2100-1600 NN Cut	6.20	6.50
$\Delta\phi^{4j}_{min}\geq 0.4$	61.1	63.8
2000-1800 NN Cut	0.192	0.204

[1] Simplified ATLAS SUSY analysis framework

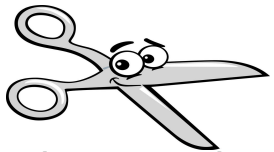
How it gets harder...

- Choice of inputs *really* matters.
- Trend in Physics+ML research is to move towards more and more detector level quantities – these may require full detector simulation – almost impossible in reinterpretation tools
- Four possible solutions, depending on exact circumstance:

How it gets harder...

- Choice of inputs *really* matters.
- Trend in Physics+ML research is to move towards more and more detector level quantities – these may require full detector simulation – almost impossible in reinterpretation tools.
- Four possible solutions, depending on exact circumstance:

Solution 1: Eliminate

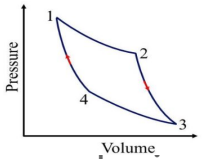


- If only one or two detector-level quantities are used, does the NN really need them as inputs?
- E.g. Jet kinematics + continuous b-tag score: can we just use b-tag true/false?

How it gets harder...

- Choice of inputs *really* matters.
- Trend in Physics+ML research is to move towards more and more detector level quantities – these may require full detector simulation – almost impossible in reinterpretation tools
- Four possible solutions, depending on exact circumstance:

Solution 2: Efficiencies

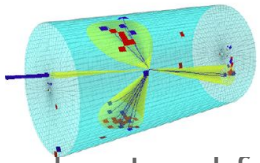


- In some situations (parameterised) efficiencies can replace ML functioning as a tagger.
- Parameterising can make this approach significantly more effective.
- **NOT** suitable for e.g. BSM signal vs background discriminators - if we train on one BSM sample, extrapolating efficiencies to other BSM models is wild...

How it gets harder...

- Choice of inputs *really* matters.
- Trend in Physics+ML research is to move towards more and more detector level quantities – these may require full detector simulation – almost impossible in reinterpretation tools
- Four possible solutions, depending on exact circumstance:

Solution 3: Simulate

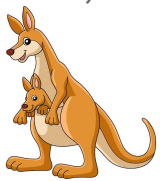


- Not my field, but I understand fast detector-sim is getting better and faster.
- Can it eventually get to the point we can use it in reinterpretation?
- Would probably require a lot more openness from the experiments with their detector sim.
- Probably not there yet, but worth checking back in.

How it gets harder...

- Choice of inputs *really* matters.
- Trend in Physics+ML research is to move towards more and more detector level quantities – these may require full detector simulation – almost impossible in reinterpretation tools
- Four possible solutions, depending on exact circumstance:

Solution 4: Surrogates



- Train network to learn output of detector-level neural net using truth-level inputs.
- Truth-level inputs may include “cheat” information - like presence of a top or bottom in the jet.
- Early stages, but promising.
- We also have a Les Houches project leveraging CMSOpenData - let me know if you’re interested!

How does this affect EFTs

- Reinterpretation of unfolded measurements - basically unaffected (even if ML used before unfolding)
- Reinterpretation of dedicated EFT searches - there can be direct or indirect dependence on ML:
  - Full dependence - e.g. NN/BDT discriminators to define signal regions, ML-based optimal observables.
  - Indirect - E.g. a Cut'n'count analysis that has a cut based on a custom jet-tagger.
- Key rule:

  Reinterpretation is easiest when the analysis team think about it from the start
  - Make sure models can be saved in a preservable format.
  - Example code snippets, metadata is very important.
  - Think about choice of inputs:
    - Do we need to use efficiencies/surrogates instead?

BONUS

Summary of Les Houches wishlist - Analysis Design, Implementation and Validation

(see full document for more detail)

Analysis Design

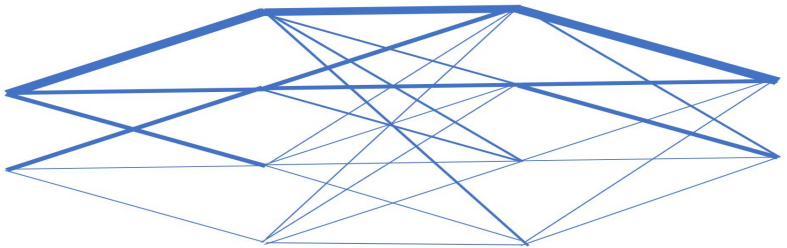
- Use an open-source framework (tensorflow, pytorch, etc)
- Ensure the network can be saved in a useful preservation format for inference (e.g. ONNX or lwtnn).
  - Just leaving in a `.h5` file or `.pkl` file is unlikely to be stable
- Be considerate with choice of inputs - if a tagger depends entirely 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.

Supplementary Material

- Like for variables in any other analysis, we need full definitions of all variables that go into and come out of the neural net.
  - This is even more important given the “black box” nature of a neural network.
- Definitions include:
  - Units (MeV v GeV)
  - Normalisations
  - Padding values
  - Phi conventions (0 – 2π vs $-\pi - \pi$)
  - Input and output ordering
- A **validated** analysis code (rivet, simpleAnalysis) automatically supplies much of this info.
- Otherwise, a short, minimal note might need to be uploaded alongside the onnx/lwtnn file.

Validation Material

- Where cuts depend on a neural net output, like for every other cut-based analysis, cutflows are a vital validation tool.
- Input and output plots (especially for most important features) could also be useful.
- When cutflows or extra plots are provided, just like for any other analysis, we really need to know the exact signal model that produced them (slha files, generator run cards, etc)
- Some understanding of feature importance is not only physically interesting, but can be essential in debugging.



What is a surrogate?

- What to do if an ML-model requires very experiment specific inputs (e.g. hits, exact btagging scores)?
  (There are more and more such networks being used in analyses)
- Train *another* network:
  - Given truth/reinterpretation level inputs
  - Mimic output score of original model case by case
  - Probably with some randomness built in
- May or may not have access to the “true” answer (e.g. does the jet really contain a top quark?).
- If yes, effectively “parametrised efficiency on steroids”

What is a surrogate?

- What to do if an ML-model requires very experiment specific inputs (e.g. hits, exact btagging scores)?
  (There are more and more such networks being used in analyses)
- Train *another* network:
  - Given truth/reinterpretation level inputs
  - Mimic output score of original model case by case
  - Probably with some randomness built in
- May or may not have access to the “true” answer (e.g. does the jet really contain a top quark?).
- If yes, effectively “parametrised efficiency on steroids”

Surrogates - what next?

- No such surrogate networks currently available.
- Les Houches working group “proof of concept” project on CMS run-1 btagger using CMSOpenData.
- Would be great to see something originating in the experiments (who have all the data) - could be a really cool qualification task or similar.

- If you have a pet generative ML model you would like to throw at this, let me know!

More SUSY-2018-30 validation plots