DASHH Data Science in Hamburg HELMHOLTZ Graduate School for the Structure of Matter

Dealing with negatively weighted Events in DNN-based LHC Analyses

Jörn Bach[1,2,3], Christian Schwanenberger[1,2], Peer Stelldinger[3], Alexander Grohsjean[2]

1 Deutsches Elektronensynchrotron DESY

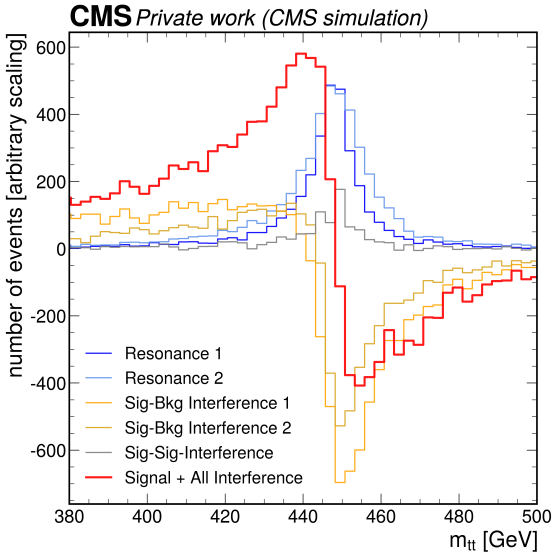
2 Universität Hamburg

3 Hochschule für angewandte Wissenschaften, HAW Hamburg

HELMHOLTZ

Negatively weighted Events

- This projected was motivated by a search for heavy Higgs bosons with CP violation
- There are a variety of interference scenarios in such a case



$$|S_{\mathrm{A/H}} + B|^2 = |S_{\mathrm{A/H}}|^2 + 2Re(S_{\mathrm{A/H}}B) + |B|^2$$

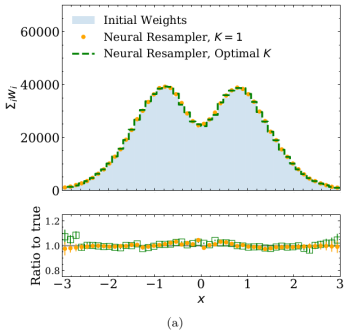
- Overall yield is positive!
- Signal can be partly negative (destructive interference)
- If we split into signal and background for a DNN training: (physically) truly negative parts of phase space!

Negatively weighted Events

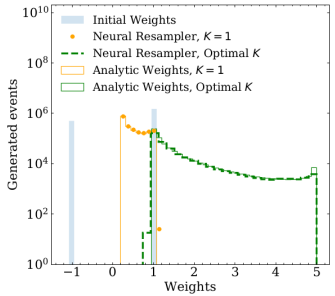
- There are negative events due to Monte-Carlo fixed order simulations – „Counter Terms“

$$d\sigma_{\mathrm{NLO}} = d\sigma_{\mathrm{LO}} + \underbrace{d\sigma_{\mathrm{V}}}_{=\infty} + \underbrace{\int d\Phi_1 d\sigma_{\mathrm{R}}}_{=\infty} = d\sigma_{\mathrm{LO}} + \underbrace{d\sigma_{\mathrm{V}} - \int d\Phi_1 C}_{\neq\infty} + \underbrace{\int d\Phi_1 \left(C + d\sigma_{\mathrm{R}}\right)}_{\neq\infty}$$

- This is a technical problem and there are solutions to this in literature – „Reweighting“
- For example ‚neural reweighting‘ (from Phys. Rev. D 102, 076004 (2020))



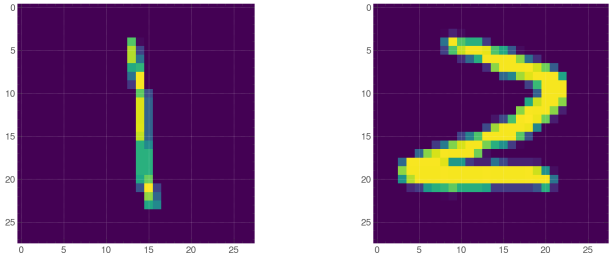
(a)



(b)

How do we consistently build a DNN based analysis for this?

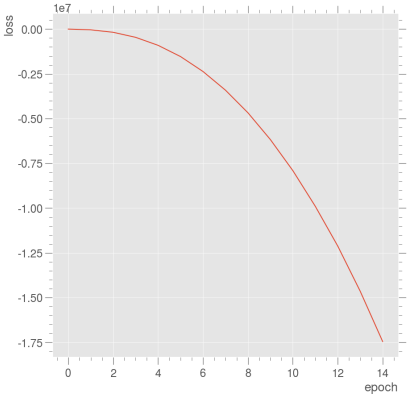
- DNNs have problems when trained on samples with ‘a number of’ negative weights
- We can see this in a simple experiment:
    - request a DNN to classify pictures of handwritten numbers
    - Weight all the samples from a class negatively
    - Evaluate on all of the other classes



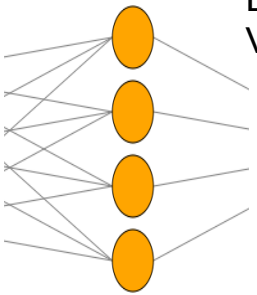
(all “weighted’ with w = 1), now artificially weight (for example the 2s) with w=-1

→ divergence!

Accuracy drops
from ~100% to 50%



We can fix the training!



Label Vector:

0
0
1
0

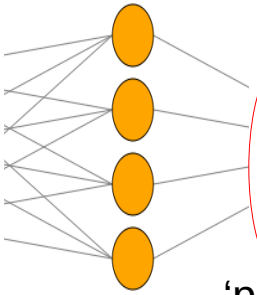
Weights:

$$W = -w$$

Loss:

$$\mathcal{L}_{\mathrm{CCE}} = \sum_{i=1}^{N} y_{\mathrm{target}} \cdot (-log(y_{\mathrm{output}}))$$

Equivalent to:



1
1
0
1

$$W = \frac{|w|}{(N_{\mathrm{out}} - 1)}$$

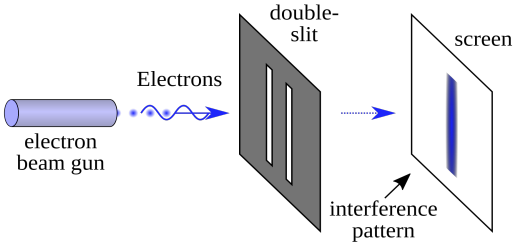
$$\mathcal{L}_{\mathrm{CCL}} = \sum_{i=1}^{N} \sum_{j=1}^{\mathrm{maxlabel}} y_{\mathrm{target}} \cdot (-log(y_{\mathrm{output}}))$$

‘partial label’

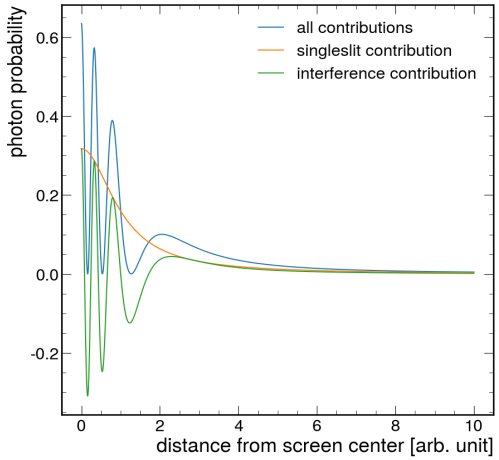
One possible partial-label loss, introduced by Feng et al (arxiv: 2007.08929)

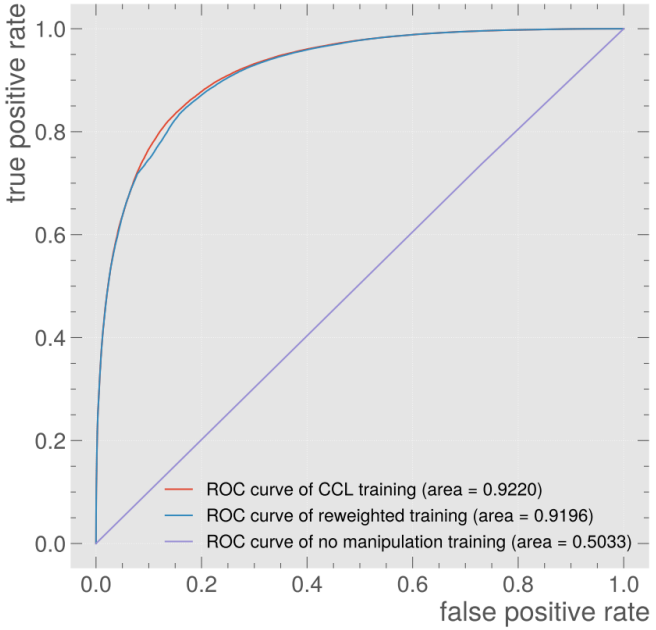
→ restores performance in MNIST toy model: Accuracy: 50% → 99%

We checked it in a scenario where we have full control

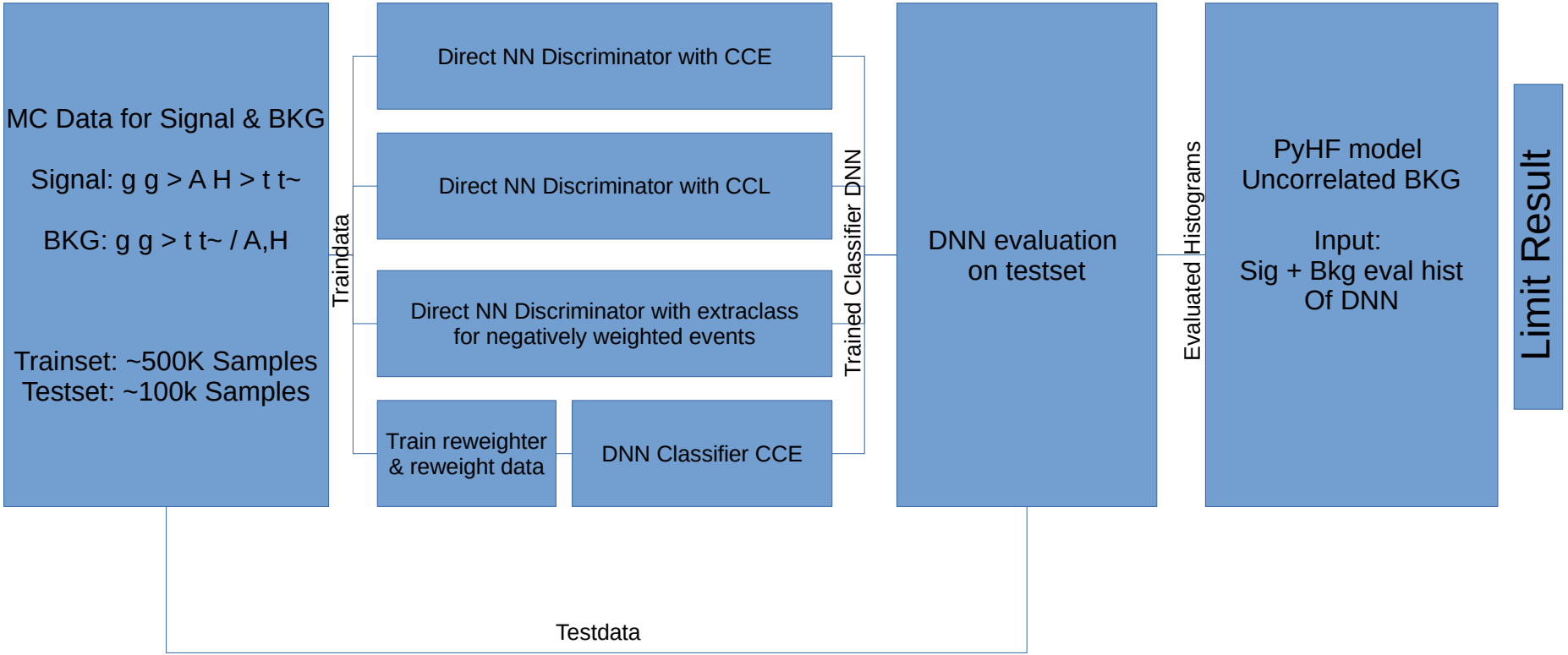


From By Original: NekoJaNekoJa Vector: Johannes Kalliauer - File:Double-slit.PNG, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=61496401





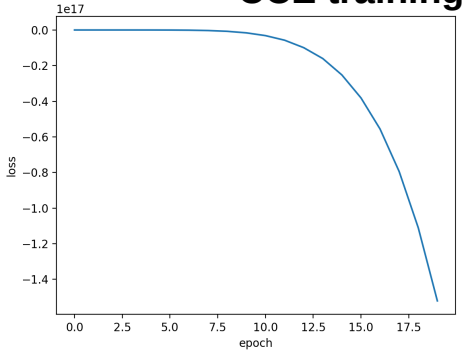
What about an analysis benchmark scenario?

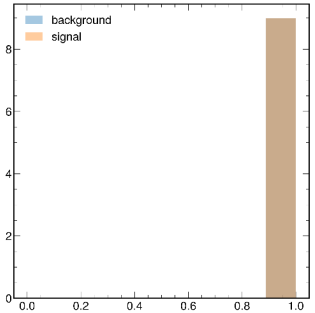


What about an analysis benchmark scenario?

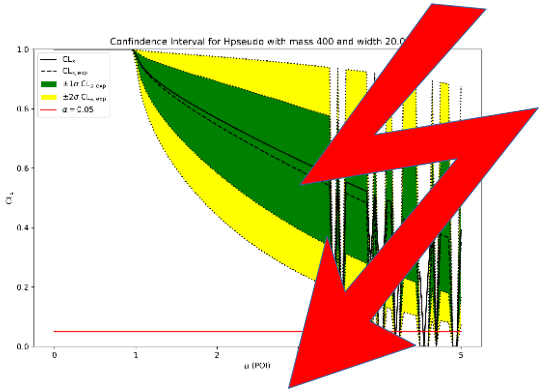
DASHH

CCE training

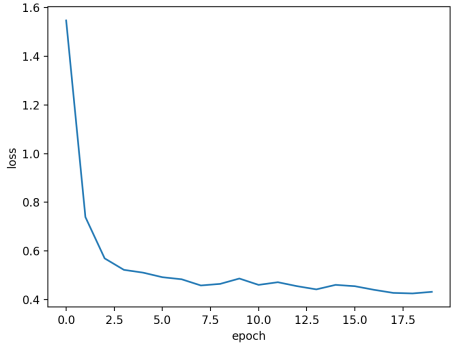


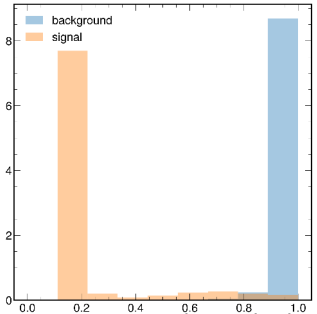


Node 0 activation

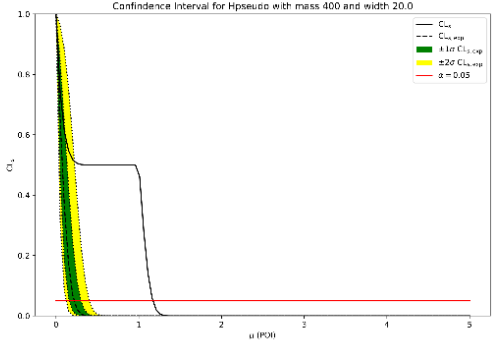


CCL training



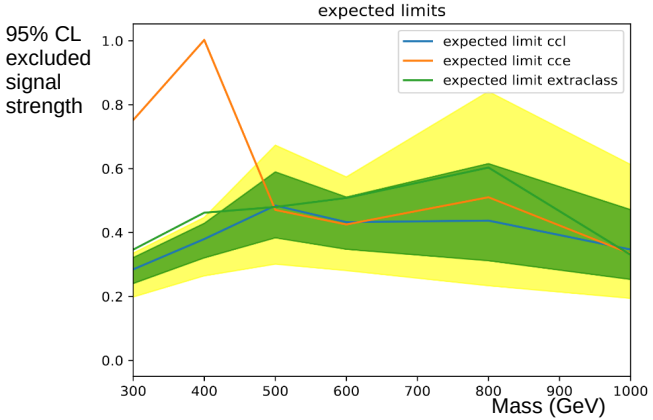


Node 0 activation



What about an analysis benchmark scenario?

- We also encountered the issue where the network evaluation produces some bins that have a negative net weight
- This is unphysical as far as DNN output bins hold physical value
- So far: worked with absolute value of bins for fit
- Currently implementing an algebra trick for yield to avoid negatively weighted events in the evaluation
- (very much WIP) results:



Conclusion

- We found a way to train DNNs in a stable manner when samples in the training distribution are negatively weighted

- It can be used in a full on DNN based LHC style analysis