University of Glasgow

Recasting hadron collider searches containing neural networks:
Experiences and suggestions

Tomasz Procter,
Belle II physics week,
October 2024

My background

- **NOT** a Belle II/tau/dark-sector 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 ML
  - arXiv:2312.14575
- Done some work on surrogate taggers.

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*[8], *Are Raklev*[6], *Humberto Reyes-Gonzalez*[6,9,10], *Krzysztof Rolbiecki*[11], *Sezen Sekmen*[12], *Gokhan Unel*[13]

[1] Jefferson Lab, Newport News, VA 23606, USA
[2] University of Glasgow, Glasgow, UK
[3] Univ. Hamburg, Germany
[4] Karlsruhe Institute for Technology, Karlsruhe, Germany
[5] Univ. Grenoble Alpes, CNRS, Grenoble INP, LPSC-IN2P3, 38000 Grenoble, France
[6] University of Oslo, 0316 Oslo, Norway
[7] Universidade Federal do ABC, Santo André, 09210-580 SP, Brazil
[8] Department of Physics, University of Genova, Via Dodecaneso 33, 16146 Genova, Italy
[9] INFN, Sezione di Genova, Via Dodecaneso 33, I-16146 Genova, Italy
[10] Institut für Theoretische Teilchenphysik und Kosmologie, RWTH Aachen, 52074 Aachen, Germany
[11] Faculty of Physics, University of Warsaw, 02-093 Warsaw, Poland
[12] Department of Physics, Kyungpook National University, Daegu, South Korea
[13] U.C. Irvine, Physics & Astronomy Dept., Irvine, CA, USA

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.

Outline

- How (and why) we recast BSM searches at the LHC
- Preserving NNs
- Successes: ATLAS SUSY 2018 30
- Future problems & solutions
- Very brief b-factory observations
- Conclusions

LHC Recasting

Hadron Collider BSM Reinterpretation/Recasting

- Generate BSM events.
- Detector simulation/emulation:
  - Combined sim+reco
  - Delphes/4-vector smearing
  - Use (parameterised) efficiencies for e.g. b-tagging.
- Run events through a simplified analysis code:
  - i.e. calculate complex variables, apply cuts
  - Rivet, CHECKMate, ColliderBit, MadAnalysis5...
- Get yields from signal regions:
  - Compare to experimental results; calculate likelihoods: ideally pyhf or similar.
- Why?
  - Ensure the LHC legacy for decades to come: New, better, BSM models will be of interest.
  - Theorists want to test more models than the experiments can handle (and many will be “already excluded”)

Hadron Collider BSM Reinterpretation/Recasting

- Generate BSM events.
- Detector simulation/emulation:
  - Combined sim+reco
  - Delphes/4-vector smearing
  - Use (parameterised) efficiencies for e.g. b-tagging.
- Run events through a simplified analysis code:
  - Rivet, CHECKMate, ColliderBit, MadAnalysis5...
  - i.e. calculate complex variables, apply cuts
- Get yields from signal regions:
  - Compare to experimental results; calculate likelihoods: ideally pyhf or similar.

These are trained on detector-level data: not crazy to wonder if too much information is lost at truth- or detector-emulation- level

Experiments use ML here...
Particle ID:
- Track finding, ...
- Quark v gluon, b-hadrons, etc.

... and here:
- S vs B discriminators
- High-level jet taggers

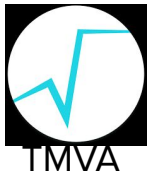
Preserving NNs

Preserving & sharing ML: why?

- So you've trained an ML model that does something really useful in your experimental context… what next?
- Releasing example datasets, training notebooks to show off new architectures:
  - ML community quite good at this!
- But what if someone needs to reuse your *exact* network/BDT for inference?
  - Or training dataset too big/experimentally restricted?
  - E.g. the NN defines an important function
  - This is an increasingly common problem for collider physics

Preserving & sharing ML: problems



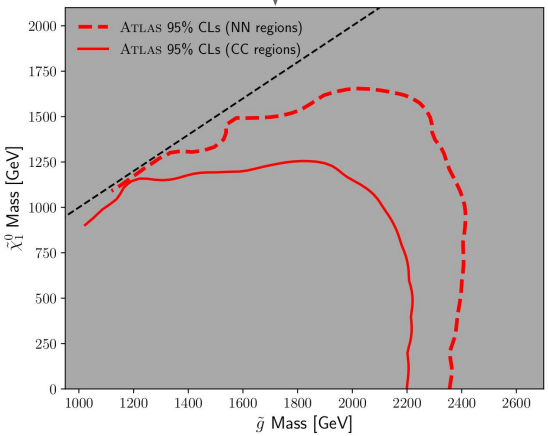


- Many software tools, many output formats, many version dependencies…
- Raw .h5 files, pickle files etc not very stable
  - May not run the same in just a couple of years.
- ONNX/ONNXruntime - industry developed tool for sharing neural nets across architectures
  - Easy to produce from tf/keras and pytorch
  - Latest/most cutting edge generative architectures may lag slightly.
- 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!!!

LHC Experiences

One (successful) example: ATLAS-SUSY-2018-30

- ATLAS search for gluino pair-production in multi-b final states
  - arXiv:2211.08028
- Some NN-based SRs, some conventional “cut-and-count” SRs.
  - NN-based regions significantly more sensitive
- After preselection cuts, SRs defined by S vs B NN discriminant:
  - Relatively simple DNN
  - Inputs:
    - Kinematics (and b-tagging info) of leading 10 R=0.4 jets
    - Kinematics (and b-tagging info) of leading 4 R=1.0 jets
    - Kinematics of leading 4 leptons
    - “Event-level variables”: MET, metphi, effective- and transverse masses, ++



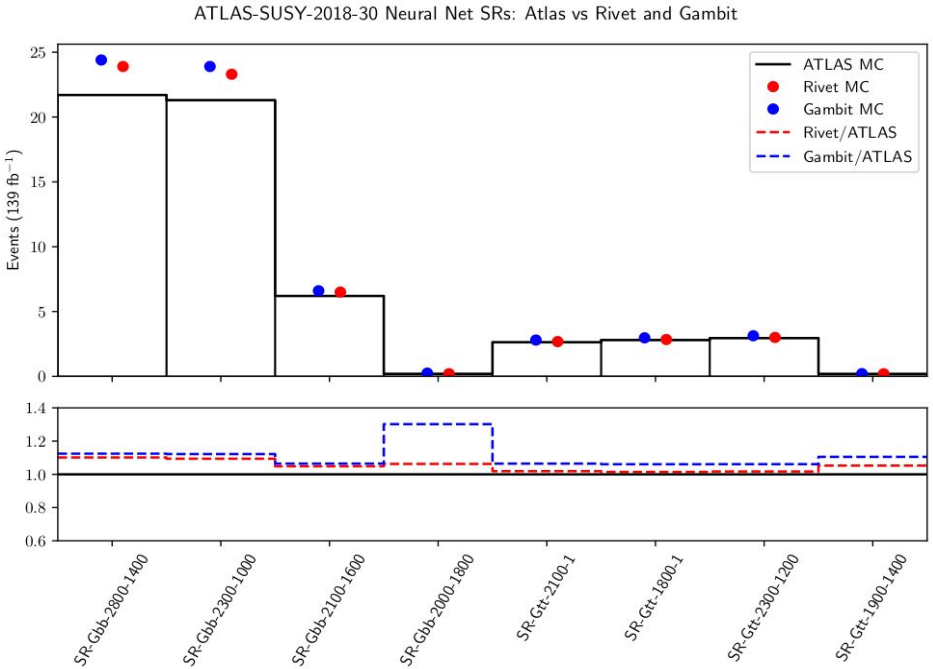
One (successful) example: ATLAS-SUSY-2018-30

- Reinterpretation *much* easier thanks to SimpleAnalysis [1] -
  - Even if not used directly, provides pseudo-code description of analysis procedure.
  - Extra important for Neural Nets because of “blackbox” nature:
    - Units
    - Padding
    - Normalisation
    - Phi-conventions ...
- ONNX files made publicly available via SimpleAnalysis
  - Great! But would be good if all experiments could standardise (e.g. Hepdata) going forward.
- Became a test-case for reinterpretation tools

[1] Simplified ATLAS SUSY analysis framework

One (successful) example: ATLAS-SUSY-2018-30

- Really good results obtained by multiple tools: replication as good as C&C regions.

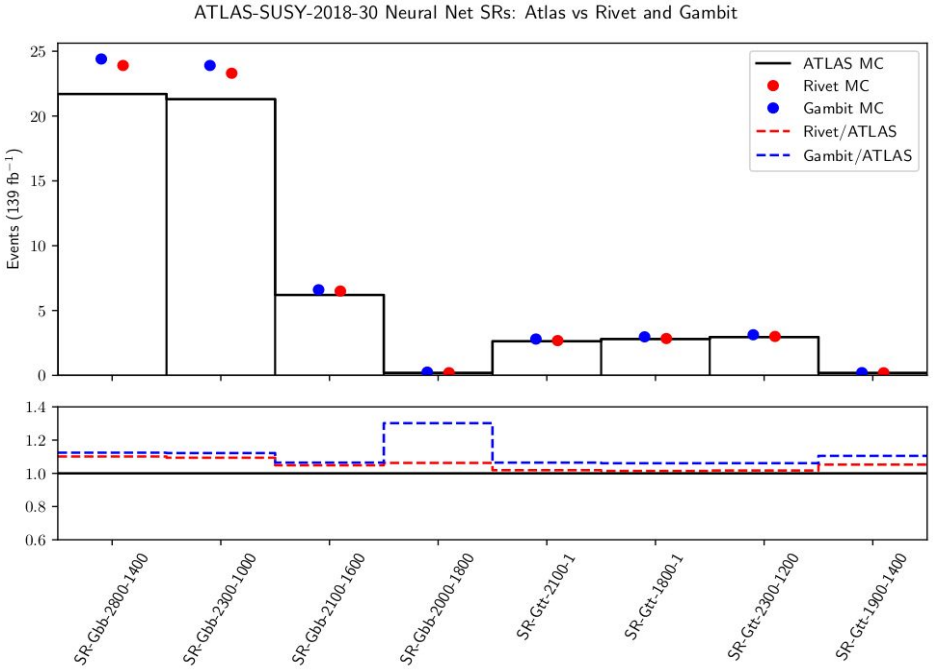


Event Count

Region	Selection	Paper (ATLAS)	RIVET	Paper (ATLAS) %	RIVET %
Common Gbb	$N_{\text{lep,base}} = 0$	80.0	83.7*	-	-
SR-Gbb-2800-1400	$\Delta\phi^{4j}_{\text{min}} \geq 0.6$	52.5	54.6	66%	65%*
	P(Gbb) ≥ 0.999	21.7	23.9	41%	44%
SR-Gbb-2300-1000	$\Delta\phi^{4j}_{\text{min}} \geq 0.6$	52.5	54.6	66%	65%*
	P(Gbb) ≥ 0.9994	21.3	23.3	41%	43%
SR-Gbb-2100-1600	$\Delta\phi^{4j}_{\text{min}} \geq 0.4$	61.1	63.8	76%	76%*
	P(Gbb) ≥ 0.9993	6.20	6.50	10%	10%
SR-Gbb-2000-1800	$\Delta\phi^{4j}_{\text{min}} \geq 0.4$	61.1	63.8	76%	76%*
	P(Gbb) ≥ 0.997	0.192	0.204	3.1%	3.2%
Common Gtt	$N_{\text{lep,sig}} = 1$ or ($N_{\text{lep,base}} =$ 1 and $\Delta\phi^{4j}_{\text{min}} \geq 0.4$)	7.66	8.1*	-	-
SR-Gtt-2100-1	P(Gtt) ≥ 0.9997	2.63	2.68	34%	33%
SR-Gtt-2100-1	P(Gtt) ≥ 0.9997	2.80	2.84	37%	35%
SR-Gtt-2100-1	P(Gtt) ≥ 0.9997	2.95	3.00	39%	37%
SR-Gtt-2100-1	P(Gtt) ≥ 0.9997	0.19	0.20	2.5%	2.5%

One (successful) example: ATLAS-SUSY-2018-30

- Really good results obtained by multiple tools: replication as good as C&C regions.

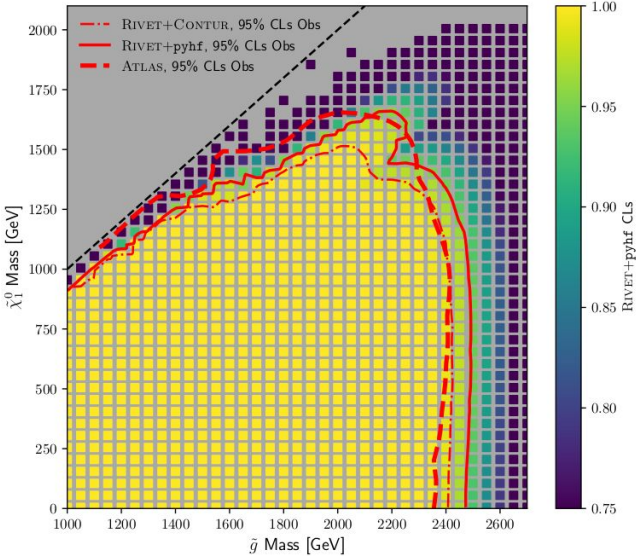


Event Count

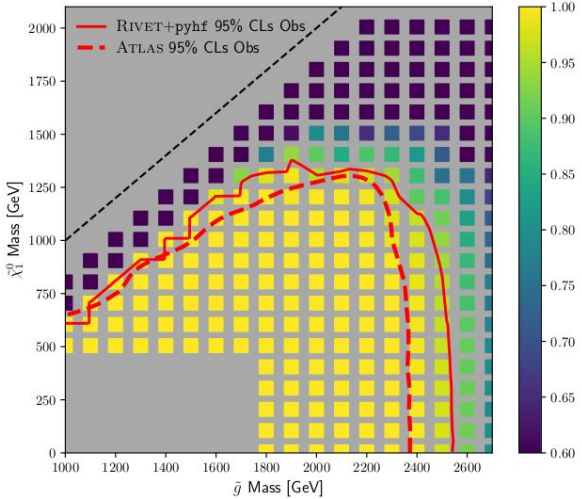
Region	Selection	Paper (ATLAS)	RIVET	Paper (ATLAS) %	RIVET %
Common Gbb	$N_{\text{lep,base}} = 0$	80.0	83.7*	-	-
SR-Gbb-2800-1400	$\Delta\phi^{4j}_{\text{min}} \geq 0.6$	52.5	54.6	66%	65%*
	P(Gbb) ≥ 0.999	21.7	23.9	41%	44%
SR-Gbb-2300-1000	$\Delta\phi^{4j}_{\text{min}} \geq 0.6$	52.5	54.6	66%	65%*
	P(Gbb) ≥ 0.9994	21.3	23.3	41%	43%
SR-Gbb-2100-1600	$\Delta\phi^{4j}_{\text{min}} \geq 0.4$	61.1	63.8	76%	76%*
	P(Gbb) ≥ 0.9993	6.20	6.50	10%	10%
SR-Gbb-2000-1800	$\Delta\phi^{4j}_{\text{min}} \geq 0.4$	61.1	63.8	76%	76%*
	P(Gbb) ≥ 0.997	0.192	0.204	3.1%	3.2%
Common Gtt	$N_{\text{lep,sig}} = 1$ or ($N_{\text{lep,base}} =$ 1 and $\Delta\phi^{4j}_{\text{min}} \geq 0.4$)	7.66	8.1*	-	-
SR-Gtt-2100-1	P(Gtt) ≥ 0.9997	2.63	2.68	34%	33%
SR-Gtt-2100-1	P(Gtt) ≥ 0.9997	2.80	2.84	37%	35%
SR-Gtt-2100-1	P(Gtt) ≥ 0.9997	2.95	3.00	39%	37%
SR-Gtt-2100-1	P(Gtt) ≥ 0.9997	0.19	0.20	2.5%	2.5%

One (successful) example: ATLAS-SUSY-2018-30

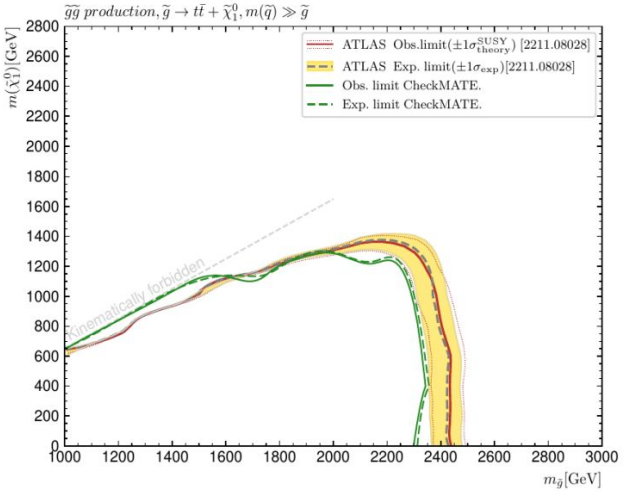
- Really good results obtained by multiple tools: just as good as C&C regions.



Rivet, sbottom-sbottom model



Rivet, stop-stop model
(mildly suspect event generation)



CheckMATE, stop-stop model, from
K. Rolbiecki, Grenoble, June 2024

BUT: not all examples have worked

- Have been other successes (particularly for BDTs), but this is not universal:
- ATLAS-SUSY-2019-04 – multijet RPV SUSY (arXiv:2106.09609):
  - Neither Rivet nor CheckMATE got good results here
  - (Traditional cut'n'count regions didn't do great, either...)

Rivet Cutflow

Jet p_T threshold	20 GeV	
$1l$ channel	ATLAS	RIVET
$= 4$ jets, ≥ 4 b-tags, NN	0.02%	0.07%
$= 5$ jets, ≥ 4 b-tags, NN	0.09%	0.41%
$= 6$ jets, ≥ 4 b-tags, NN	0.11%	0.73%
$= 7$ jets, ≥ 4 b-tags, NN	0.07%	0.53%
$= 8$ jets, ≥ 4 b-tags, NN	0.03%	0.0%

Validation - cutflow

- Pretty much everything went wrong
- Clearly a problem with lepton id
- Too few events with high jet multiplicity

Slide from K. Rolbiecki, Grenoble, June 2024, describing CheckMATE implementation

- And many analyses depend on NNs/BDTs but don't publish anything.
  - This is also means recasting can't work.
  - Some analyses *can't* publish anything – proprietary software, poor record-keeping.

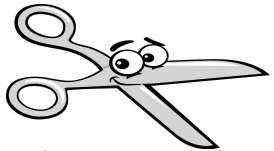
How it gets harder...

- 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

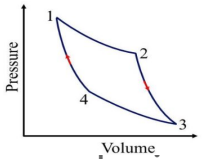


- If only one or two detector-level quantities are used, does the NN really need them as inputs?
- E.g. Jet kinematics + continuous b-tag score: can we just use b-tag true/false?

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

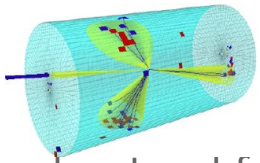


- In some situations (parameterised) efficiencies can replace ML functioning as a tagger.
- Parameterising can make this approach significantly more effective.
- **NOT** suitable for e.g. BSM signal vs background discriminators - if we train on one BSM sample, extrapolating efficiencies to other BSM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

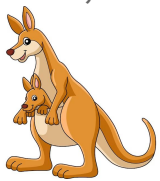


- Not my field, but I understand fast detector-sim is getting better and faster.
- Can it eventually get to the point we can use it in reinterpretation?
- Would probably require a lot more openness from the experiments with their detector sim.
- Probably not there yet, but worth checking back in.

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.
  - “Super-parameterised efficiency on steroids”
- See e.g. arxiv:2402.15558: early stages, but promising.

Belle II context

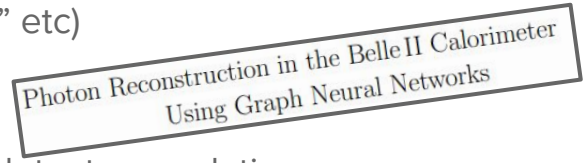
How does this affect Belle II?
(and dark-sector/tau in particular?)

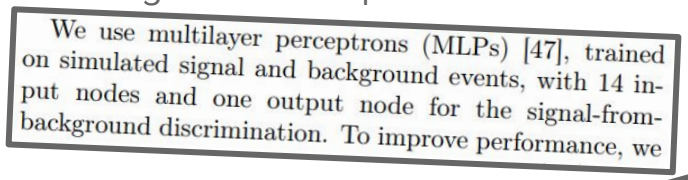
- Reinterpretation of unfolded measurements (CKM angles, BRs for rare hadrons) - basically unaffected (even if ML used before unfolding).
  - (Though such measurements can still be preserved!)
- However direct searches (e.g for ALPs, Z', A' etc, as we have discussed this week), appear (to me) a natural choice for recasting:
  - Detector-level
  - Could be sensitive to more general models
- We'd certainly be keen to see implementations in Rivet
  - Very explicitly a multi-collider framework (includes analyses from BaBar, Belle, Petra, ++), albeit mainly measurements.
  - Though "other reinterpretation frameworks are available"!
- And if ML used here, then you ***do*** need to worry about how to preserve the network.

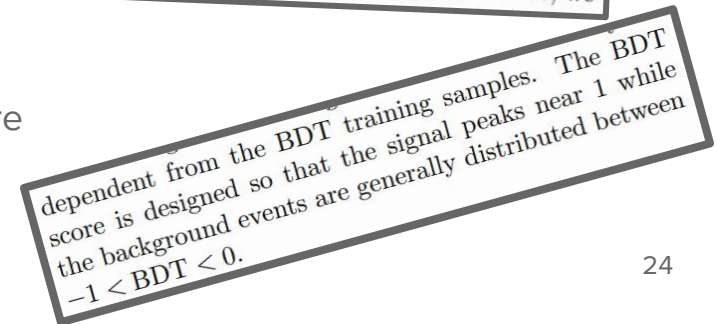
How does this affect Belle II?
Other BSM searches at B-factories and Belle II NNs

(From a very naïve inspire search of “Belle II” and “ML” or “Neural Net” etc)

- NN Trigger/Reco (arXiv:2306.04179), (arXiv:2402.14962)
  - In a potential recast, these would likely be absorbed into detector emulation.
- GNN Flavour ID (arXiv:2402.17260)
  - If used in a search, this is the sort of ML that needs to be thought about for preservation.
- Tau resonance search at Belle II (arXiv:2306.12294):
  - “Event levels” MLPs used for S vs B discriminations
- Dark photon search at BaBar (arXiv:1702.03327):
  - (See Stefania’s talk yesterday morning)
  - Signal/control region definition dependent on a BDT score


Photon Reconstruction in the Belle II Calorimeter Using Graph Neural Networks


We use multilayer perceptrons (MLPs) [47], trained on simulated signal and background events, with 14 input nodes and one output node for the signal-from-background discrimination. To improve performance, we


dependent from the BDT training samples. The BDT score is designed so that the signal peaks near 1 while the background events are generally distributed between $-1 < \mathrm{BDT} < 0$.

How does this affect Belle II?

Reinterpretation is easiest when the analysis team think about it from the start

- Make sure ML models can be saved in a preservable and re-usable format.
- Example code snippets, metadata are *very* important.
- Think about choice of inputs:
  - Do we need to use efficiencies/surrogates instead?

Conclusions

Conclusions

- We have made recasting ML-dependent BSM searches work at the LHC
- BUT: this requires help from the experiments:
  - Actually provide the networks!
  - And metadata/validation material even more important than normal

- If you want to recast BSM searches at Belle II that depend on ML:
  - It can be done!
  - Don’t wait until the analysis is done to start thinking about it:
    - (so now is the perfect time!)

BONUS

Summary of Les Houches "guidelines" - Analysis Design, Implementation and Validation

(see full document for more detail – arXiv:2312.14575)

Analysis Design

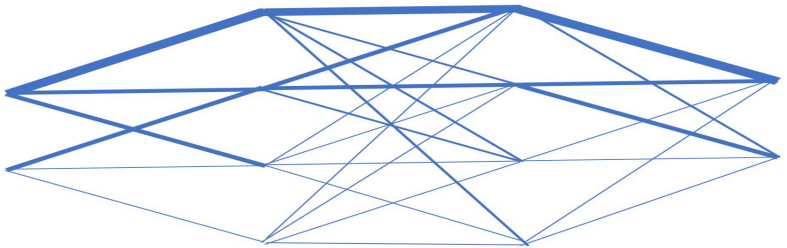
- Use an open-source framework (tensorflow, pytorch, etc)
- Ensure the network can be saved in a useful preservation format for inference (e.g. ONNX or lwtnn).
  - Just leaving in a `.h5` file or `.pkl` file is unlikely to be stable
- Be considerate with choice of inputs - if a tagger depends entirely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

LH Guidelines

- We would welcome feedback/suggestions from Belle II community, if there is something specific to your context that we did not cover