Machine Learning Uncertainties with Adversarial Neural Networks

Peter Galler

University of Glasgow, School of Physics and Astronomy

in collaboration with

Christoph Englert, Philip Harris, Michael Spannowsky

3rd IML Machine Learning Workshop

CERN, 16.04.2019

University of Glasgow

Science & Technology Facilities Council

UK Research and Innovation

Introduction

- Machine learning (ML): powerful tool to reveal and exploit correlations in a high-dimensional parameter space
- HEP: ML identifies regions in phase space which are sensitive to new physics (NP) signals
- these regions can be highly exclusive and perturbatively poorly understood $\longrightarrow$ systematic uncertainties
- unsupervised data-driven methods might not be available in these regions in particular for NP models with new interactions or rare processes
  $\longrightarrow$ ML trained by Monte Carlo data subject to theoretical uncertainties
- due to ability to find patterns in (MC) data ML correlates uncertainties with event classification: virtue of ML $\longrightarrow$ nuisance

Decorrelation of Nuisance Parameters

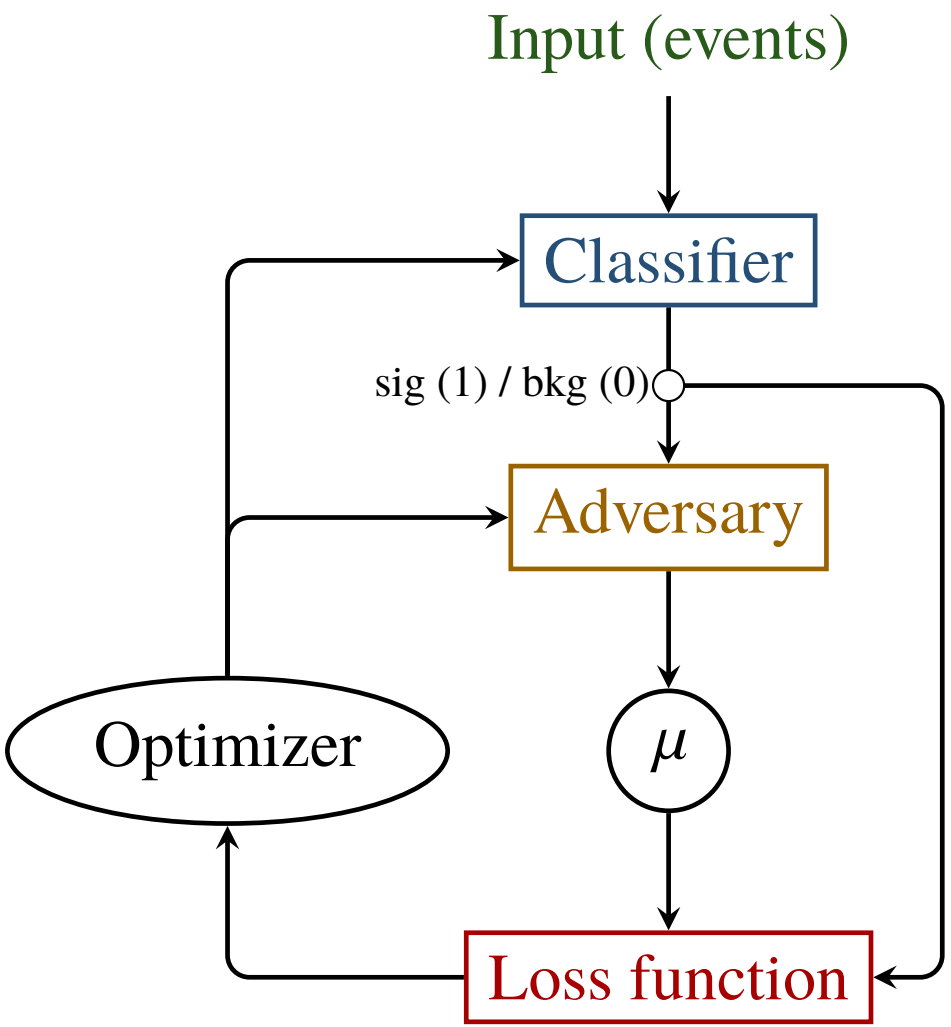
Solution: use a pivotal quantity [DeGroot, Schervish, '75] as a classifier

Pivot

quantity whose distribution is invariant with nuisance parameters

- [1611.01046, Louppe, Kagan, Cranmer]: training method for pivotal classifiers using adversarial networks
- showed that it works for systematic uncertainties which affect classification on an event-by-event basis (e.g. event reconstruction uncertainties)
- Here: use pivotal classifiers in the context of *theoretical* scale uncertainties affecting the event sample as a whole

Classifier vs. Adversary



Nuisance parameter
$\mu = \mu_0, 2\mu_0, \frac{1}{2}\mu_0$

Classification in sig. vs. bkg.

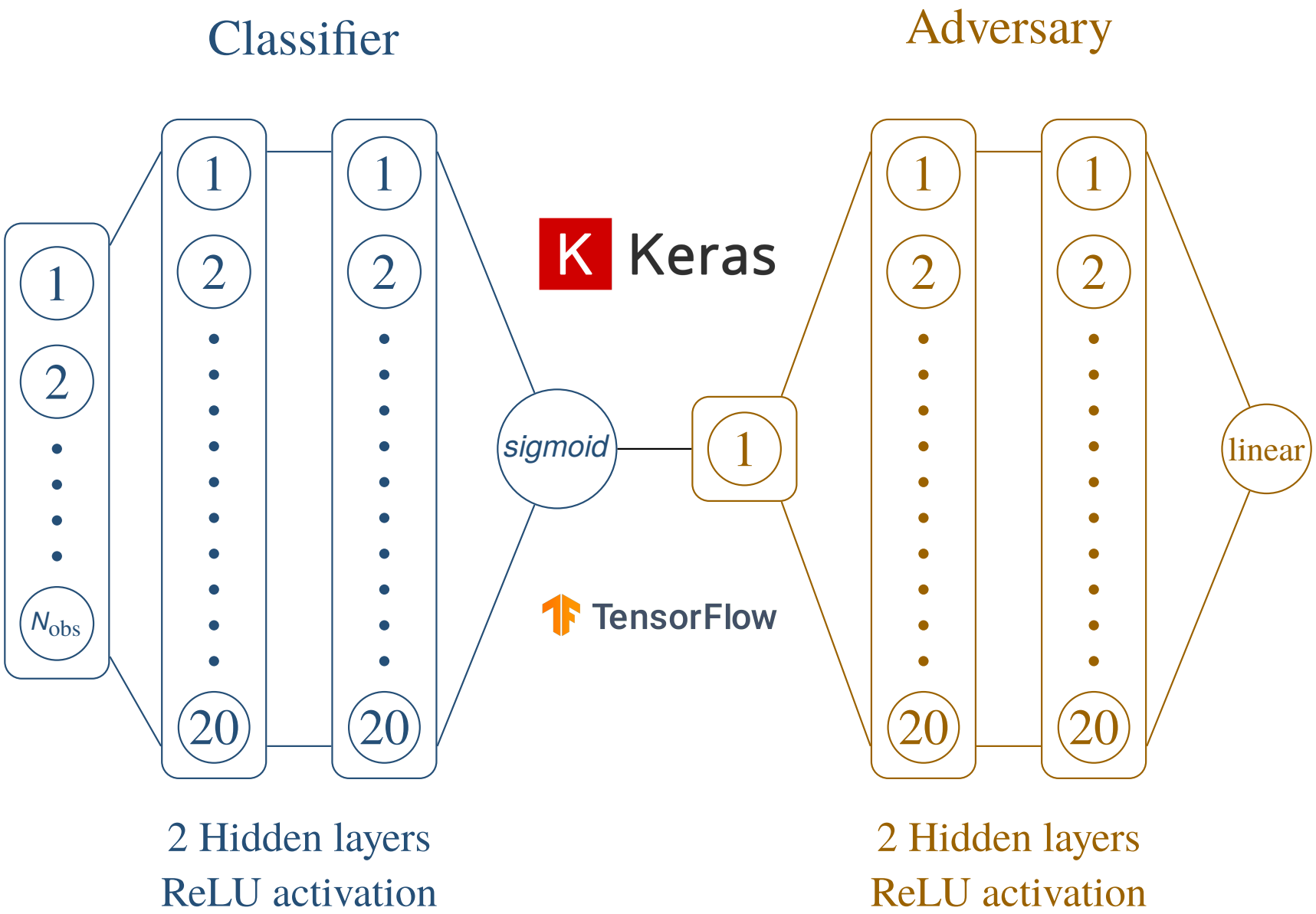
Regression on μ based only from classifier output

Combined loss function

$$L = L_{\text{Class}} - \kappa L_{\text{Adv}}$$

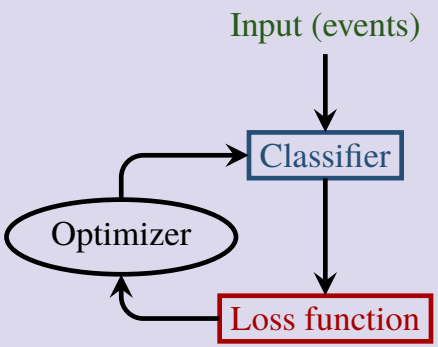
L_{Class}: binary cross-entropy; L_{Adv}: mean squared error

Implementation: Architecture

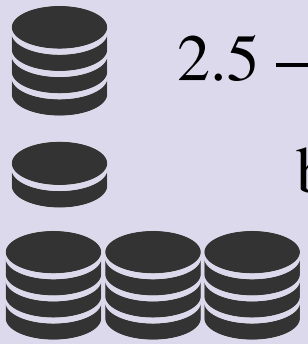


Implementation: Training

Initialization



Training parameters

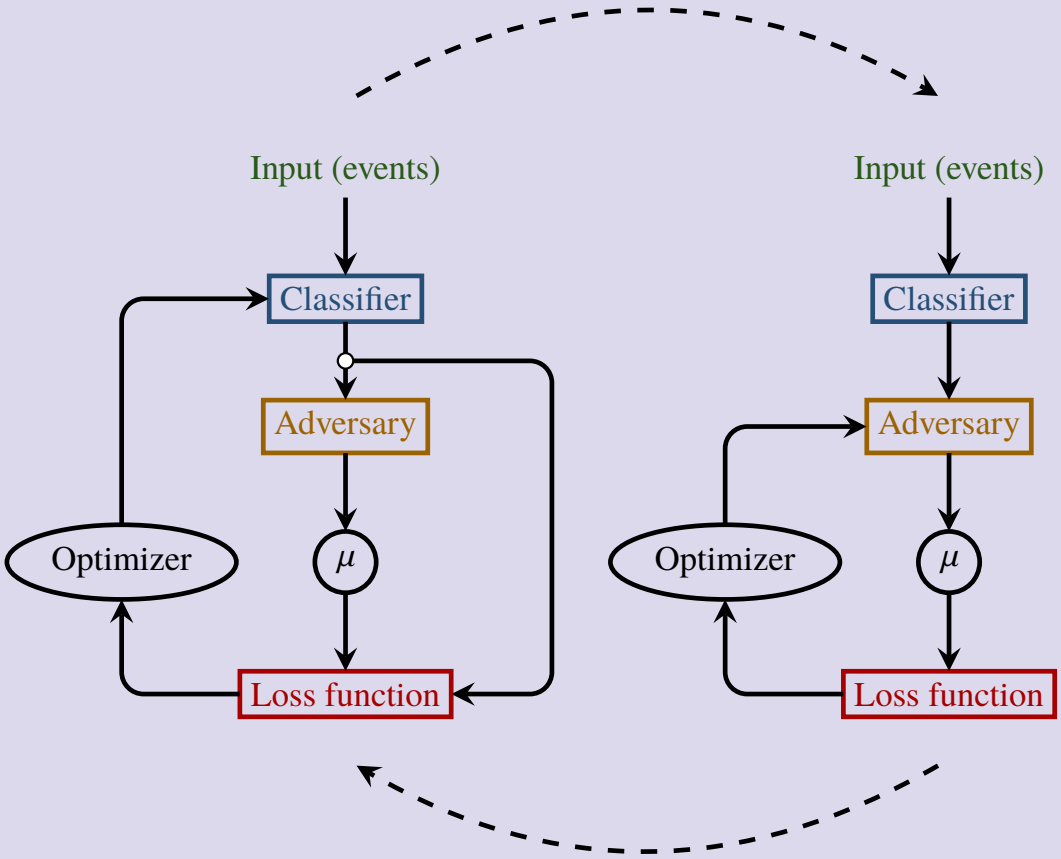


$2.5 - 4 \times 10^5$ events

batch size: 500

epochs: 500

Batch-wise alternating training



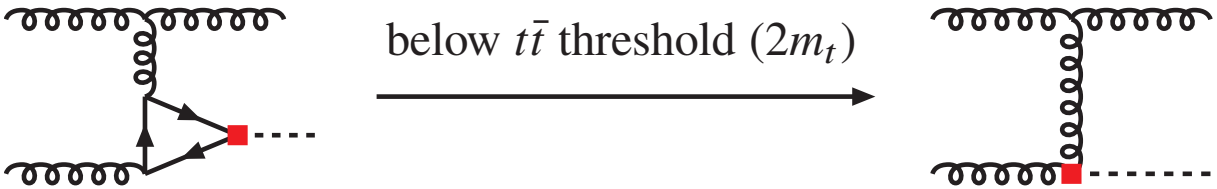
Learning Rate

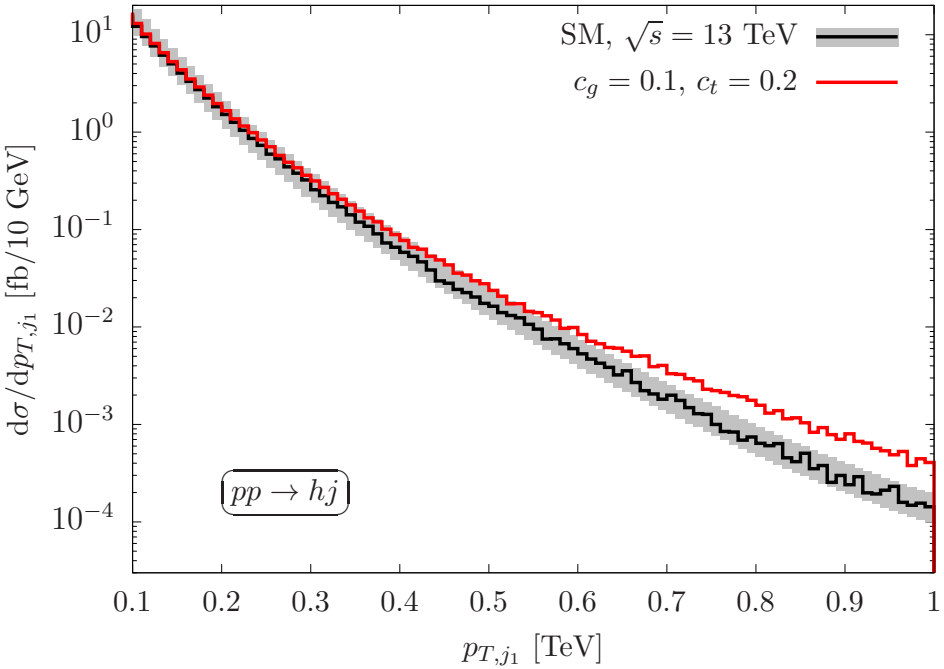
$$\lambda = 10^{-4}$$

Physics Case: EFT in Higgs + Jets Production

study modified top Yukawa and effective gluon-Higgs couplings:

$$\mathcal{L} \;=\; c_t\,\bar{t}t\,h \;+\; c_g\frac{g_s^2}{16\pi^2 v}G^{a\,\mu\nu}G^a_{\mu\nu}\,h$$





$$\mathcal{L}_{\rm low} = \left(c_g - \frac{\sqrt{2}}{3}\frac{c_t}{y_t}\right)\frac{g_s^2}{16\pi^2 v}G^{a\,\mu\nu}G^a_{\mu\nu}\,h$$

At low energies the modified top Yukawa coupling can compensate contributions from effective gluon-Higgs couplings if $c_g \sim \sqrt{2}/3\,c_t$

Example for a Pivotal Classifier

Process: $pp \to hjj$

nuisance parameter: $\mu = \mu_R = \mu_F = \{\mu_0/2, \mu_0, 2\mu_0\}$ (scale uncertainty)

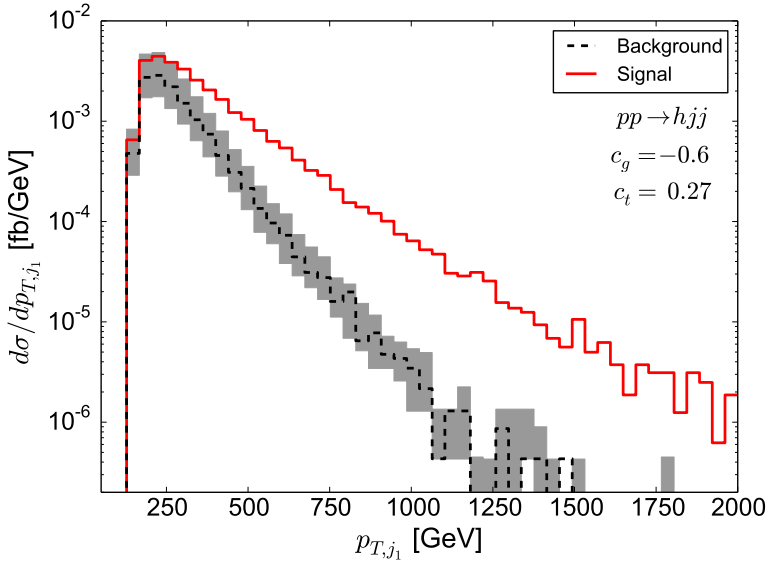
$\mu_0 = \sqrt{p_{T,j_1} p_{T,j_2}}$

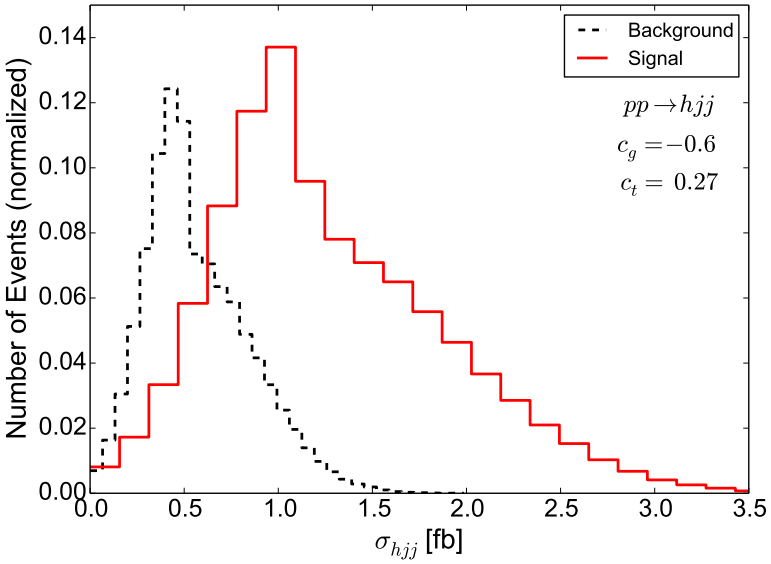
NN classifier blind to flat uncertainties

take flat uncertainties into account by including total cross section σ_{tot} as a feature

Problem: σ_{tot} is not an event-by-event observable

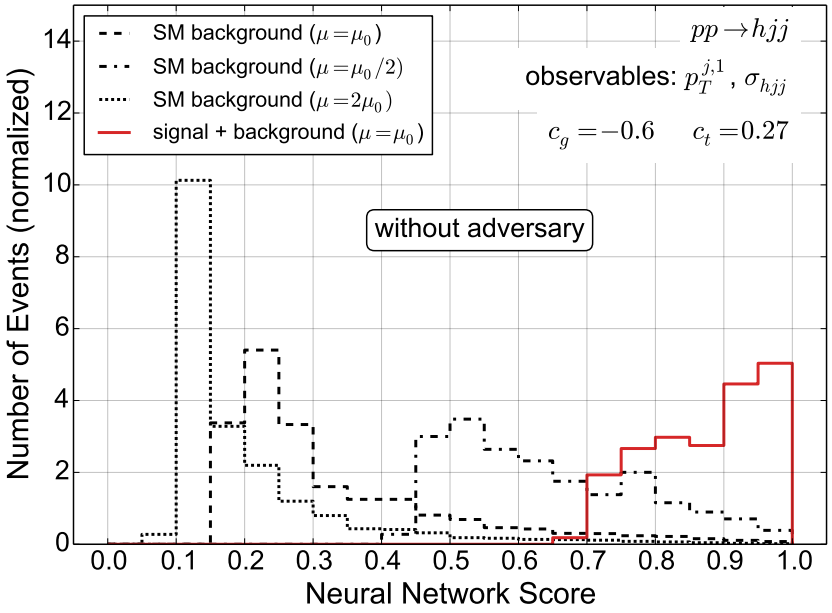
Solution: choose prior for distribution of σ_{tot}, e.g. asymmetric Gaussian

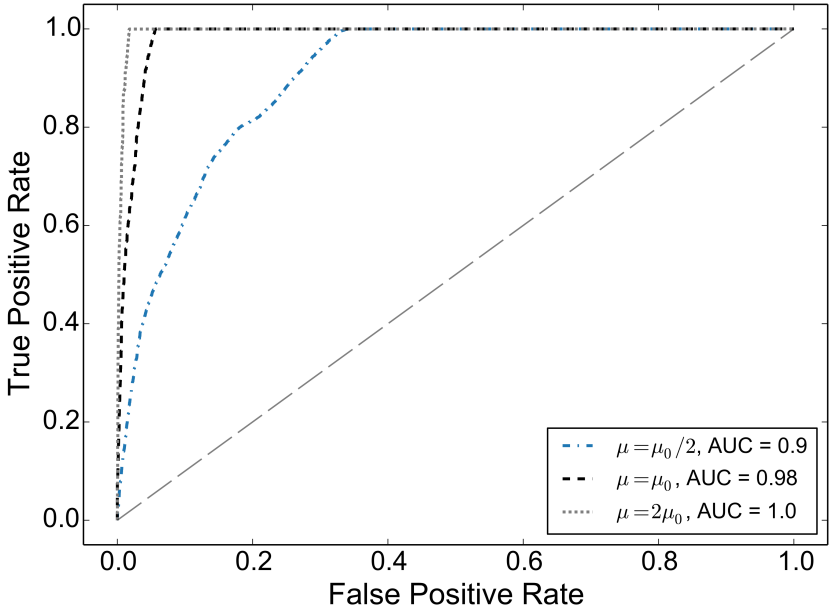




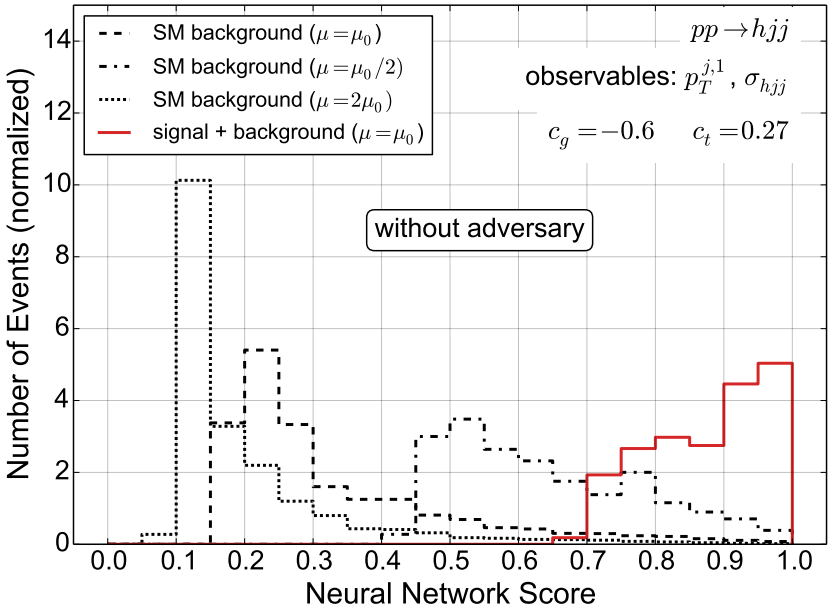
Classifier output independent of prior if signal and bkg. distributions overlap

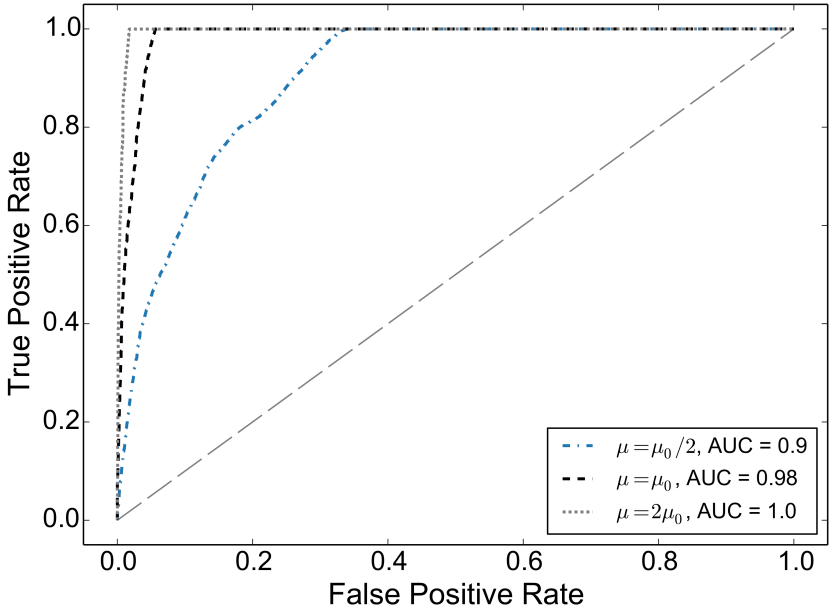
Example for a Pivotal Classifier

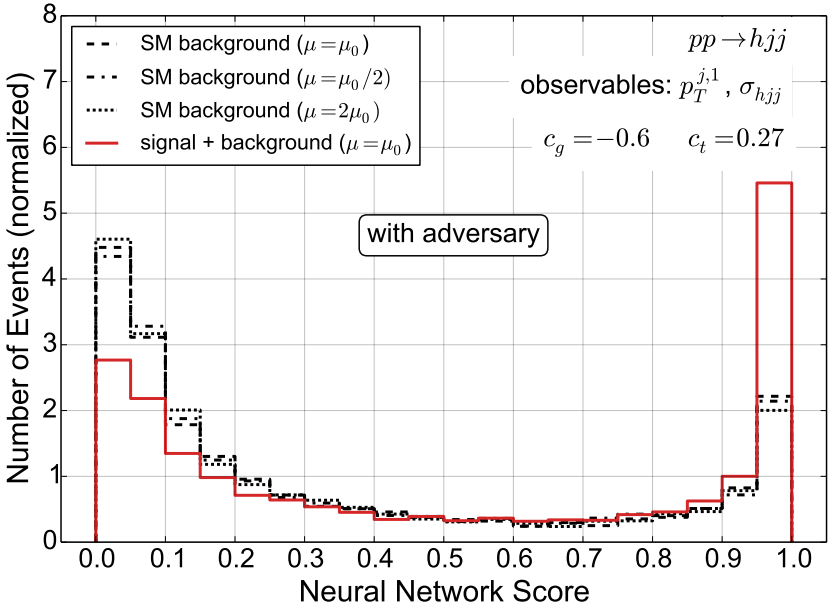


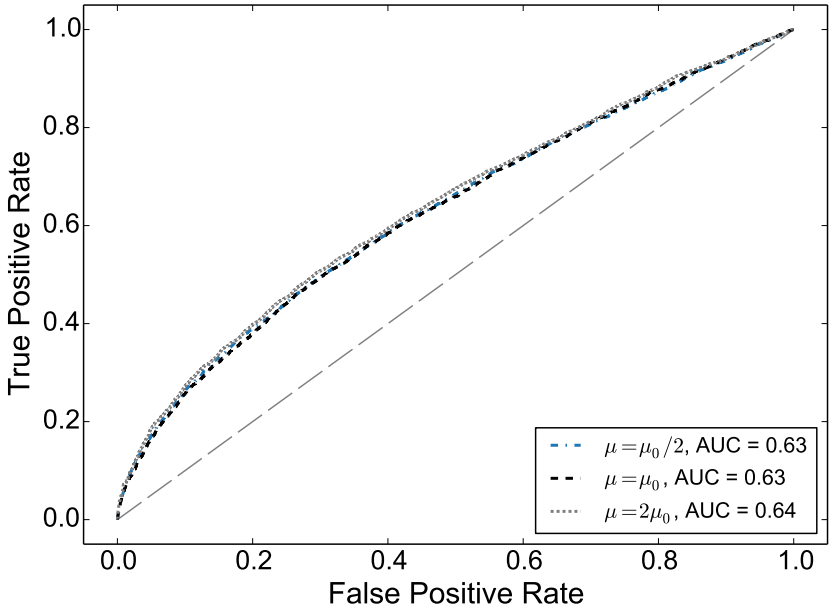


Example for a Pivotal Classifier







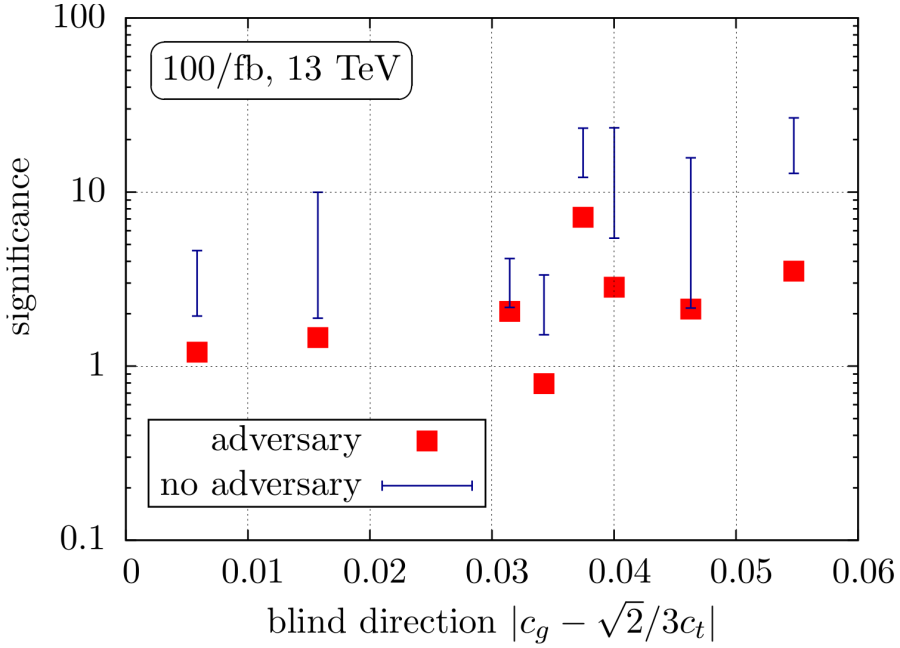


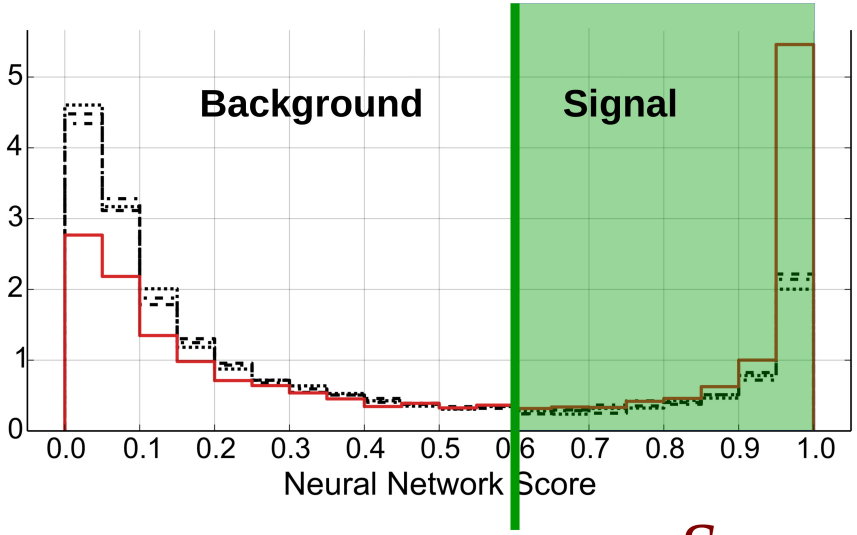
Application to the Physics Case

combine classifier output from two processes:

Process	Observables/Features
$pp \to hj$	$\sigma_{hj}, p_{T,j}, \eta_j$
$pp \to hjj$	$\sigma_{hjj}, p_{T,j_1}, \eta_{j_1}, p_{T,j_2}, \eta_{j_2}, \phi_{jj}, m_{jj}, \eta_{jj}$

compare expected median significance for normal and pivotal classifier near the blind direction $c_g = \sqrt{2}/3\, c_t$





maximize $\frac{S}{\sqrt{B}}$

$\to$ pivotal classifier always more conservative but also more robust

Summary

- pivotal classifiers (PC) can be trained using adversarial networks
- can be applied to archive independence w.r.t. theo. scale uncertainties
- PCs identify phase space regions which allow for classification AND exhibit reduced scale dependence
- trade-off: accuracy (classification power) vs. robustness
- PCs provide 'natural' treatment of systematic uncertainties in a ML setup