Adversarial Classifier Network

Proof of concept to reduce systematic uncertainty in the ATLAS ttH(H->bb) classification

Paul Glaysher, José M. Clavijo, Judith Katzy (DESY)
Ilyas Fatkhullin (Moscow Inst. Phy. & Tech.)

15 Nov 2019
4th ATLAS Machine Learning Workshop

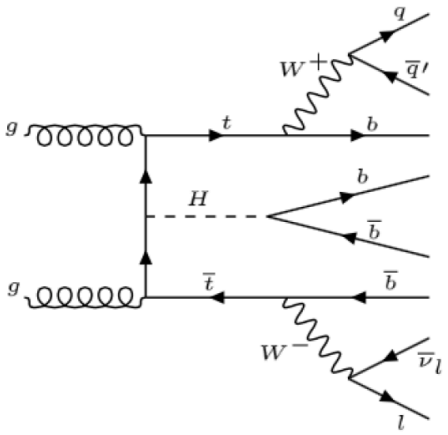
HELMHOLTZ ASSOCIATION

DESY

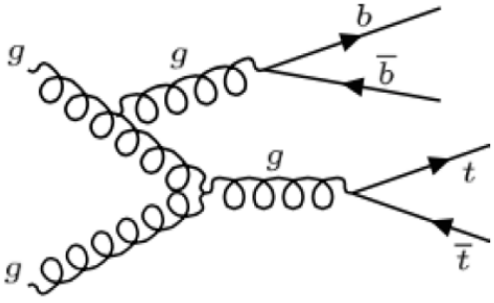
Intro

- For the example of a classification of ttH(H->bb) vs $t\bar{t}$+b-jets we will present a method of reducing a dominant systematic through adversarial domain adaptation
- The classification problem and event selection follow the ATLAS ttH(H->bb) 1712.08895 paper.
  - Using open-data MC with Delphes simulation to explore the algorithm

ttH(H->bb) signal

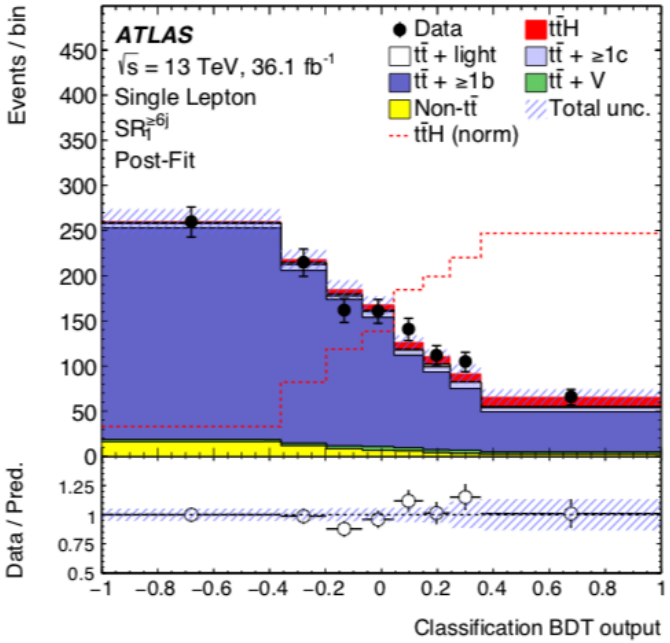


$t\bar{t}$+b-jets background



Motivation

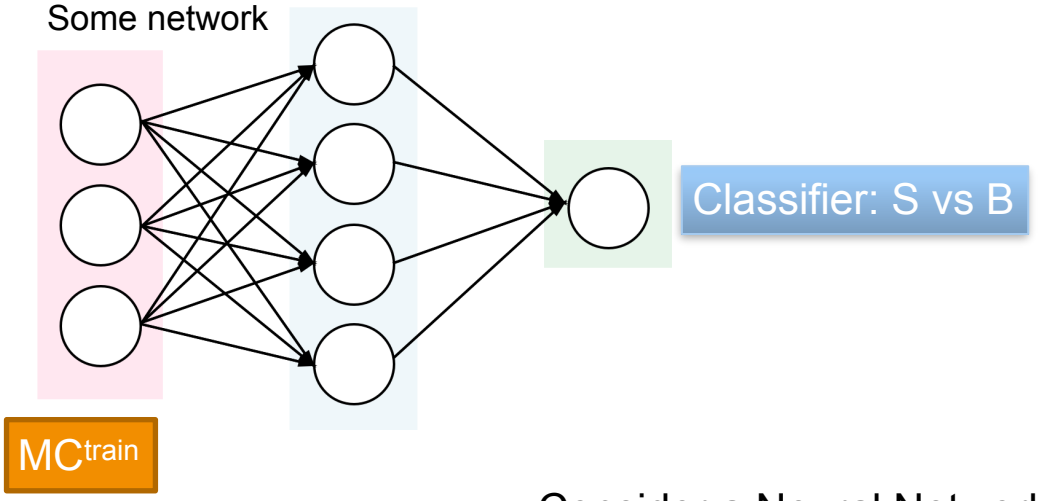
- A classification BDT to separate ttH(H->bb) from tt+b-jets background [1712.08895]
- Signal presence established from fit to classifier score
- BDT trained on nominal S vs B Monte Carlo
- BDT applied to
  - nominal MC for S+B expectation
  - Alternative MC model for systematic uncertainty



- Dominant uncertainty: different response of BDT to nominal vs **alternative background MC**
  - Uncertainty (shaded band) similar size as signal (red)
  - A type of over-training: bias towards training model
  - We want a BDT (or Neural Network) classifier score that looks the same when applied to inputs from different MC generators or systematic variations of the nominal training sample
  - Will show how a Adversarial Network can help

Idea

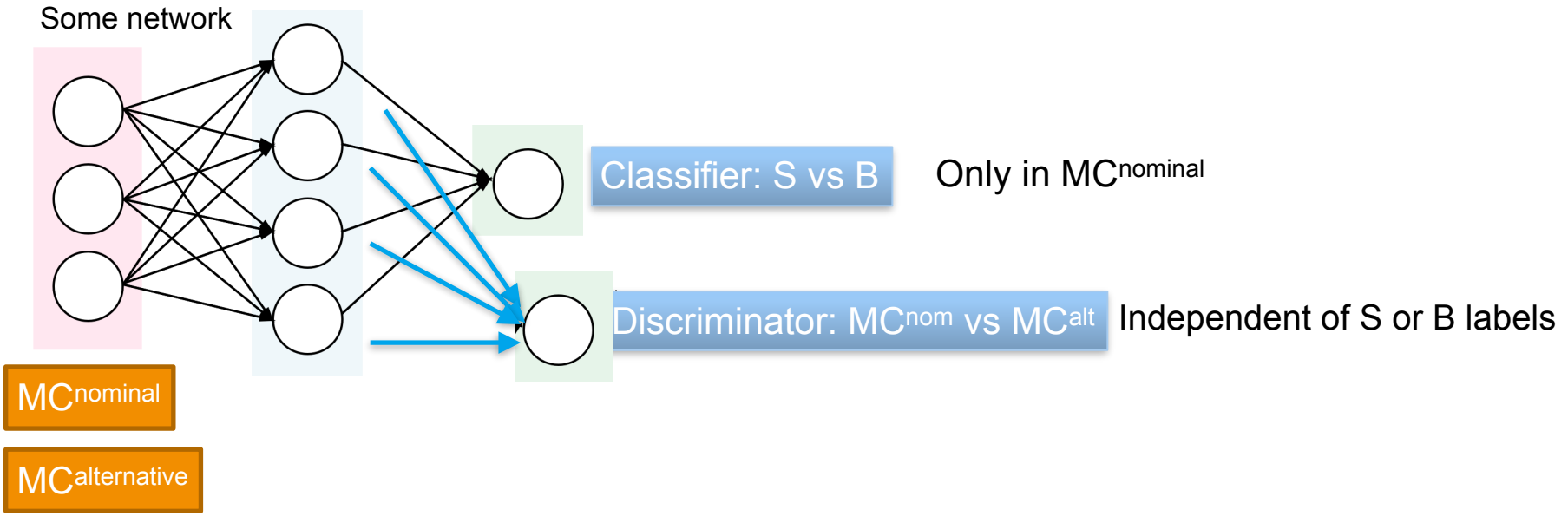
- Want a classifier that doesn't learn differences among MC samples



Consider a Neural Network as classifier (instead of BDT).
Optimising for for best classification power was not our focus here.

Idea

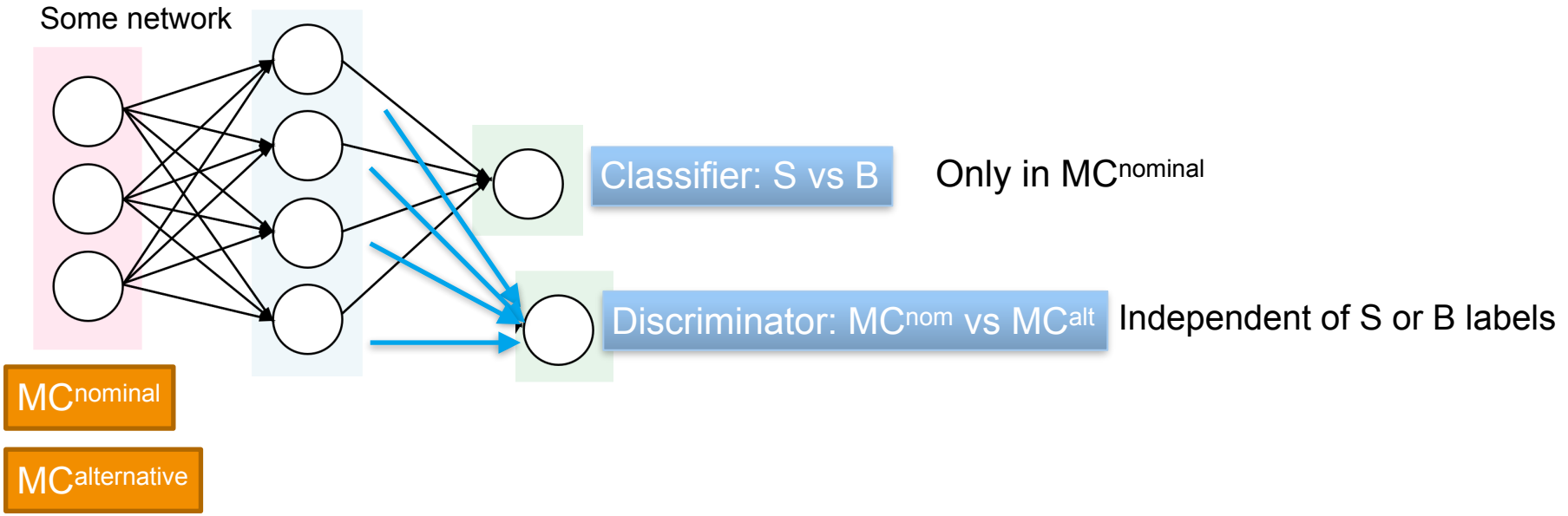
- Want a classifier that doesn't learn differences among MC samples



- Add a 2nd classifier, a 'discriminator', that learns the difference between MC generators
- Want best possible S vs B classification and worst possible nominal vs alternative classification
- Discriminator prevents Classifier from learning on phase space that differs among MC generators

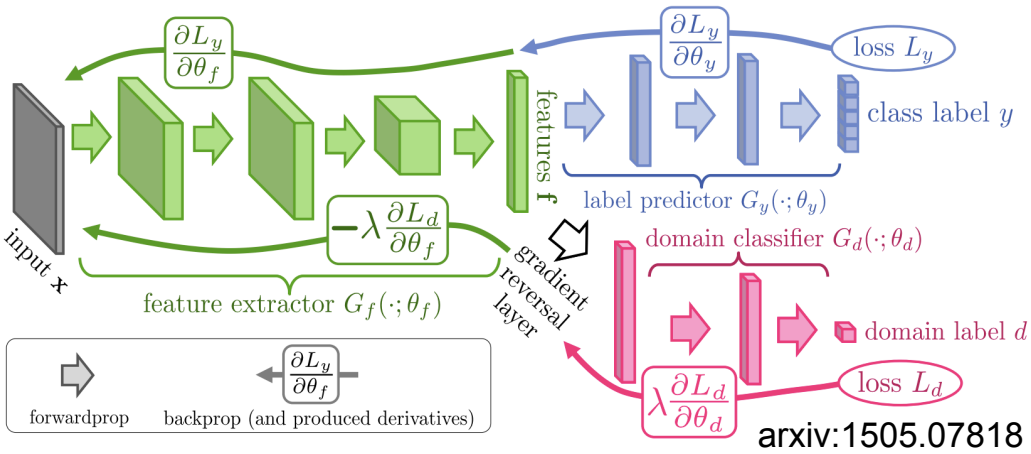
Idea

- This principle is called ‘adversarial domain adaptation’



- Last common layer ‘latent space’ must satisfy both conditions
- Classifier and Discriminator conditions can be in competition
  - Overall optimum will possibly reduce classification power, measured on nominal MC

Adversarial Domain Adaptation



S vs B label classifier

MC generator classifier

arxiv:1505.07818

- Here domain = different background MC samples
- Will keep signal the same
  - Classify: only in **source-domain** = $S^{nominal}$ vs $B^{nominal}$
  - Discriminate: **source-domain** = $S^{nominal}+B^{nominal}$ vs **target-domain** = $S^{nominal}+B^{alternative}$
- In practice discriminator is minimised (as the classifier), but the update to the loss function is multiplied by -1 with gradient reversal layer
- Relative discriminator strength controlled by λ term
  - Train network to move in reverse direction with respect to choices that make a difference between $B^{nominal}$ and $B^{alternative}$
- Minimise both conditions simultaneously in one global loss function

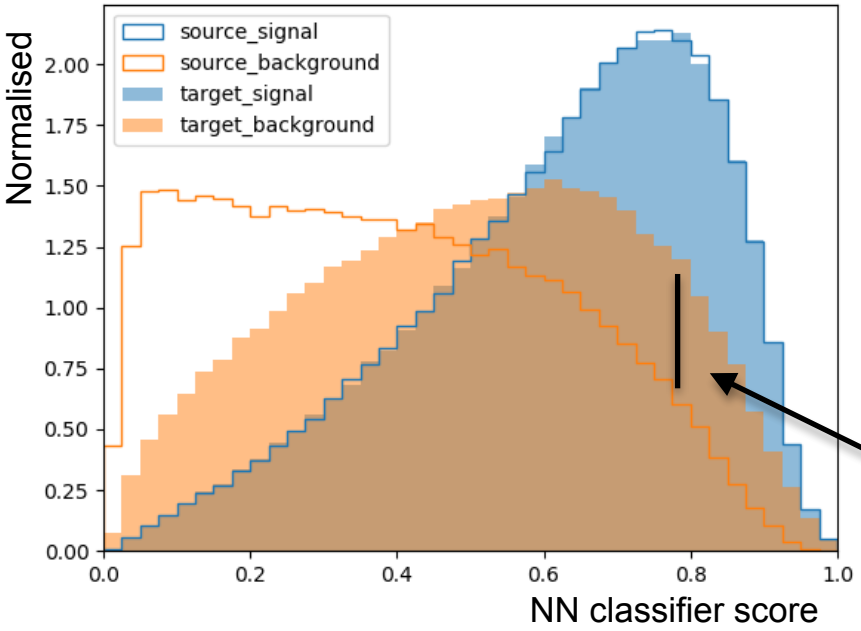
Test setup

- Use python, Keras, SciKit-Learn, convert Root files with *uproot*
- Run on GPU (optional complication)
- Use open data samples with Delphes fast sim
  - ttH: MadGraph/HW6 → $S^{nominal}$ (https://hepsim.jlab.org)
  - tt+jets: MadGraph/P6 → $B^{nominal}$ (https://hepsim.jlab.org)
  - tt+bb: PP8 → $B^{alternative}$ (https://www.physik.uzh.ch/data/PowhegBox+OpenLoops/ttbb/)
- Event Selection:
  - type of variables (40 variables) chosen similar to single lepton channel of 1712.08895, (some cuts loosened to gain stats)
  - 1 lepton with pt> 20 GeV and ≥5 jets with pt>25 GeV
  - ≥3 b-jets, with 70% WP, b-efficiency, light/c-rejection is parameterised according to JHEP08(2018)089
  - Smallest sample tt+jets, ~ 300k events

Classification response without discriminator

- Feature extractor network, [20, 16, 13, 10] neurons + softmax classification layer.
- 16k events batch size. ELU activation function.
- Situation without adversarial discriminator

here: 2000 epochs



Train on source, see response to
- Source: nominal Bkg sample (lines)
- Target: alternative Bkg sample (area)

- Signal agrees by construction.
- Different Background generators show large discrepancy.

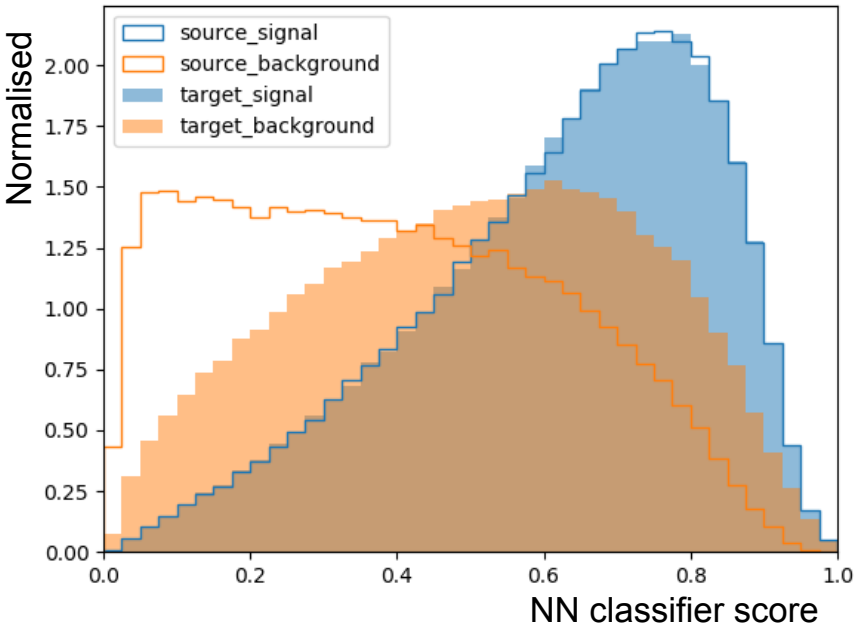
This is the same type of background generator uncertainty in the reference analysis, it is a type of training bias. Large in comparison to signal.

Expected S/B≈5%, so uncertainty on background estimate greatly diminishes signal sensitivity.

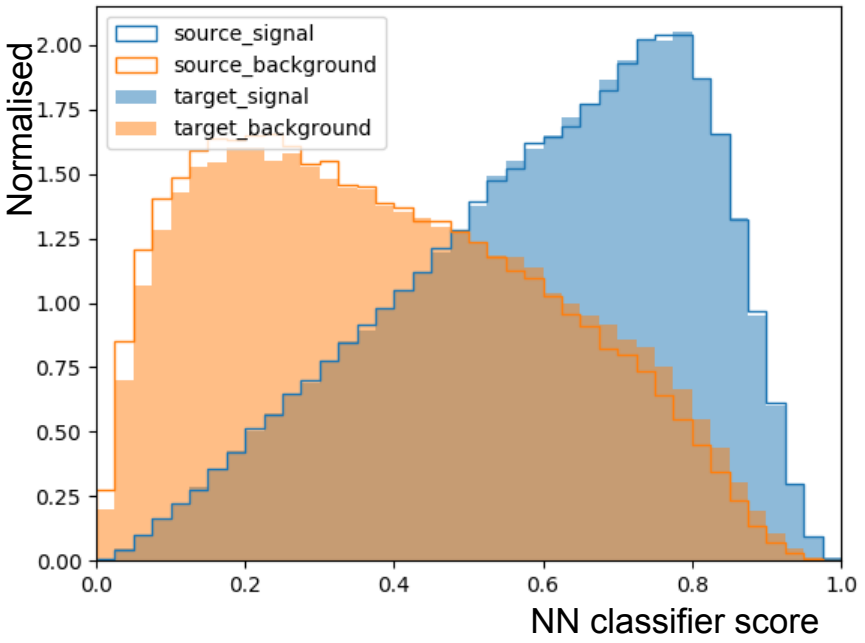
Plots from José M. Clavijo

Classification response with discriminator

No discriminator



With discriminator $\lambda = 50$



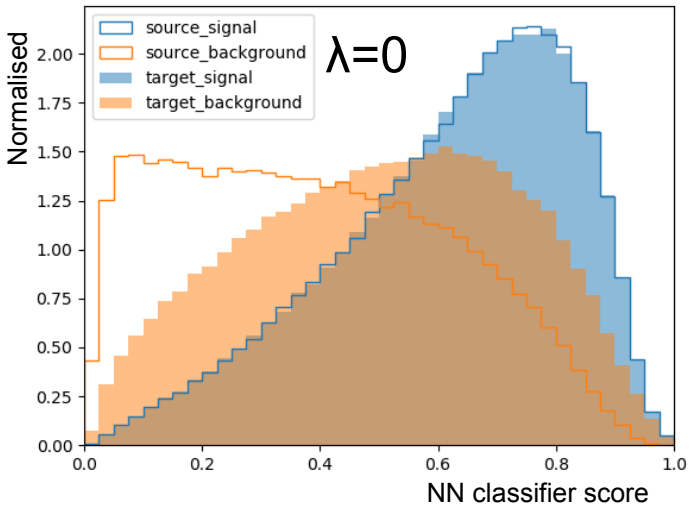
Difference in orange curves due to background generator choice

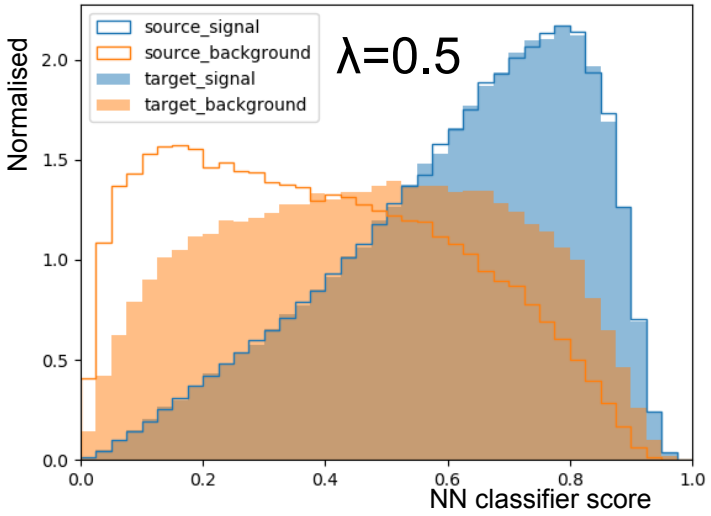
Discriminator network:
[20,35,50] neurons
+ softmax output layer

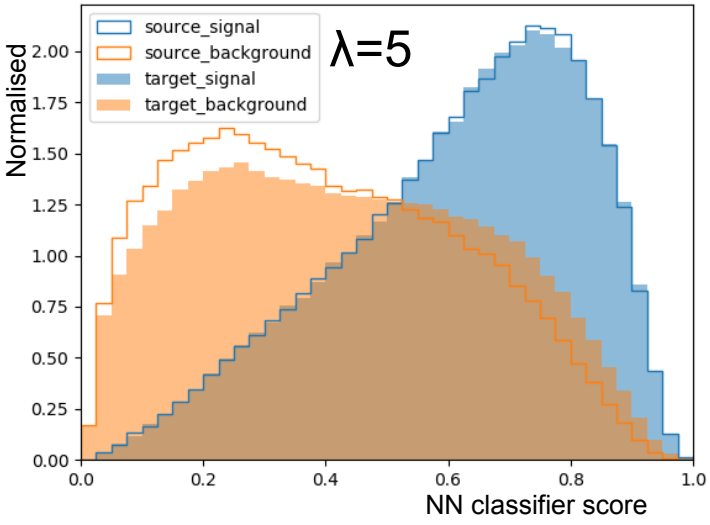
- Same setup, but with adversarial discriminator greatly reduces uncertainty from choice of background generator!
- Particularly beneficial in signal-rich region
  - Go from ~ 20-50% shape difference to ~5-10% shape difference
- Shape of signal response also slightly modified

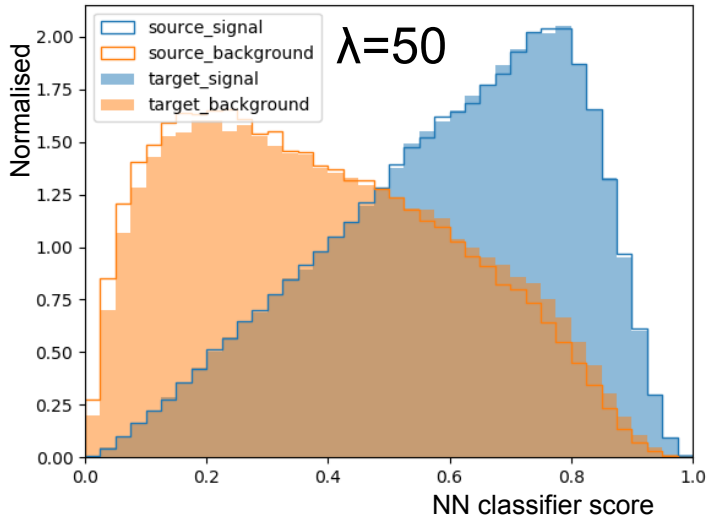
Scan over λ

- λ: free parameter to scale importance of discriminator relative to the classifier in the minimisation



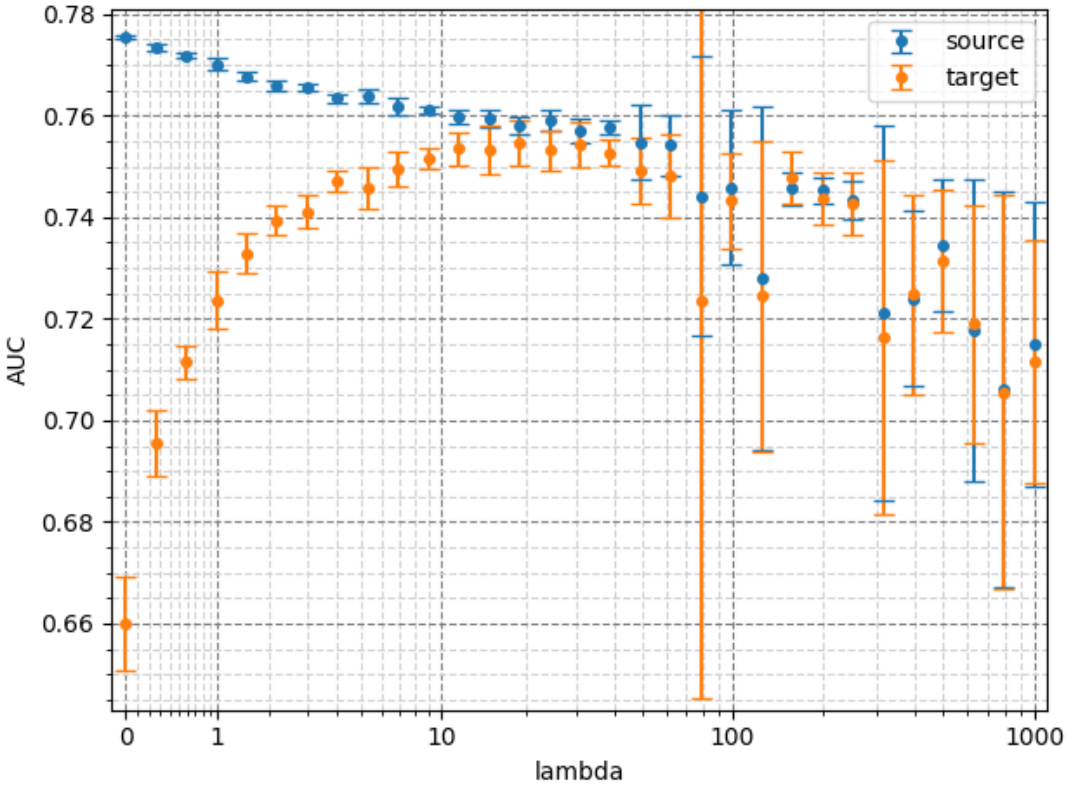






Scan over λ

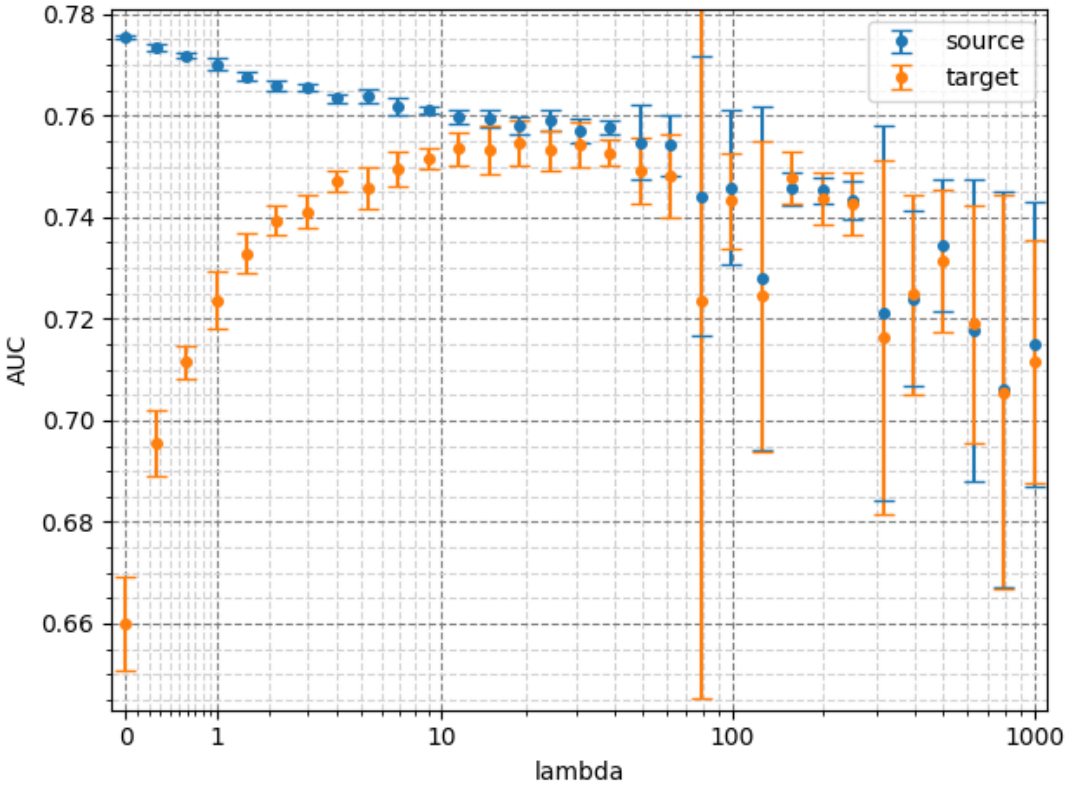
- Area under ROC curve on source and target
- averaged over 10 runs (random weight initialisation and train/test split)
- Reduction in classification power of source sample buys improved classification power in target and thus better agreement among the domains



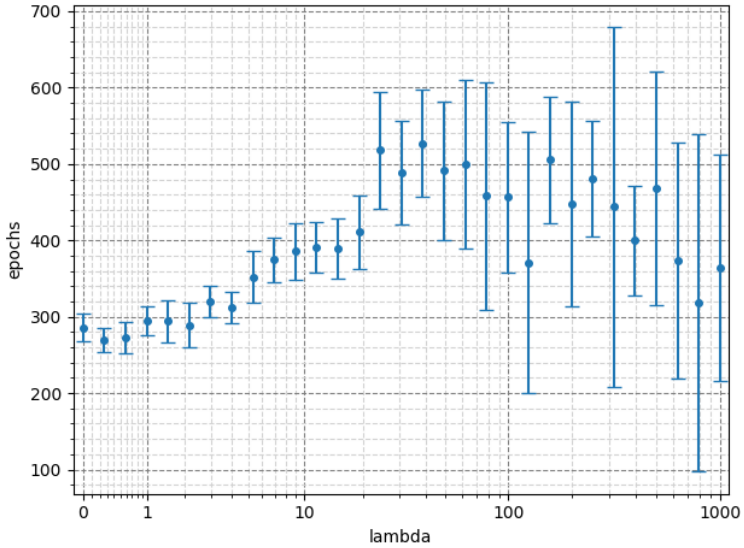
- λ ≈ 20 - 50 best choice
- Higher λ brings no improvement within statistical margin and only degrades training convergence
  (see larger variance in training over the 10 test runs)

Scan over λ

- Area under ROC curve on source and target
- averaged over 10 runs (random weight initialisation and train/test split)
- Reduction in classification power of source sample buys improved classification power in target and thus better agreement among the domains



- Stopping condition: when change in loss function is <1% from one 50 epoch window to the next
- Can see how higher λ leads to more epochs



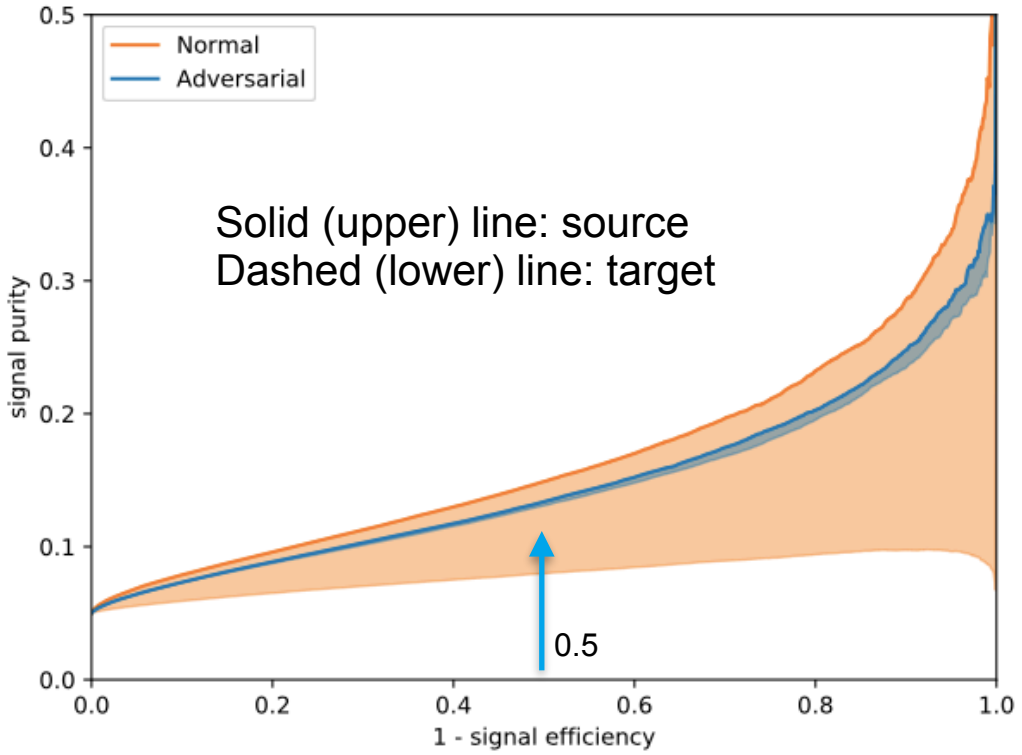
Performance

- The performance of the classifier and its systematic variation needs to be properly evaluated in an Asimov profile likelihood fit
- Here as a proxy for the performance, we quote the signal purity (=S/ S+B) for a fixed signal efficiency of 0.5
- The central value is taken from the nominal (source) sample
- the error due to background modelling uncertainty, like in the search, is taken as the difference between nominal and alternative (target)

Without adversarial:
$\text{Purity}_{\text{sig}} = 0.15 \pm 0.07$

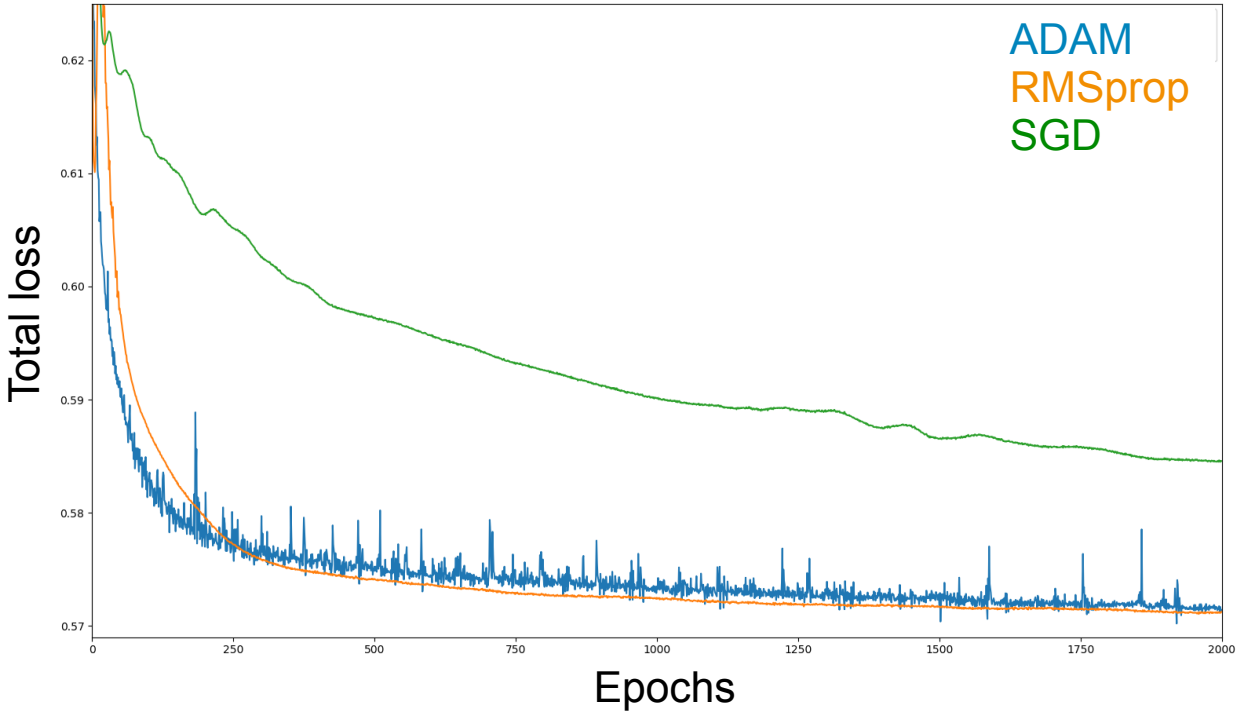
With adversarial:
$\text{Purity}_{\text{sig}} = 0.13 \pm 0.01$

=> better sensitivity expected with adversarial training of the ttH(bb) classifier



On optimiser choice

- Training stability and convergence an issue with such a network
  - Improved stability when switching from ADAM to RMSprop optimiser



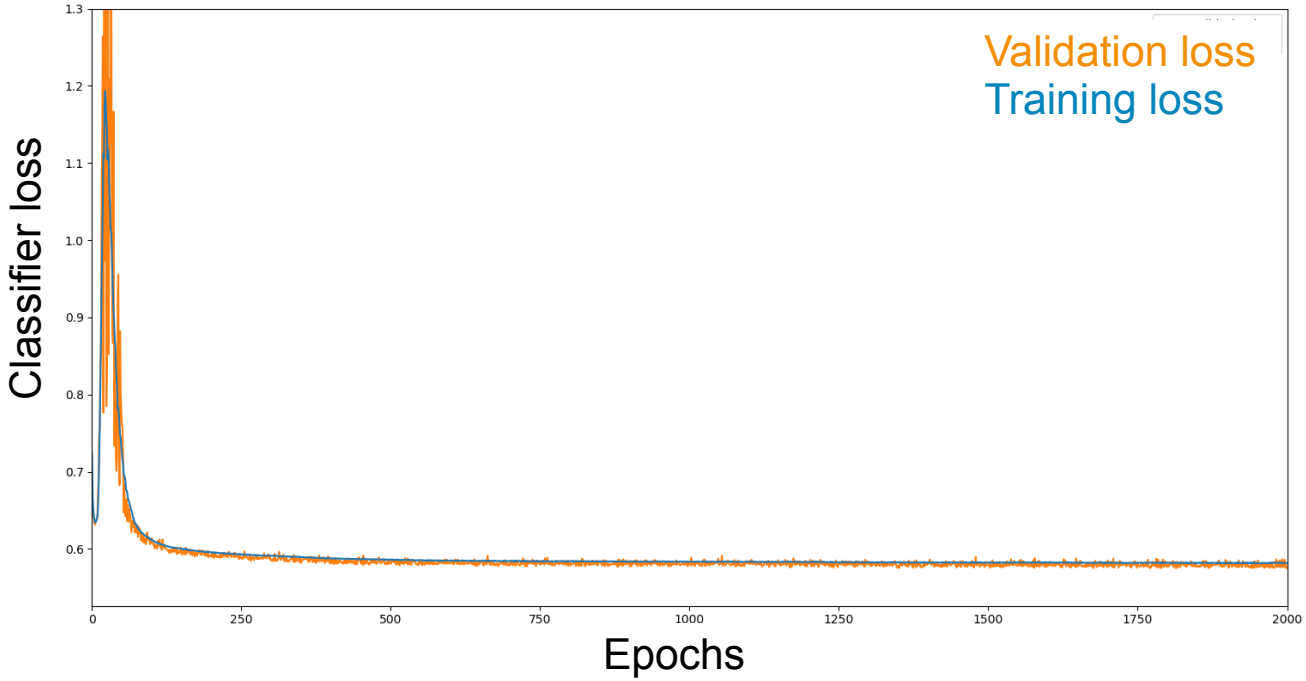
The ‘momentum’ term in ADAM, favours minimisation in same direction of previous epoch. This causes fluctuation in our case where we have minimise the sum of two loss functions.

RMSprop (ADAM without momentum) does much better. -> what we use here

Stochastic Grad. Desc. For comparison, is very slow in reaching optimum.

On overtraining

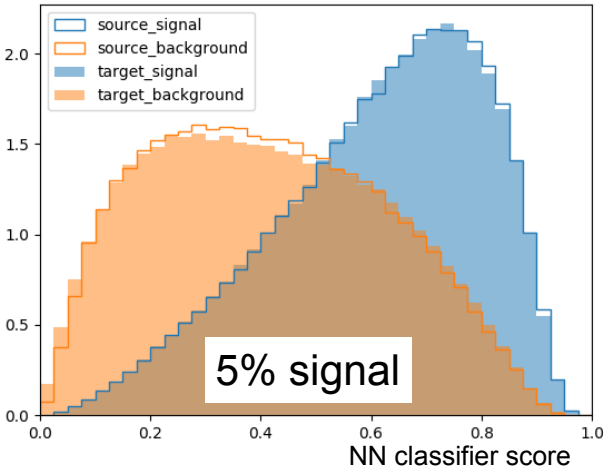
- One nice side-effect of the discriminator is that it prevents the classifier overtraining on the source sample
- plot: Loss for training and validation set of source sample over large number of epoch (normally use ~ 300 epoch)
  - Robust against overtraining

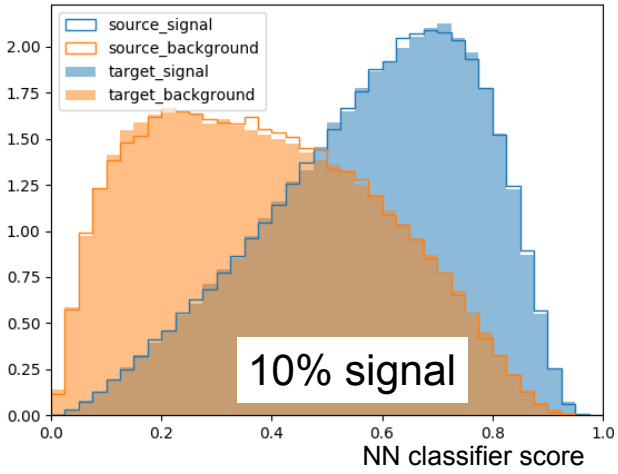


Sensitivity to S/B ratio

The fraction of signal to background is only used in the discrimination of S+B.

- Same signal fraction for both source and target
  - No difference, actual fraction in discriminator not important



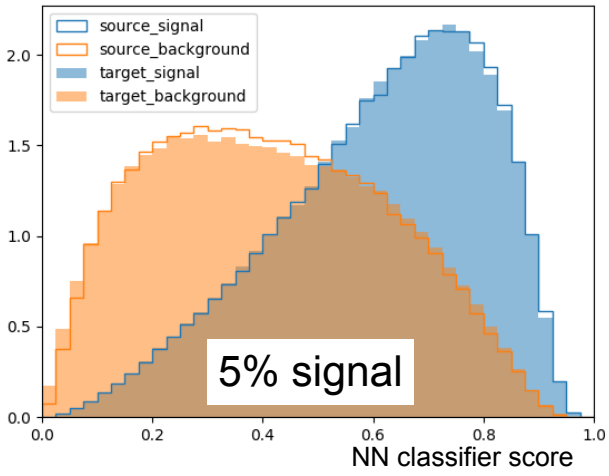


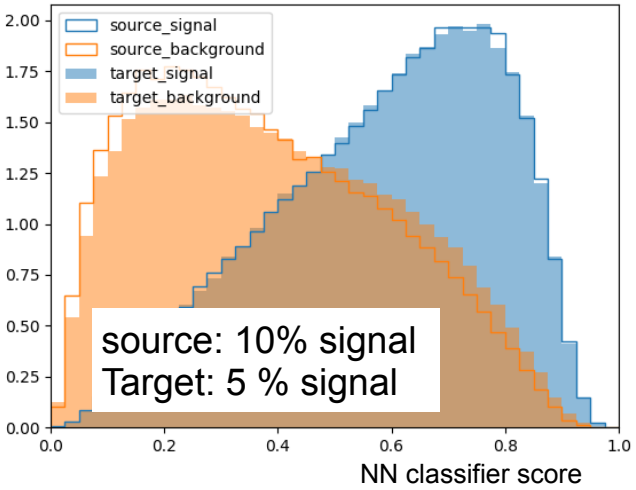
Slight difference due to random initialisation. AUC over 10 runs equivalent to 76% within error.

Sensitivity to S/B ratio

The fraction of signal to background is only used in the discrimination of S+B.

- Mismatch of signal fraction in source and target
  - impact on classifier response proportional to mismatch





- Higher signal fraction in source will cause higher mis-classification of target background as more signal-like (and vice versa)
  - Important to understand differences in S/B between source and target domains

Some thoughts

- Although not demonstrated, can in principle extend this approach to discriminate multiple variations during training
- It remains to be explored, how reducing the sensitivity of systematics by adding them to the classifier training would alter the strategy of the subsequent profile likelihood fit.
- The discriminator does not use class labels: could use technique to train classifier on labelled MC sample and correct against unlabelled real data in the discriminator
  - Sensitivity to S/B mismatch becomes important
- Training convergence is non-trivial for such a network. We have achieved a relatively stable configuration, which fails only in freak cases.
- Recommend to try:
  - low λ, many epochs, RMSprop, fine-tune network architecture:
  - small number of classifier layers wrt. feature extractor
  - More discriminator than classifier layers

Conclusion

- The ATLAS ttH(bb) analysis is limited by the background modelling uncertainties, that result in a bias of the classifier towards the Monte Carlo generator used for training.
- We demonstrate that **adversarial domain adaptation** can produce a more generator independent classifier, while preserving most of the classification power
- The impact of the uncertainty due to the choice of background model on expected signal purity (a proxy measure of the analysis sensitivity) can be improved from ~44% to ~3.6%
  - Conditional on the choice of background samples used for this study
- The benefits of this approach probably generally apply to other search channels as well

Backup

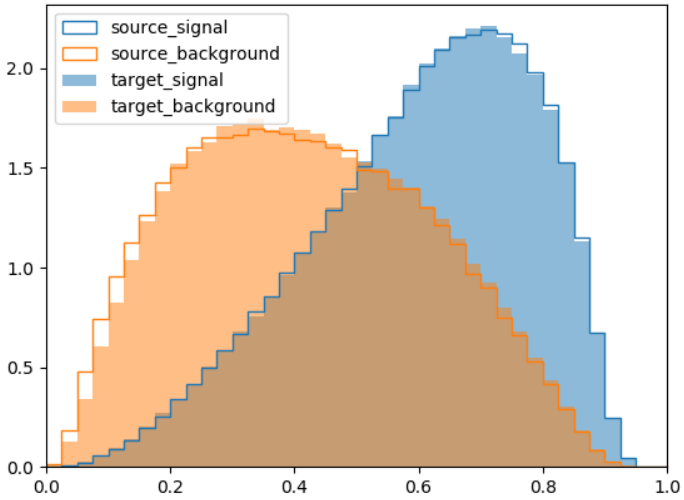
Training variables

- Not the same as in the paper, 40 variables in total

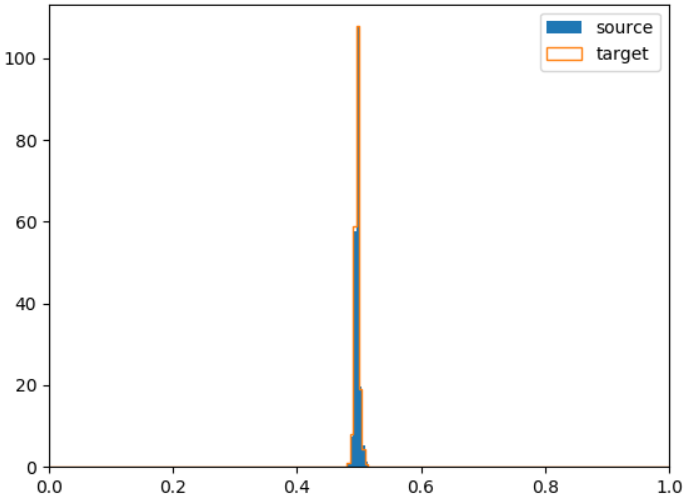
dRbb_avg	average dR of all b-jet pairs
dRbb_MaxPt	dR of the b-jet pair with the highest sum of p_T
dRbb_MaxM	dR of the b-jet pair with the highest invariant mass
dRlb1-dRlb3	dR of the charged lepton and the b-jet with the 1st-3rd largest p_T
dRlbb_MindR	dR of the charged lepton and total b-jet pair system which has the smallest dR
dRlj_MindR	minimum dR between the charged lepton and any jet
Mbb_MaxM	maximum invariant mass of any b-jet pair
Mbb_MindR	invariant mass of b-jet pair which has the smallest dR
Mbj_MaxPt	invariant mass of two jets with the largest p_T sum, where exactly one of the jets is a b-jet
Mjjj_MaxPt	invariant mass of any three jets with the largest p_T sum
pT_lep	transverse momentum of the charged lepton
HT_jets	sum of transverse momentum of all jets
HT_all	sum of transverse momentum of all jets and the charged lepton
nJets_Pt40	number of jets with $p_T \geq 40$ GeV
nbTag	number of b-jets
nHiggsbb30	number of b-jet pairs with an invariant mass within 30 GeV of the Higgs boson mass of 125 GeV
MET	missing transverse energy
dEtajj_MaxdEta	largest difference in longitudinal angle η of any two jets
Centrality_all	ratio of momentum sum over the energy sum of all objects
Hi_all, H2_jets	1st-5th Fox Wolfram transverse moment [9] of all objects

Example 1

Classifier out

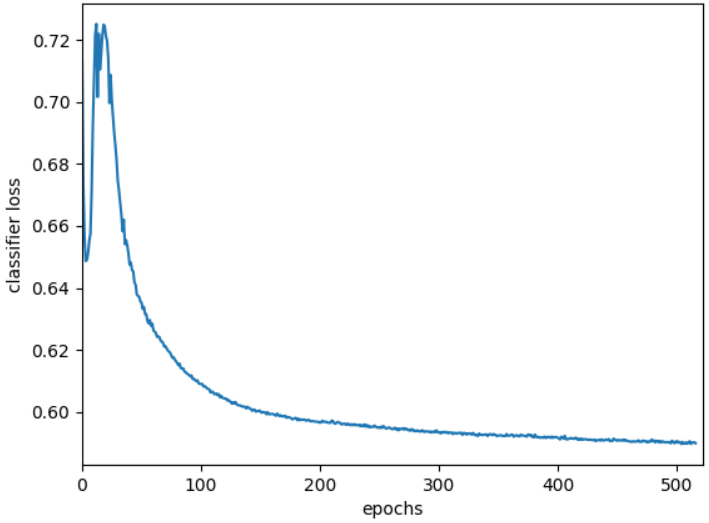


Discriminator out



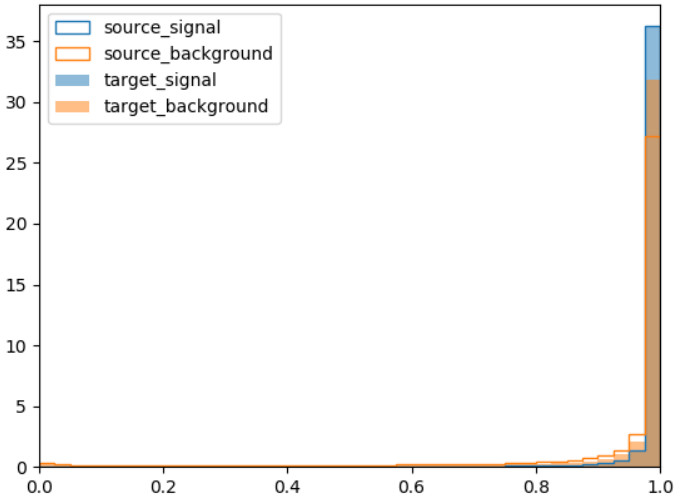
- Lambda: 78.4
- Epochs: 516
- Source AUC: 0.7536
- Target AUC: 0.7464

Classifier loss on source

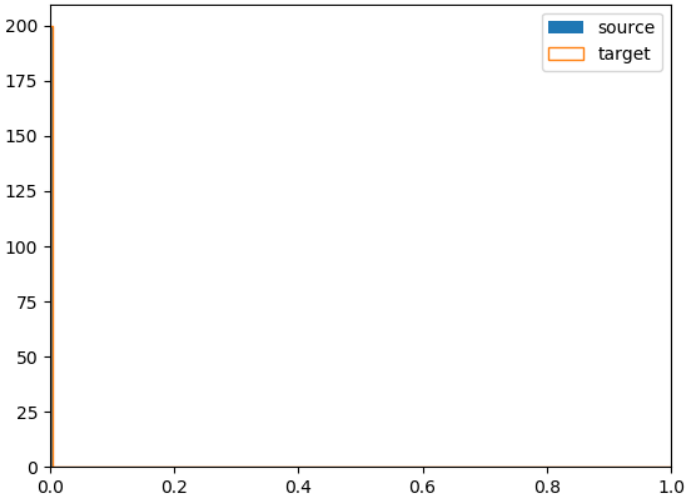


Failed example 2

Classifier out



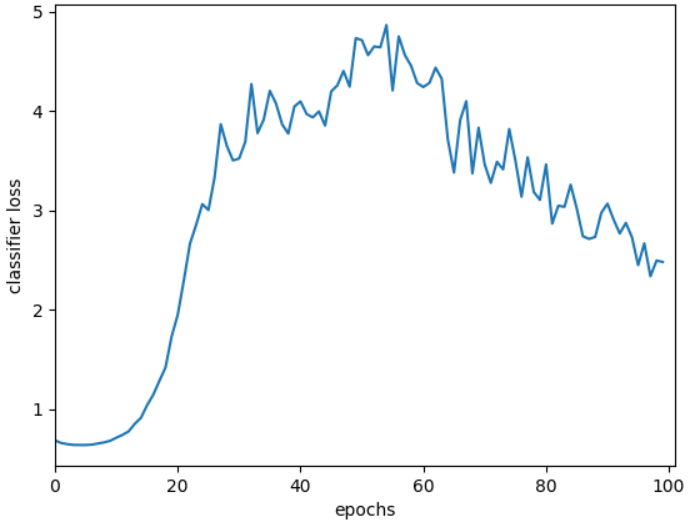
Discriminator out



- Lambda: 78.4
- Epochs: 99
- Source AUC: 0.6668
- Target AUC: 0.5025

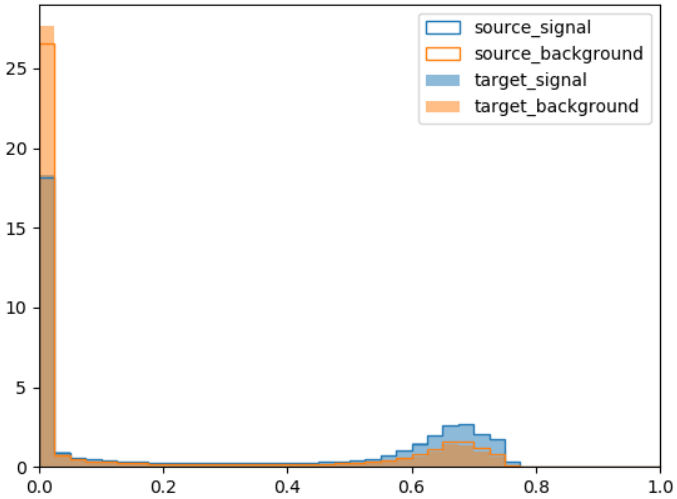
- Stuck in edge case minimum, which satisfies the discriminator
- Might eventually converge with more epochs

Classifier loss on source

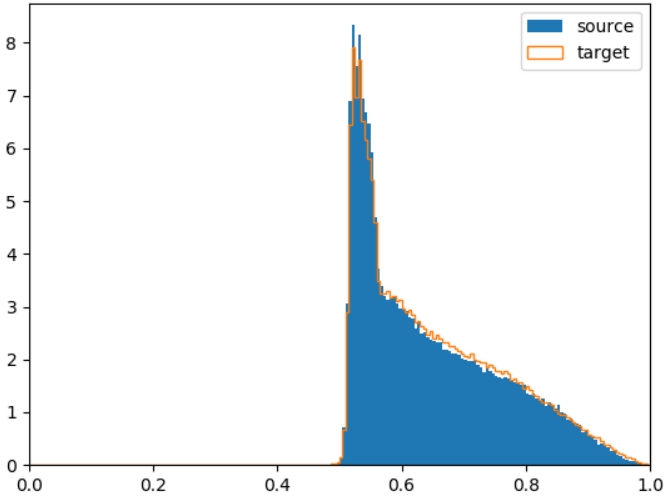


Failed example 3

Classifier out



Discriminator out



- Lambda: 630
- Epochs: 113
- Source AUC: 0.6432
- Target AUC: 0.6585

- Loss minimisation of two competing networks sometimes fails

Classifier loss on source