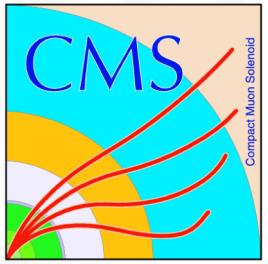
# Machine Learning within (DESY) CMS

- **A general overview with special emphasis on local activities**




Dirk Krücker - DESY CMS group

16.9.2019

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

DESY.

# Overview

## CMS activities

CMS is very active in applying Machine Learning and especially Deep Learning to a variety of subjects

- About ~25 dedicated ML contributions on conferences in 2019-present (not including physics analyses)
- Annual dedicated CMS Machine Learning workshops
- ML is transforming the field and the traditional workflows
  - New tools and new terminology

Main fields of applications

- **Object tagging** and object calibration especially Jets and taus are **flagship** applications
- **Physics analyses**
- In addition
  - Triggers
  - Reconstruction
  - Data Quality Monitoring
  - Simulation
  - **Computing work flows**

# Jet Tagging

# DeepJet

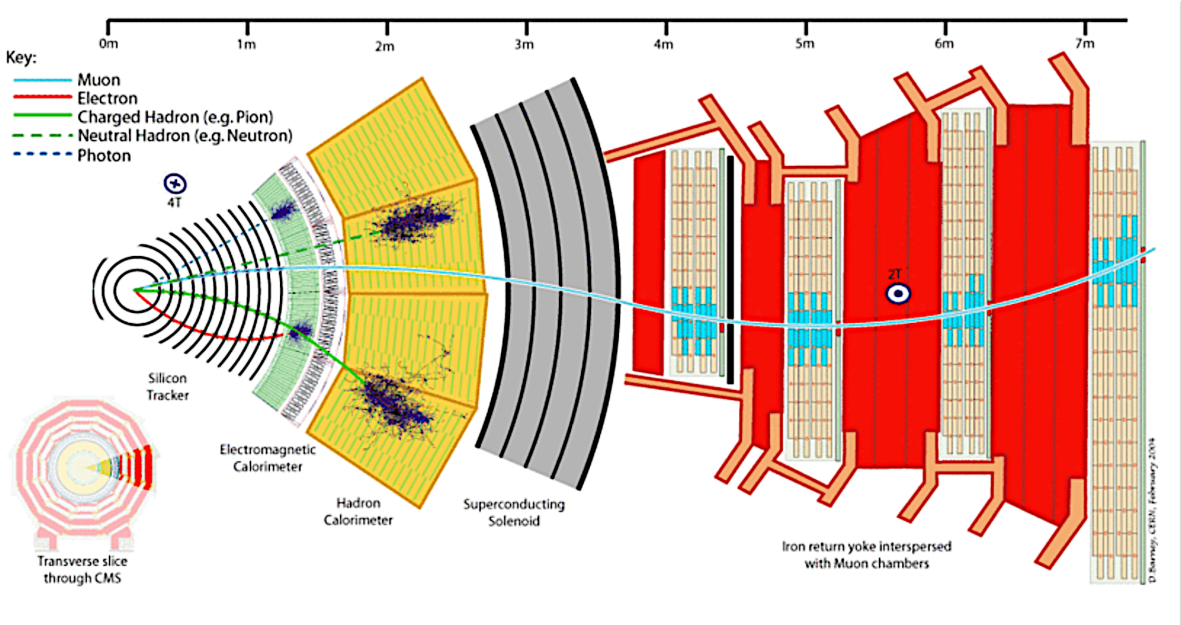
## The latest approach on b-quark jet tagging

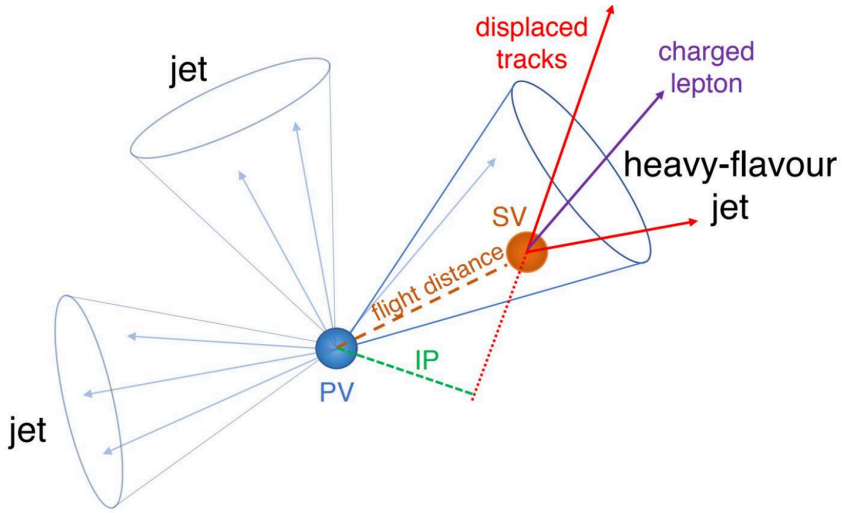
CMS uses a **Particle Flow** approach

- Identify each particle by the combined information of all sub-detector
- combine to jets (anti-kt)

Jet identification had been one of the first application of Deep Learning in HEP

- **b-quark jets** vs g/u/d/s-jets etc.
- The classical approach looks for secondary vertices
- There is plenty of information in the correlation between the individual tracks and vertices and global event characteristics

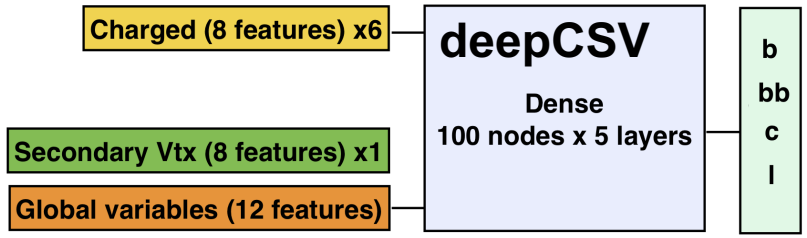


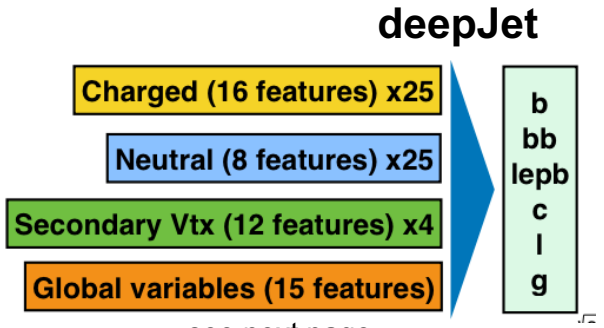



# Evolution of CMS b-tagging

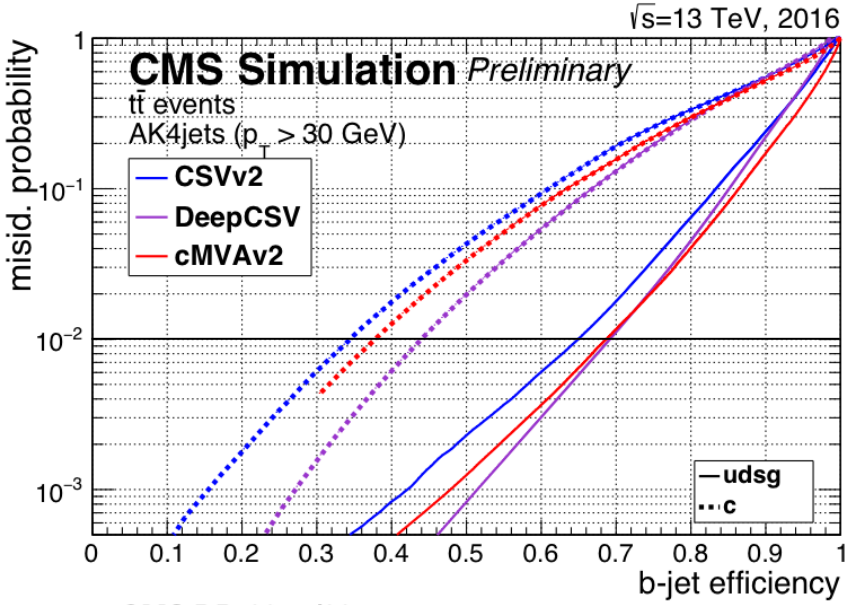
## A short history (for more details M. Verzetti ML4Jets WS 2018)

- 2015 handcrafted **CSVv2(nn)** and cMVAv2(BDT)
  - Mainly the same information
- 2017 **deepCSV**
- mid 2017 **deepJet** (DeepFlavour)
  - The gain is due to the usage of all available particle information in a deep network

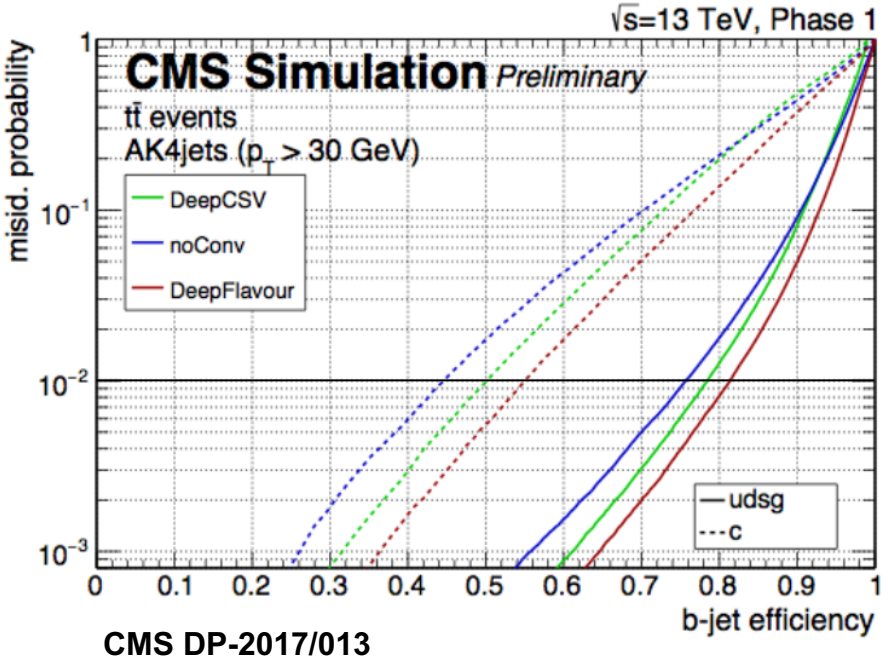



see next page

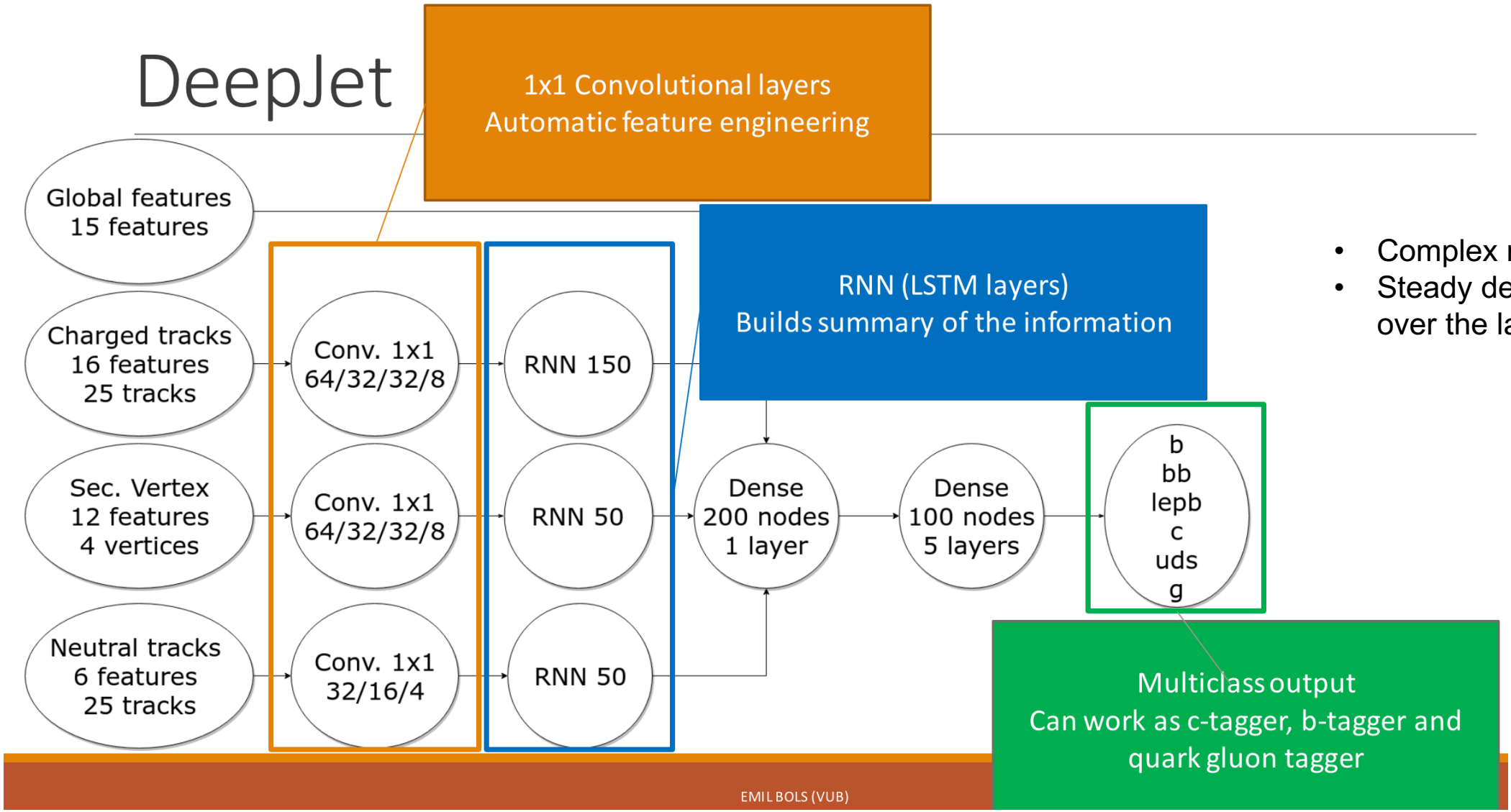



CMS DP-2017/005




CMS DP-2017/013

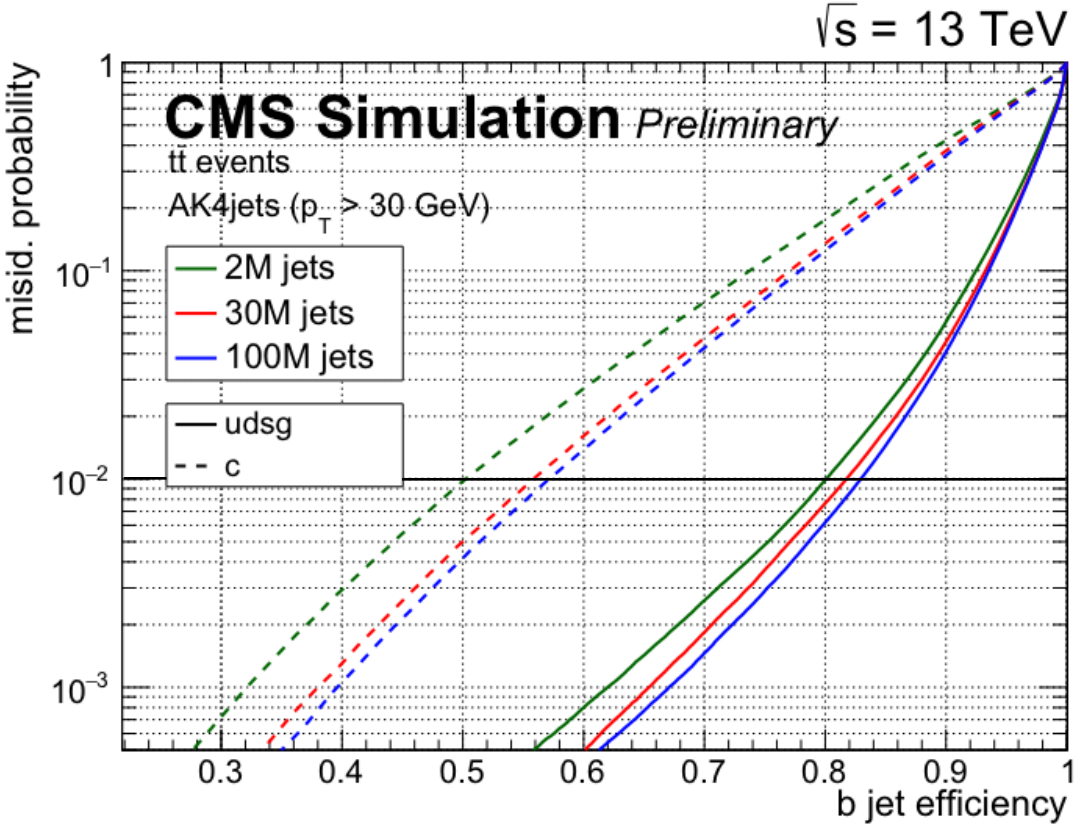
# DeepJet

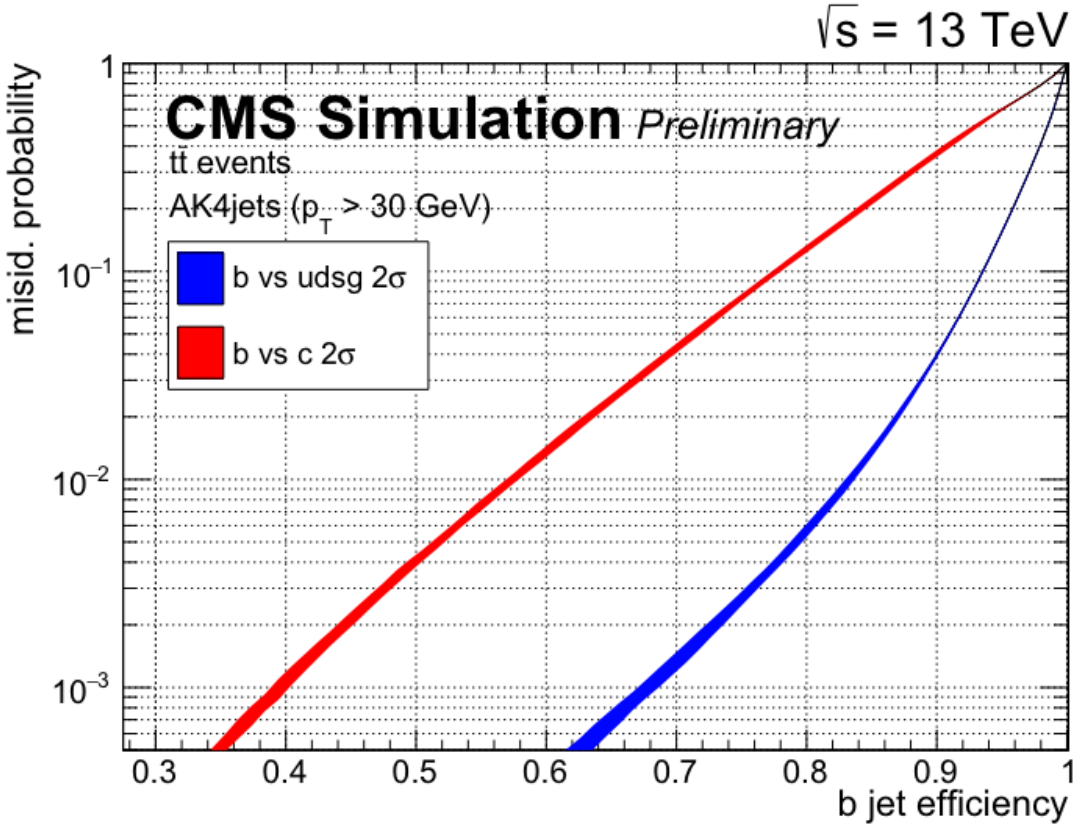



- Complex machinery
- Steady development over the last 3y

# Training DeepJet

## Big data



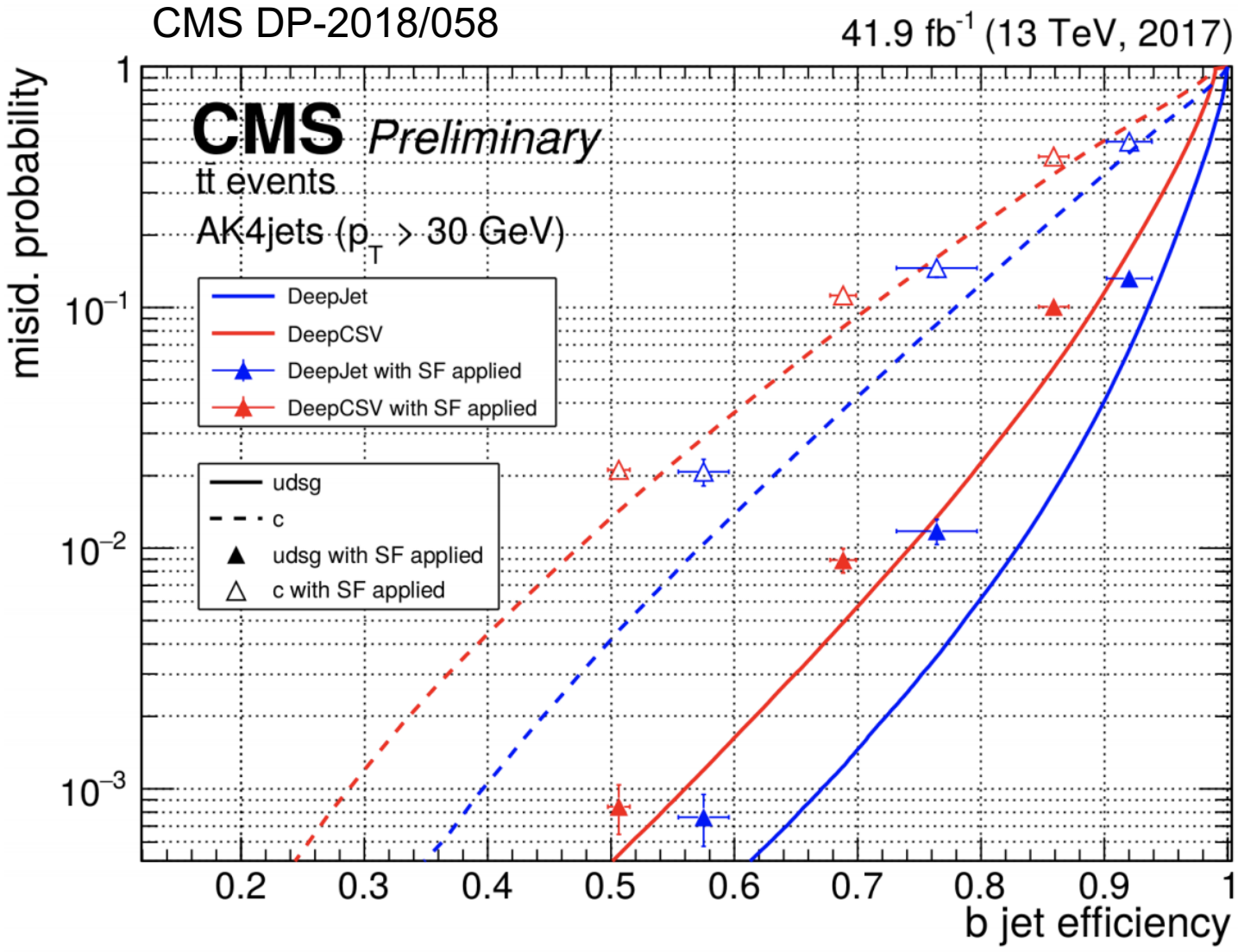

- Training needs large amount of data

- Different weight initialization
  - Training is stable and reproducible

# Scale factors

## Back to earth

- Training on MC
- Performance on data
  **Scale Factors (SF)** that describes the difference between simulation and data
  - Good performance on data but smaller efficiency

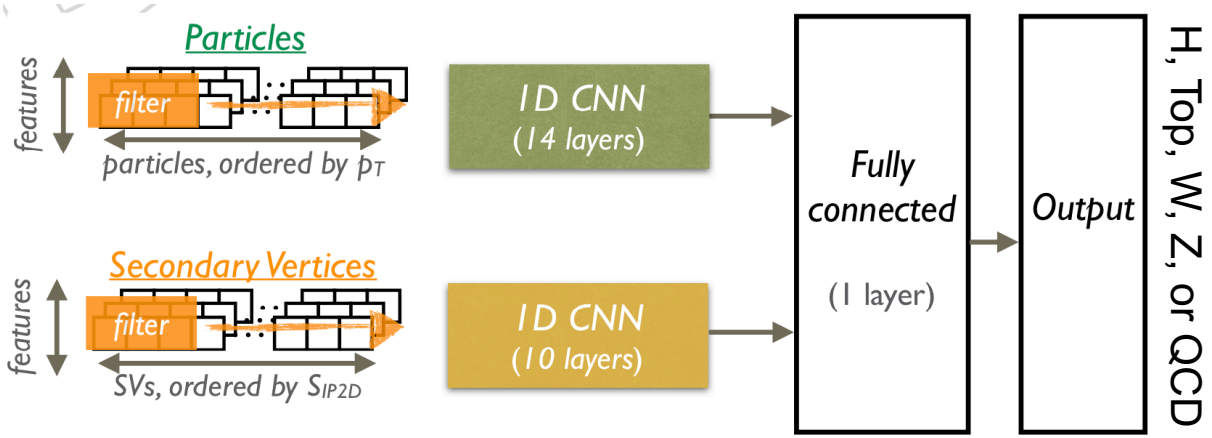


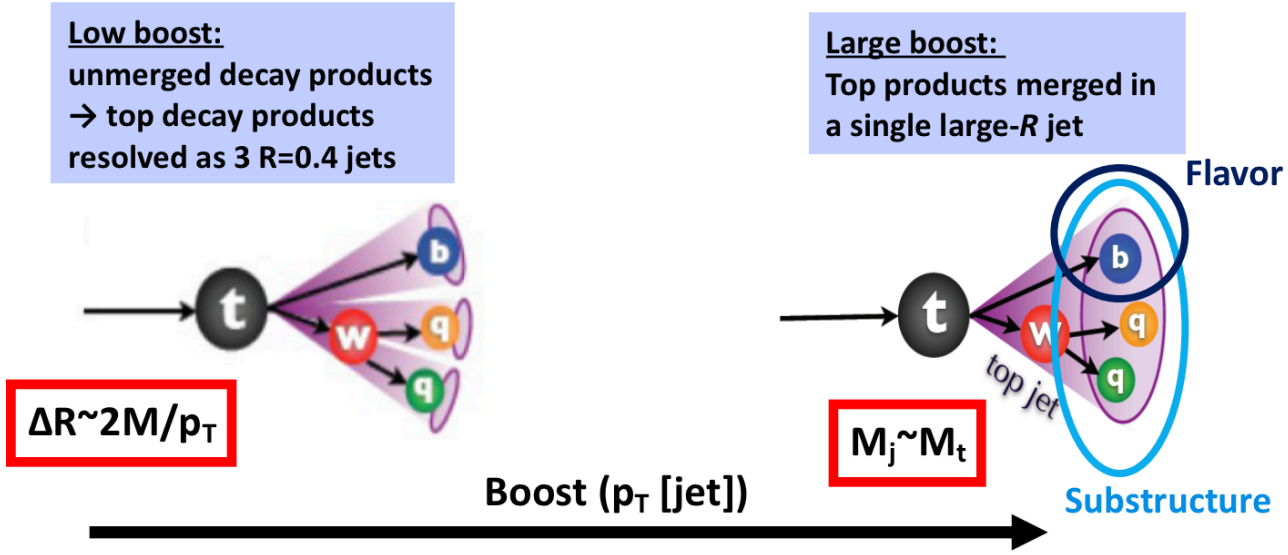
# Top etc. tagging

## Highly-boosted top quarks

- A top decays into a b quark and a W ⟶ lep,had
- If the momentum is large the form a common jet
  - **Boosted topology in fat jets** (AK8)
- Top-tagging ⟺ identify such jets

Typical problem how to handle a varying number of particles, here: 1d CNN

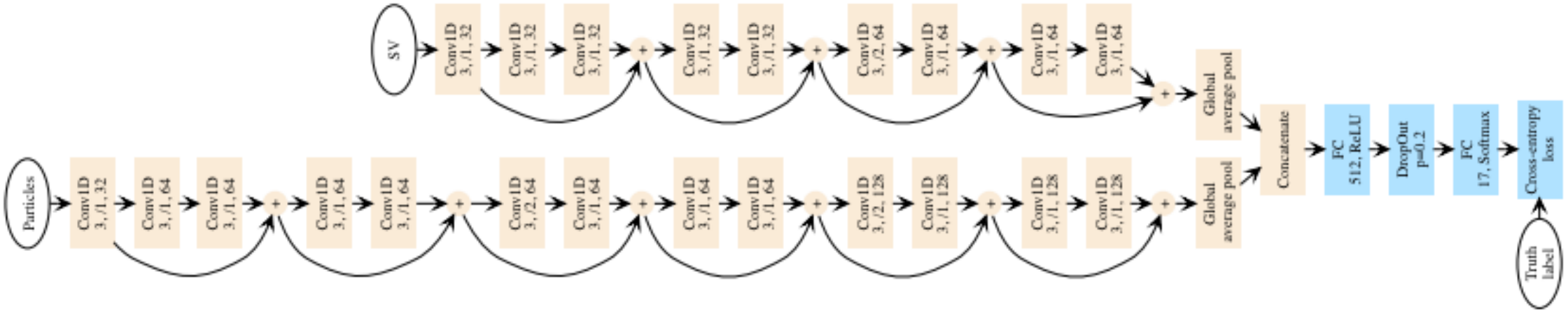



DeepAK8

- Up to 100 particles
- Each particle comes with 42 features
- 14 layers is indeed deep
- Up to 7 SV with 7 features
- Trained with 50 million jets
- **Residual net** (short cuts between layers)

**CMS publication will appear soon**




Convolution allows to handle a varying number of particle within a jet

# How to feed in a changing number of objects

## A common subject

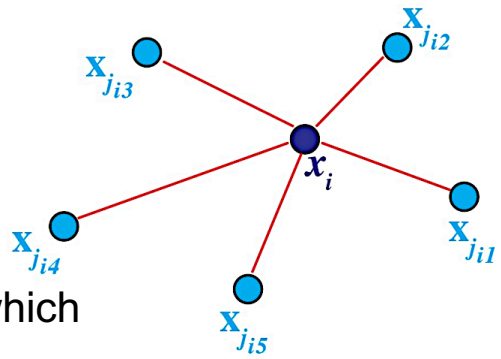
Many Deep Learning approaches have been developed for imaging, i.e. dense regular grid of points; or text processing, i.e. sequences

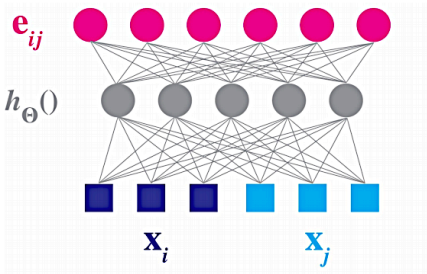
- Not straightforward how to connect to typical HEP problems
  - Jet images in $(\eta, \phi)$ plane are sparse
  - Varying number of particles in cone from jet to jet
- Standard solution so far
  - Recurrent nets: $p_T$ ordered particles
  - Convolution: particles as 1d string of inputs
- New approach: Geometric learning or
  - **Graph networks**
    (Hot subject on this year NIPS dozens of papers)

- H. Qu and L. Gouskos.
  „ParticleNet: Jet Tagging via Particle Clouds“
  https://arxiv.org/abs/1902.08570
  - Bsed on Y. Wang et al.
    „Dynamic Graph CNN for Learning on Point Clouds“
- M. Fey and J.E. Lenssen.
  „Fast Graph Representation Learning with PyTorch Geometric“
  https://github.com/rusty1s/pytorch_geometric

# EdgeConv network

## Particle Cloud

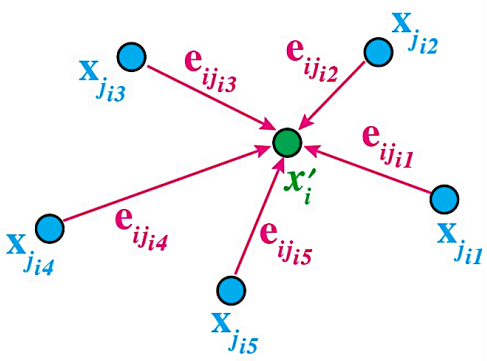



- Compute k-nn Graph
  - Defines a local neighborhood on which a convolution operation work
  - For the first layer this is the ($\Delta\eta, \Delta\phi$,$p_T$) plane wrt. jet axis
- Apply EdgeConv, here

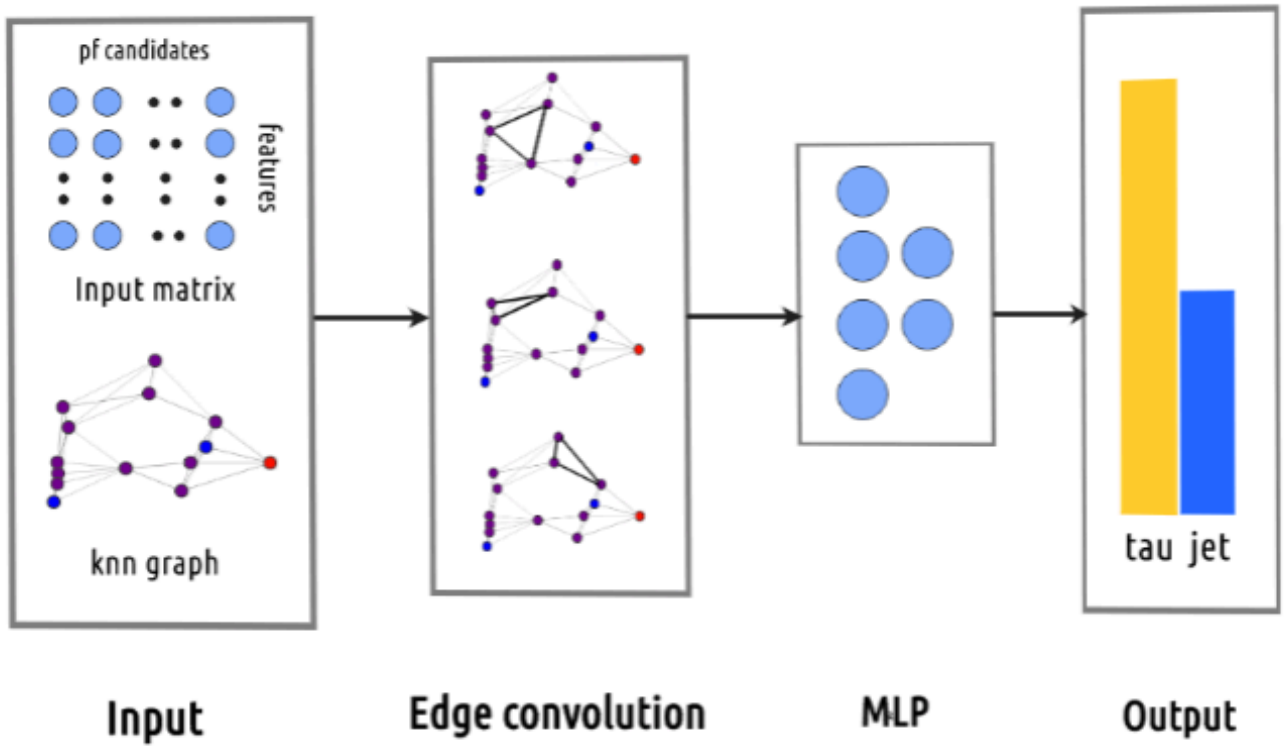
$$e'_{ijm} = \mathrm{ReLU}(\boldsymbol{\theta}_m \cdot (\mathbf{x}_j - \mathbf{x}_i) + \boldsymbol{\phi}_m \cdot \mathbf{x}_i)$$

$$x'_{im} = \max_{j:(i,j)\in\mathcal{E}} e'_{ijm}$$




- Dynamic Graph CNN means the recalculation of the k-nn in the $x'$ space

# EdgeConv network

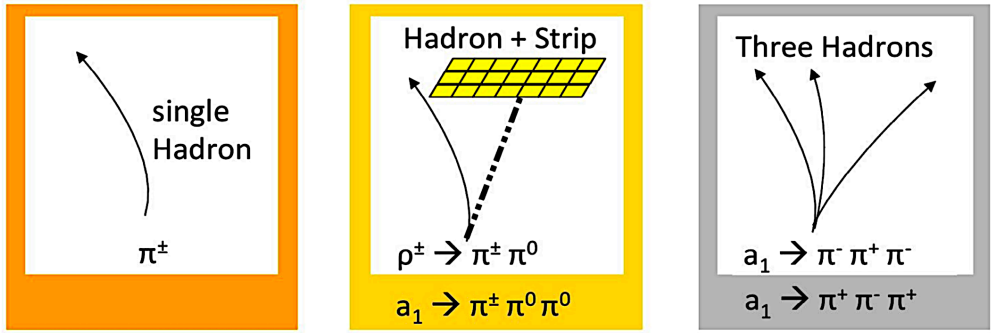


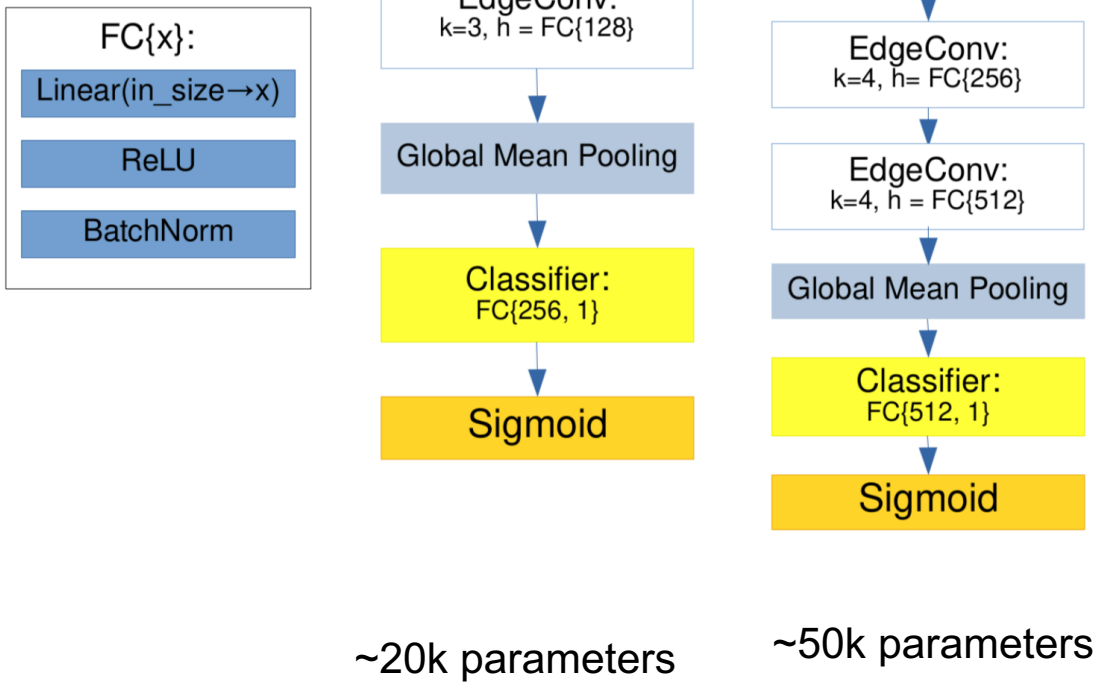

Thanks to Leonid Didukh

# Tau-lepton tagging

## A newly started project with Graph Networks – no official CMS results yet

- CMS investigates deep NN for tau-lepton identification
- Similar complexity as the shown b tagger
- We started to investigate a Graph Network approach
- Promising results similar performance with a model that contains 10 times less parameters
  - Time advantage in interference




~20k parameters

~50k parameters

# Searches

# Searches

## Examples for ML methods in searches

Two example analyses (there are **many more using ML**)

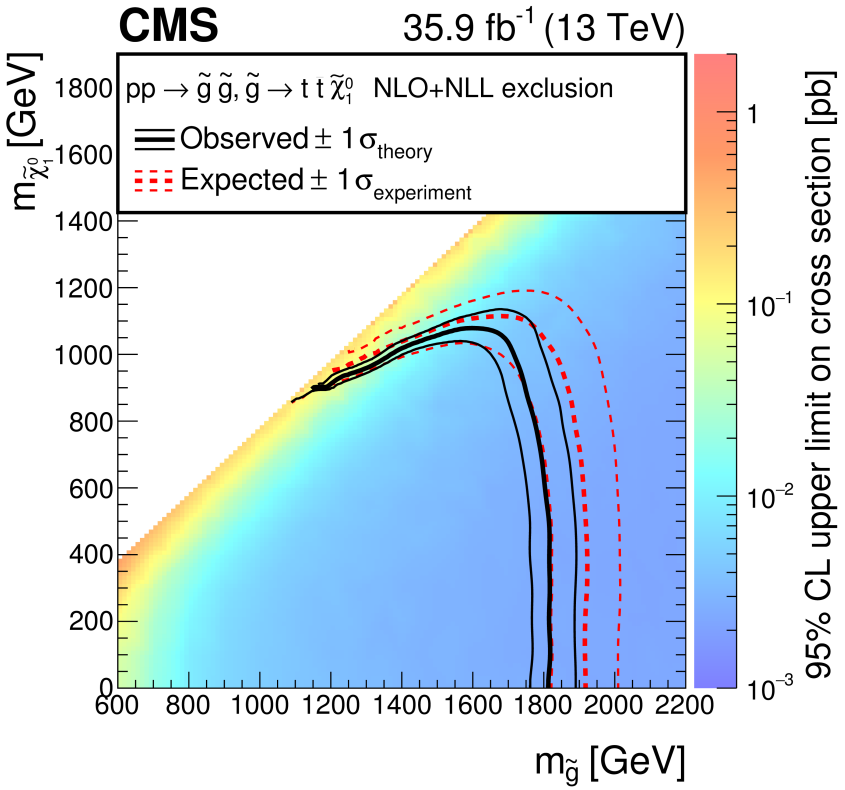
- Supersymmetry
- Higgs

Please note!

- These are ongoing analyses
- Results are not approved
- Therefore no numbers, no data plots

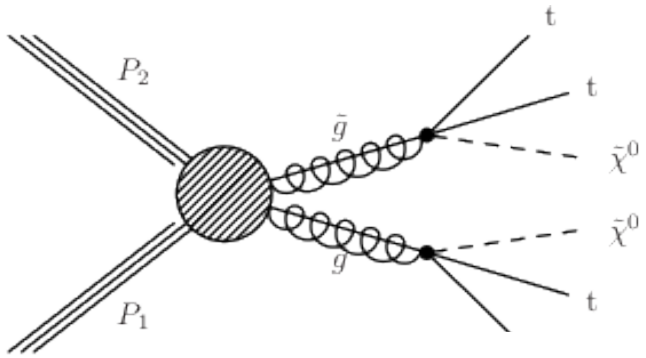
# Ex.1: Search for supersymmetry in events with one lepton

## Search for supersymmetry in events with one lepton and multiple jets exploiting the angular correlation between the lepton and the missing transverse momentum in proton-proton collisions at √s = 13 TeV

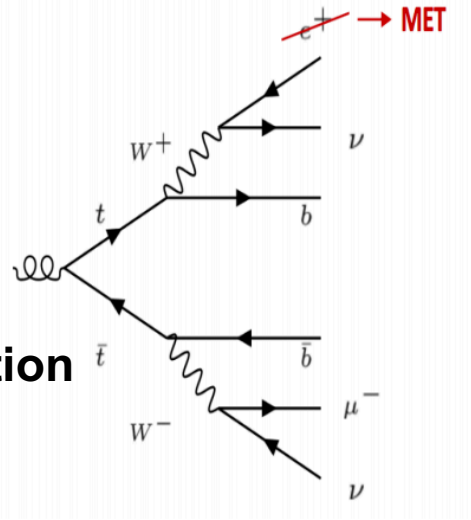



published Run II result on 2016 data

- **SUSY** signal: T1tttt
  2 **gluinos** that decay
  to 4 top quarks
  + 2 **neutralinos**

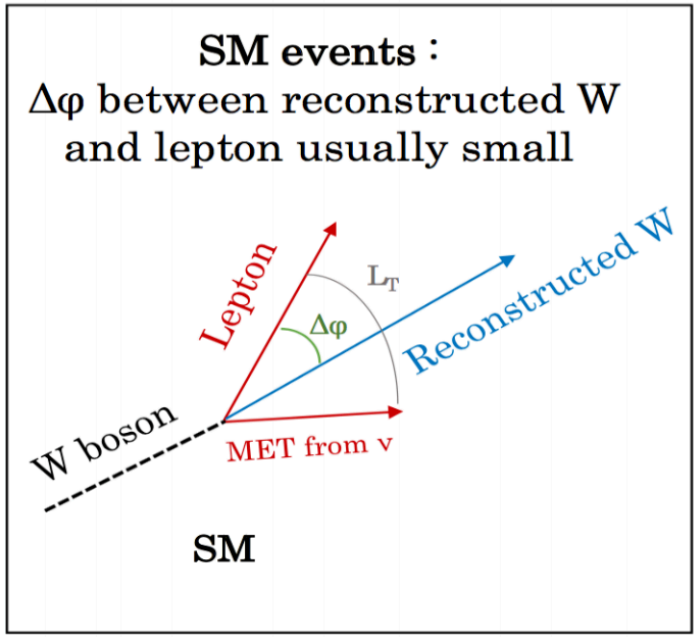


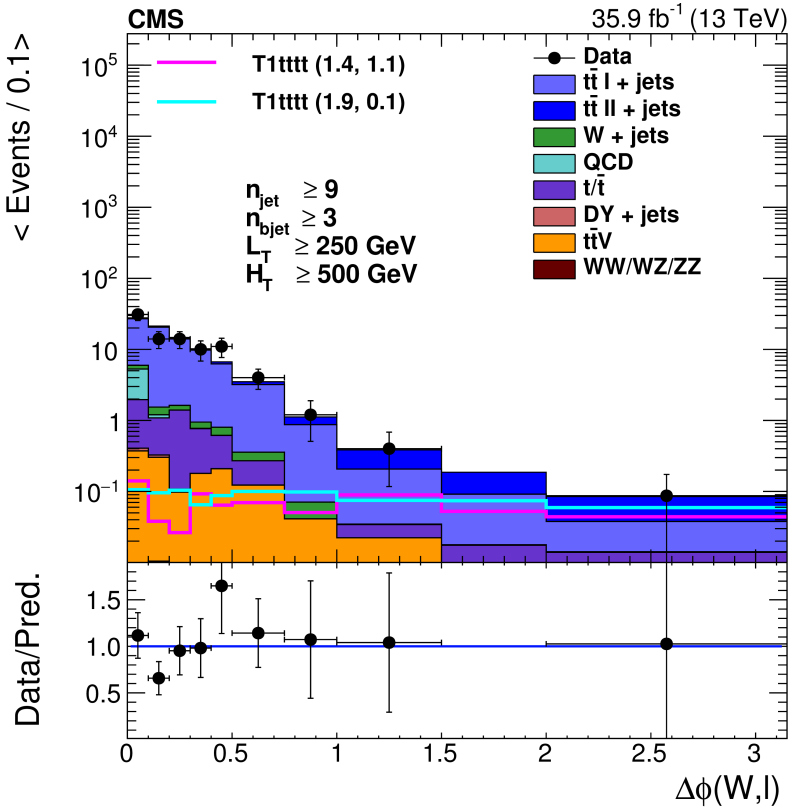

- Important **SM** background
  top-anti-top pair (**ttbar**) with
  2 leptons where
  **one lepton is missed in the reconstruction**



# SUSY deltaPhi

## Previous approach




- Can we do better with a neural network?
  - Sure but
- How to connect the NN response to the observed data to estimate the background in the signal region?

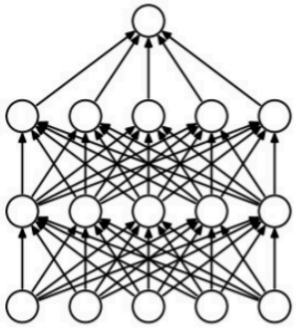
Estimate the background
in the high dPhi region by looking
into the low dPhi region at different jet multiplicities

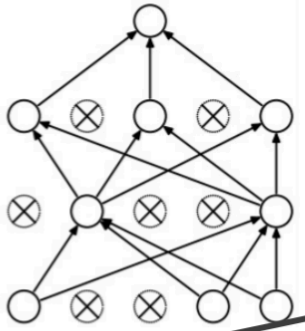
# DNN DETAILS

100 epochs, batch size = 2048
Dropout with rate of 0.01

Early stopping to stop training once the model performance stops improving on a hold out validation dataset with patience of 10 epochs.

Preselect events :
- I Lep , no Veto, LT > 250 , HT > 500, nJ > 3 nbJ >= 1, iso_track veto





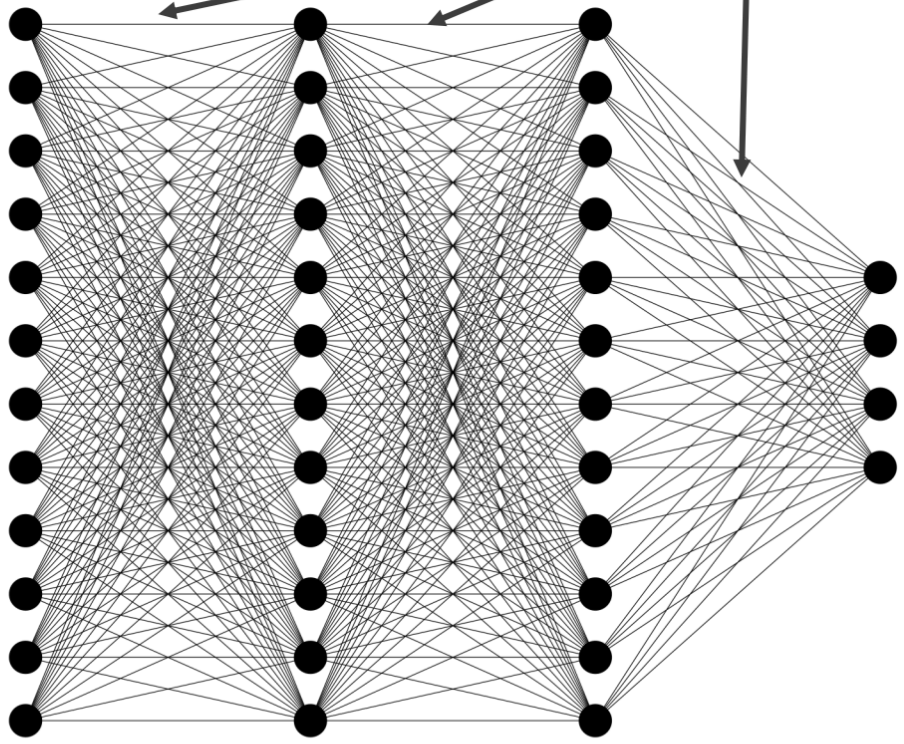
**regularisation**

Dropout layer

**Input Features**

$$MET, MT, Jet_{p_T}^{1/2}$$
$$Lep_{p_T}, L_T, H_T, n_{bJets}$$
$$n_{Top}, n_{jets}, \Delta\phi, Lep_{relIso}$$
$$Lep_{MinilIso}, IsoTrack_{p_T}$$
$$IsoTrack_{MT_2}$$
$$m_{\tilde{g}}, m_{\chi_1^0}$$

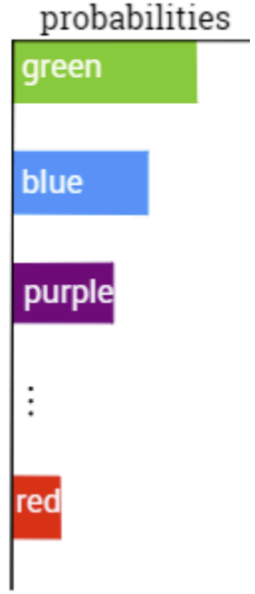
17 input variable



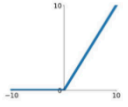
Class 0 → DiLep Ttbar
Class 1 → SemiLep Ttbar
Class 2 → Wjets + others
Class 4 → Signal

SoftMax

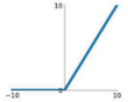
$$\frac{e_1^z}{\sum_{k=1}^K e_k^z}$$
$$\frac{e_2^z}{\sum_{k=1}^K e_k^z}$$
$$\frac{e_3^z}{\sum_{k=1}^K e_k^z}$$
$$\vdots$$
$$\frac{e_K^z}{\sum_{k=1}^K e_k^z}$$



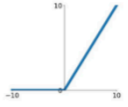

**ReLU**
$\max(0, x)$



**ReLU**
$\max(0, x)$



**ReLU**
$\max(0, x)$

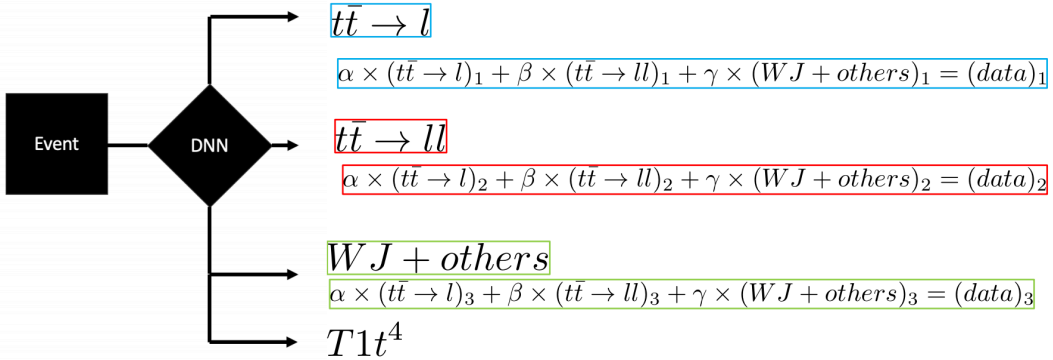


# Multiclassification

## Data augmented background estimation

- 4 classes
  - 3 class represent different backgrounds
  - 1 class is the search region
  - The simulation is normalized to the data in the background classes by solving the equation system

| background | $\alpha$ | $\beta$ | $\gamma$ |
|---|---|---|---|
| 1500_1000 | 0.84± 0.01 | 1.0± 0.05 | 0.72 ± 0.03 |
| 1500_1200 | 0.84 ± 0.01 | 0.98 ± 0.04 | 0.72 ± 0.03 |
| 1600_1100 | 0.84 ± 0.01 | 0.98 ± 0.05 | 0.73 ± 0.03 |
| 1700_1200 | 0.84 ± 0.01 | 0.98 ± 0.05 | 0.73 ± 0.03 |
| 1800_1300 | 0.84 ± 0.01 | 0.98 ± 0.04 | 0.71 ± 0.03 |
| 1900_100 | 0.84 ± 0.01 | 0.97 ± 0.05 | 0.71 ± 0.03 |
| 1900_1000 | 0.84 ± 0.01 | 0.96 ± 0.04 | 0.71 ± 0.03 |
| 1900_800 | 0.84 ± 0.01 | 0.94 ± 0.05 | 0.72 ± 0.03 |
| 2200_100 | 0.84 ± 0.01 | 0.96 ± 0.05 | 0.73 ± 0.03 |
| 2200_800 | 0.84 ± 0.01 | 0.96 ± 0.04 | 0.71 ± 0.03 |




$$Y_i = A_{ij} X_i$$

$Y_i \to$ data counts
$A_{ij} \to$ MC counts
$X_i \to$ scale factor for each background
$i \to$ number of CRs
$j \to$ background number

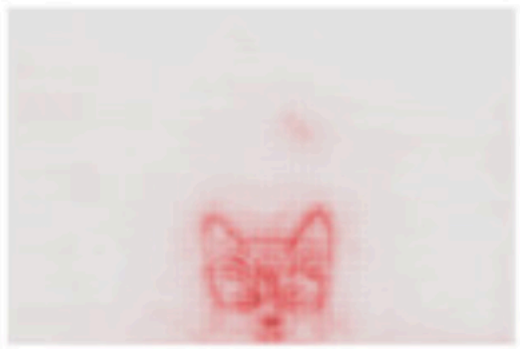
$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \times \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} data_1 \\ data_2 \\ data_3 \end{bmatrix}$$

- **Creates a nice data/MC agreement -> estimate Background in signal region**
  (not shown here)
- Independent of the signal point

# How to know what is relevant?

## Opening the neural net black box in a Higgs search

- Complex neural nets do not tell us how they come to a decision
  - $10^{th}$ thousands of parameters
  - Several dozens of input variables
- Important to understand what is driving the decision
  - If the network is just a chain of matrix multiplication and function mappings why not just do a Taylor expansion
  - arXiv:1512.02479



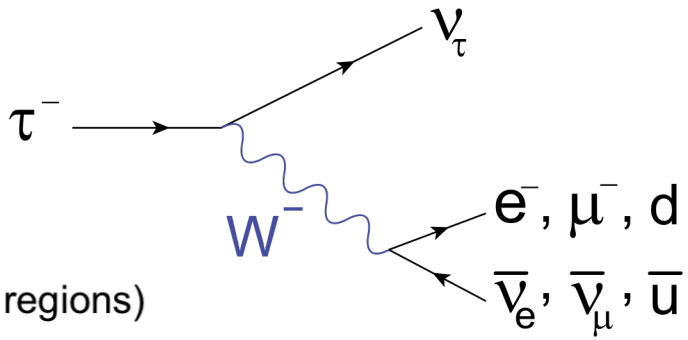
Deep Taylor Decomposition

- **Higgs $\longrightarrow \tau\,\tau$**
  - Observation published last year (CMS) *Phys. Lett. B* **779** (2018) 283
  - New study with a multiclass NN ongoing

## Relevance propagation of variables as applied in H $\longrightarrow \tau\tau$

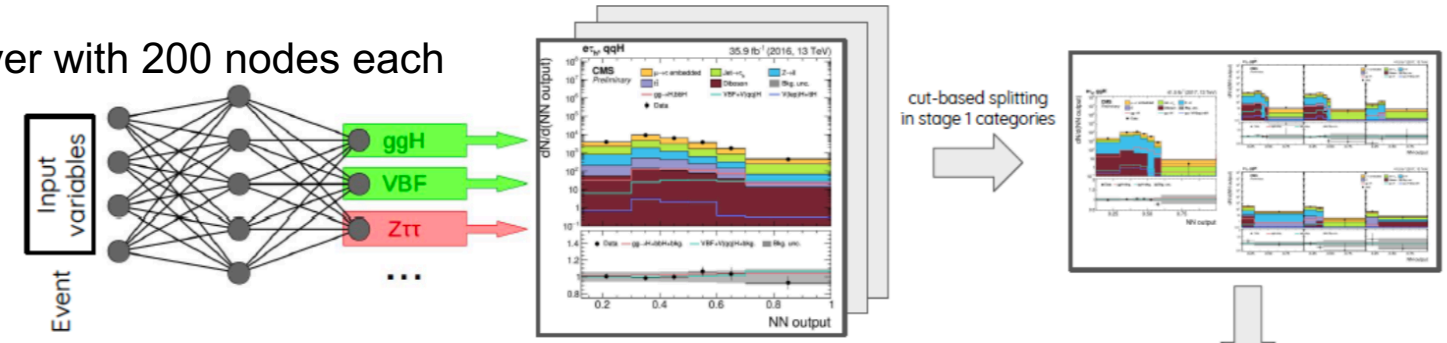
Teresa Lenz, Mareike Meyer, Alexei Raspereza for the HIG-18-032 team

# Analysis strategy

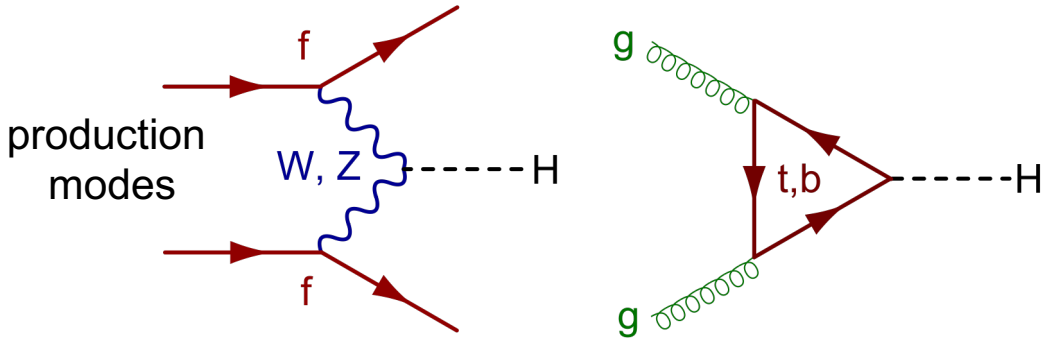



- four most sensitive final states of ττ-pair studied: **eμ, eτh, μτh, τhτh**
- loose baseline selection (trigger requirements, suppression of large backgrounds)
- **multi-class NN** with **2 signal classes (ggH, qqH) & several background classes** (control regions)
- selection & validation of NN input variables based on 1D and 2D GoFs → **relevance propagation of variables**

2 hidden layer with 200 nodes each




cross section



# Relevance propagation: idea

S.Wunsch, R. Friese, R.Wolf, and G. Quast, Computing and Software for Big Science 2 (Sep, 2018) 5, doi:10.1007/s41781-018-0012-, arXiv:1803.08782

- relate output space of NN to input space

➡ identify characteristics of input space that have large influence on output for a given task

- **decompose NN function into Taylor expansion in each element of the input space**
- Taylor coefficients contain information about the sensitivity of the NN response to the inputs
- dependence on phase space: mean of absolute values of Taylor coefficients evaluated for all elements of test sample

set of input variables, evaluated for element k of test sample

$$\langle t_i \rangle \equiv \underbrace{\frac{1}{N}\sum_{k=1}^{N}}_{\text{sum over test sample of size N}} \left| t_i( \{x_j\}|_k) \right| \qquad i \in \mathcal{P}(\{x_j\})$$

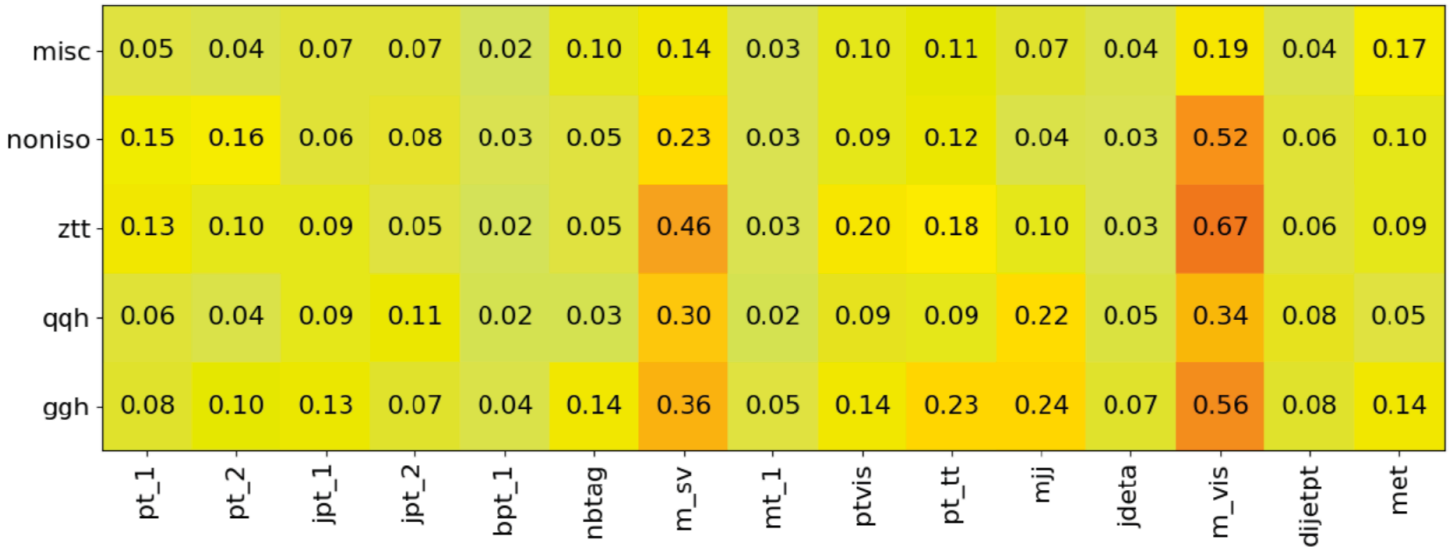
$t_i$: $i^{\text{th}}$ Taylor coefficient

**Size of the Taylor coefficients as sensitivity metric**

- **first order** Taylor coefficient: **influence of single input elements**
- **second order** Taylor coefficent: **influence of pair-wise or auto-correlations**

$$T(x,y) = f(a,b) + (x-a)f_x(a,b) + (y-b)f_y(a,b) + \frac{1}{2!}\Big((x-a)^2 f_{xx}(a,b) + 2(x-a)(y-b)f_{xy}(a,b) + (y-b)^2 f_{yy}(a,b)\Big) + \cdots$$

# Application to H → ττ: 1st order coefficients




- **we can learn which variables were important for the NN to identify the respective process**

# 2nd order coefficients

**2016, $\tau_h\tau_h$ channel**

| ztt | | | noniso | | | misc | |
|---|---|---|---|---|---|---|---|
| m_vis | 2.52 | m_vis | m_vis | 1.01 | m_sv | m_vis | 0.60 |
| m_vis | 1.43 | | m_vis | 0.53 | m_vis | m_vis | 0.60 |
| m_sv | 1.36 | m_sv | m_vis | 0.50 | m_sv | m_sv | 0.25 |
| ptvis | 0.96 | pt_1 | m_vis | 0.40 | pt_tt | m_vis | 0.22 |
| pt_tt | 0.90 | pt_2 | m_vis | 0.36 | ptvis | m_vis | 0.21 |
| m_vis | 0.82 | m_vis | met | 0.27 | pt_1 | m_vis | 0.21 |
| m_vis | 0.79 | m_vis | dijetpt | 0.23 | | m_vis | 0.20 |
| | 0.61 | | m_sv | 0.23 | pt_2 | m_vis | 0.17 |
| | 0.47 | jpt_1 | m_vis | 0.21 | jpt_1 | m_vis | 0.17 |
| pt_tt | 0.34 | m_sv | m_sv | 0.20 | m_sv | ptvis | 0.17 |
| m_sv | 0.30 | ptvis | m_vis | 0.19 | m_sv | pt_tt | 0.16 |
| dijetpt | 0.29 | pt_tt | m_vis | 0.18 | m_vis | met | 0.16 |
| m_vis | 0.29 | | pt_1 | 0.18 | | met | 0.15 |
| m_vis | 0.29 | m_sv | ptvis | 0.16 | m_vis | dijetpt | 0.15 |
| dijetpt | 0.26 | | pt_2 | 0.15 | | m_sv | 0.14 |
| met | 0.26 | m_sv | pt_tt | 0.14 | jpt_1 | m_sv | 0.11 |
| m_vis | 0.25 | jpt_2 | m_vis | 0.13 | mjj | m_vis | 0.11 |
| met | 0.22 | nbtag | m_vis | 0.13 | jpt_2 | m_vis | 0.11 |
| m_sv | 0.21 | pt_1 | m_sv | 0.11 | m_sv | met | 0.10 |
| m_sv | 0.20 | jpt_1 | m_sv | 0.11 | pt_1 | m_sv | 0.09 |
| ptvis | 0.19 | m_sv | met | 0.11 | m_sv | dijetpt | 0.09 |
| | 0.19 | | pt_tt | 0.10 | | pt_tt | 0.09 |
| | 0.17 | | met | 0.10 | | nbtag | 0.08 |
| pt_tt | 0.16 | mjj | m_vis | 0.10 | nbtag | m_vis | 0.08 |
| m_vis | 0.16 | pt_1 | pt_2 | 0.10 | | ptvis | 0.08 |
| mt_1 | 0.15 | | ptvis | 0.10 | pt_2 | m_sv | 0.08 |
| m_vis | 0.15 | pt_2 | m_sv | 0.09 | jpt_2 | m_sv | 0.07 |
| mjj | 0.14 | bpt_1 | m_vis | 0.08 | | jpt_2 | 0.07 |
| m_vis | 0.13 | | jpt_2 | 0.08 | ptvis | pt_tt | 0.07 |
| m_sv | 0.13 | met | met | 0.08 | m_sv | mjj | 0.07 |
| | 0.12 | jdeta | m_vis | 0.08 | | jpt_1 | 0.06 |
| ptvis | 0.12 | ptvis | pt_tt | 0.07 | mt_1 | m_vis | 0.06 |
| m_vis | 0.12 | pt_1 | met | 0.07 | met | met | 0.06 |
| m_sv | 0.12 | | dijetpt | 0.07 | jdeta | m_vis | 0.06 |
| pt_tt | 0.11 | jpt_2 | m_sv | 0.07 | | mjj | 0.05 |
| met | 0.11 | | jpt_1 | 0.06 | bpt_1 | m_vis | 0.05 |
| met | 0.11 | jpt_2 | met | 0.06 | jpt_1 | ptvis | 0.05 |
| | 0.10 | jpt_1 | ptvis | 0.06 | pt_tt | met | 0.05 |
| | 0.10 | m_vis | dijetpt | 0.06 | jpt_1 | met | 0.05 |
| dijetpt | 0.10 | pt_1 | pt_1 | 0.06 | jpt_2 | met | 0.05 |
| | 0.10 | pt_2 | met | 0.06 | nbtag | m_sv | 0.05 |
| m_sv | 0.09 | nbtag | met | 0.06 | nbtag | met | 0.05 |
| | 0.09 | mt_1 | m_vis | 0.06 | jpt_1 | dijetpt | 0.05 |
| | 0.09 | jpt_1 | met | 0.06 | | pt_1 | 0.05 |
| | 0.09 | pt_1 | pt_tt | 0.06 | ptvis | met | 0.05 |
| | 0.08 | pt_1 | dijetpt | 0.05 | jpt_1 | jpt_2 | 0.05 |
| | 0.08 | m_sv | mjj | 0.05 | jpt_1 | jpt_1 | 0.05 |
| | 0.08 | jpt_1 | jpt_1 | 0.05 | pt_1 | pt_2 | 0.04 |
| | 0.08 | pt_2 | dijetpt | 0.05 | pt_1 | met | 0.04 |
| | 0.07 | pt_1 | ptvis | 0.05 | pt_1 | pt_tt | 0.04 |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| jpt_1 | ptvis | 0.11 | dijetpt | dijetpt | 0.09 | nbtag | | | | | | | | | |
| bpt_1 | m_vis | 0.11 | ptvis | dijetpt | 0.09 | nbtag | | | | | | | | | |
| nbtag | m_sv | 0.11 | mjj | dijetpt | 0.09 | jpt_1 | | | | | | | | | |
| | met | 0.11 | jdeta | m_vis | 0.09 | ptvis | | | | | | | | | |
| pt_tt | mjj | 0.10 | jpt_2 | jpt_2 | 0.09 | pt_tt | | | | | | | | | |
| jpt_1 | jpt_2 | 0.10 | | pt_tt | 0.08 | | | | | | | | | | |
| ptvis | ptvis | 0.10 | | ptvis | 0.08 | jdeta | | | | | | | | | |
| pt_1 | pt_2 | 0.10 | m_sv | mt_1 | 0.08 | ptvis | | | | | | | | | |
| pt_tt | pt_tt | 0.10 | bpt_1 | m_vis | 0.08 | | | | | | | | | | |
| mjj | mjj | 0.10 | pt_tt | pt_tt | 0.08 | bpt_1 | | | | | | | | | |
| | pt_2 | 0.10 | jpt_2 | mjj | 0.08 | | | | | | | | | | |
| jpt_1 | pt_tt | 0.10 | jpt_1 | mjj | 0.08 | bpt_1 | | | | | | | | | |
| jpt_1 | jpt_1 | 0.10 | pt_tt | dijetpt | 0.08 | pt_tt | | | | | | | | | |
| | jpt_1 | 0.09 | | jpt_2 | 0.08 | m_sv | | | | | | | | | |
| ptvis | mjj | 0.09 | pt_1 | dijetpt | 0.07 | | | | | | | | | | |
| ptvis | dijetpt | 0.09 | jpt_1 | jpt_1 | 0.07 | pt_1 | | | | | | | | | |
| jpt_1 | mjj | 0.09 | jpt_1 | ptvis | 0.07 | | | | | | | | | | |
| pt_tt | met | 0.09 | | jpt_1 | 0.07 | pt_1 | | | | | | | | | |

- 50 highest ranked variables or combinations

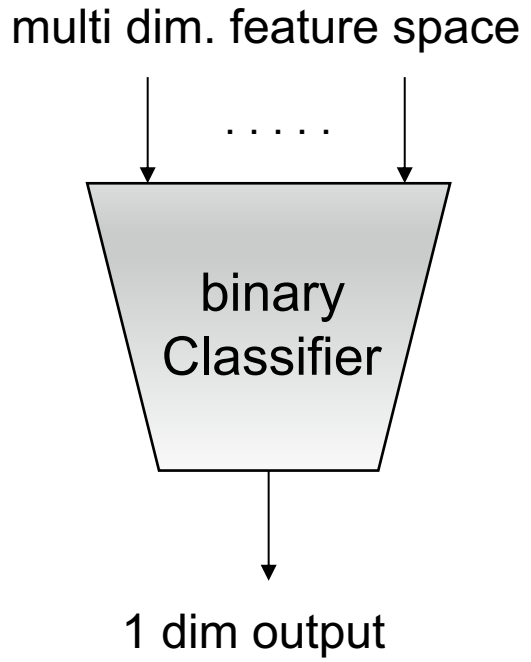
# Machine Learning Research for HEP

# Learning for Discovery - The Asimov Loss

## Setting the stage

**How to optimize for a search?**

- We are looking for the best search bin
- Background by some sideband measurement with systematic uncertainty
  - **s** and **b** with systematic uncert. $\sigma_b = e_{sys}\, b$
- The best statistical answer to this problem:
  **Asymptotic formulae for likelihood-based tests of new physics** by CCGV
  arXiv:1007.1727 [physics.data-an]
  - 1 bin Poisson „expected“ discovery significance we call this: **„*Asimov*“ significance**



# Learning for Discovery

## *Asimov* expected discovery significance

- *Asimov* significance looks lengthy

$$Z_A = \left[ 2 \left( (s+b) \ln \left[ \frac{(s+b)(b+\sigma_b^2)}{b^2 + (s+b)\sigma_b^2} \right] - \frac{b^2}{\sigma_b^2} \ln \left[ 1 + \frac{\sigma_b^2 s}{b(b+\sigma_b^2)} \right] \right) \right]^{1/2}$$

- but simplifies without $\sigma_b$ to $Z_A(\sigma_b = 0) = \sqrt{2\left((s+b)\ln(1+s/b) - s\right)}$
- and in the limit of small, i.e. $s, \sigma_b \ll b$, to the more common $s/\sqrt{b+\sigma_b^2}$
- Here we go for small **b** and therefore use the **full expression** above which performs well for even a few events
- NB: $Z_A$ **does not scale** with Luminosity L

# Learning for Discovery

## Cross-entropy and Asimov significance

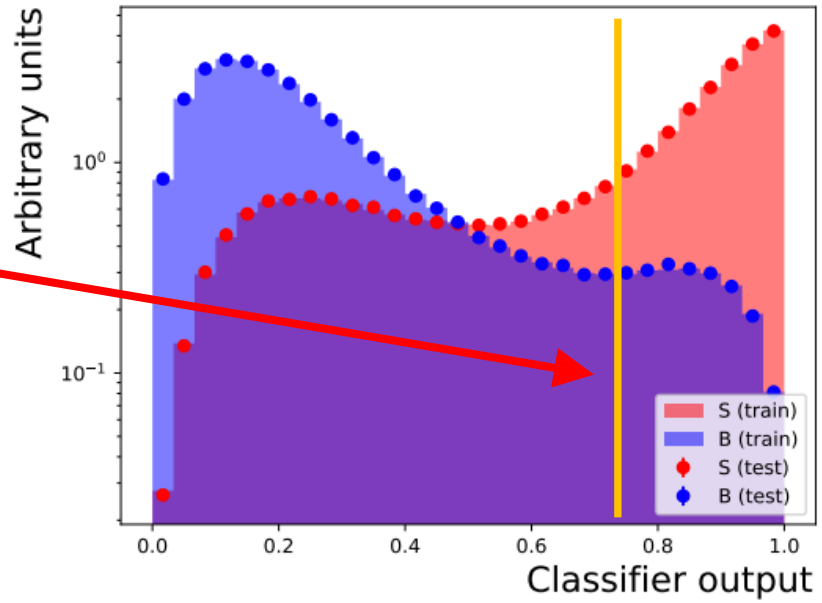
- Neural networks for binary classification are typically trained with cross-entropy
  - Cross-entropy requires the network to model the true label distribution
  - It optimizes accuracy: $\mathrm{Acc} = \frac{s_{true}+bt_{rue}}{all}$
- In searches we are not interested in high accuracy, i.e. correct labelling of all background and signal event but
  - We are looking for a phase space region with high signal purity for optimal **discovery significance**

**binary cross-entropy**

$$-\frac{1}{N}\sum_{i=1}^{N}\left[y_i^{true}\log(y_i^{pred}) + (1-y_i^{true})\log(1-y_i^{pred})\right]$$

true label

predicted label



# Learning for Discovery

## Cross-entropy and Asimov significance

- When optimizing a classifier a typical approach is to optimize the accuracy: $\mathrm{Acc} = \frac{s_{true}+btrue}{all}$
- For neural networks standard approach for training a binary classifier is the **cross-entropy**
- Accuracy maximizing is equivalent to minimizing the cross-entropy:

$$-\frac{1}{N}\sum_{i=1}^{N}\left[y_i^{true}\log(y_i^{pred}) + (1-y_i^{true})\log(1-y_i^{pred})\right]$$

- **Can we optimize directly for the Asimov significance, i.e. can we use it as a loss function ?**

# Learning for Discovery

## Cross-entropy and Asimov significance

- When optimizing a classifier a typical approach is to optimize the accuracy: $\mathrm{Acc} = \frac{s_{true}+btrue}{all}$
- For neural networks standard approach for training a binary classifier is the **cross-entropy**
- Accuracy maximizing is equivalent to minimizing the cross-entropy:

- **Can we optimize directly for the Asimov significance, i.e. can we use it as a loss function ?**

$$Z_A = \left[ 2 \left( (s+b) \ln \left[ \frac{(s+b)(b+\sigma_b^2)}{b^2 + (s+b)\sigma_b^2} \right] - \frac{b^2}{\sigma_b^2} \ln \left[ 1 + \frac{\sigma_b^2 s}{b(b+\sigma_b^2)} \right] \right) \right]^{1/2}$$

The sum is now inside

# Learning for Discovery

## Cross-entropy and Asimov significance

- When optimizing a classifier a typical approach is to optimize the accuracy: $\mathrm{Acc} = \frac{s_{true}+btrue}{all}$
- For neural networks standard approach for training a binary classifier is the **cross-entropy**
- Accuracy maximizing is equivalent to minimizing the cross-entropy:

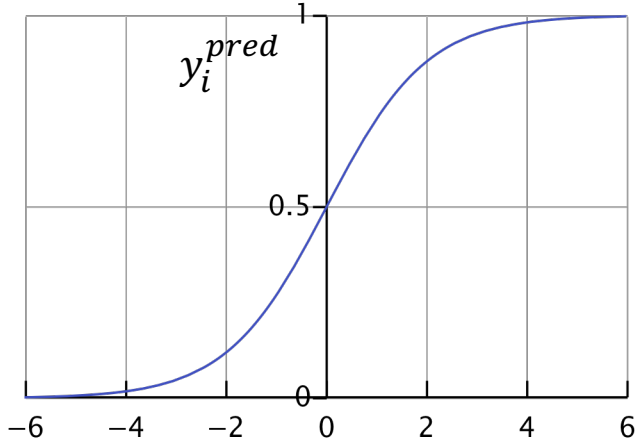
- **Can we optimize directly for the Asimov significance, i.e. can we use it as a loss function ?**
- **Caveat**: To define the number of signal and background events we need to cut on the discriminator output
  - Makes it non-differentiable ??
  - Differentiability is needed for gradient descent learning

$$Z_A = \left[2\left((s+b)\ln\left[\frac{(s+b)(b+\sigma_b^2)}{b^2+(s+b)\sigma_b^2}\right] - \frac{b^2}{\sigma_b^2}\ln\left[1+\frac{\sigma_b^2 s}{b(b+\sigma_b^2)}\right]\right)\right]^{1/2}$$

# Learning for Discovery - Main Idea

## Asimov significance as loss function

- A single sigmoid output neuron
- Replace the discrete number of signal and background events by a smooth function of the predicted label $y_i^{pred} \in [0,1]$




$$s = W_s \sum_i^{N_{batch}} y_i^{pred} \times y_i^{true}$$

$$b = W_b \sum_i^{N_{batch}} y_i^{pred} \times (1 - y_i^{true})$$

**$1/Z_A(s,b)$ becomes a smooth function of $y_i^{pred}$**

$W_s, W_b$ some weights to get the physical number of events depending on luminosity and efficiency to get the

# Learning for Discovery – applied to a toy SUSY search

## Asimov Loss function and Classifier Output - arXiv:1806.00322 [hep-ex]

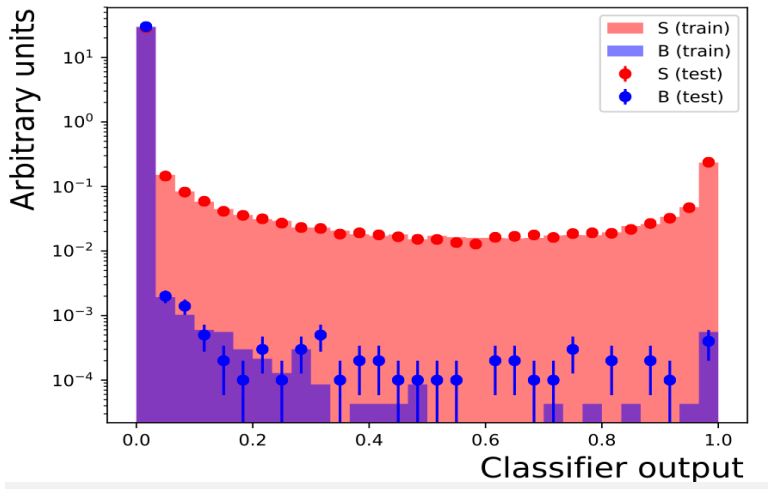
**Asimov loss training**

- best $Z_A = 6.2 \pm 0.6$
- **Acc = 59%**
- AUC=0.80
- Tries to find a background free region

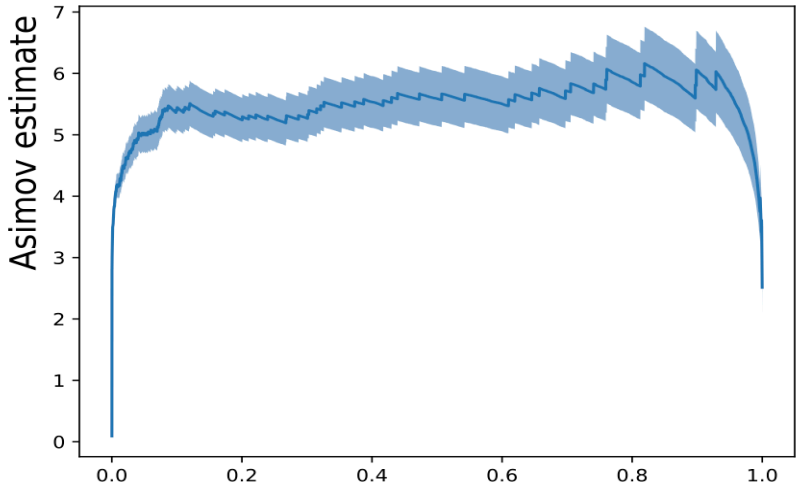
**Cross-entropy training + purity cut**

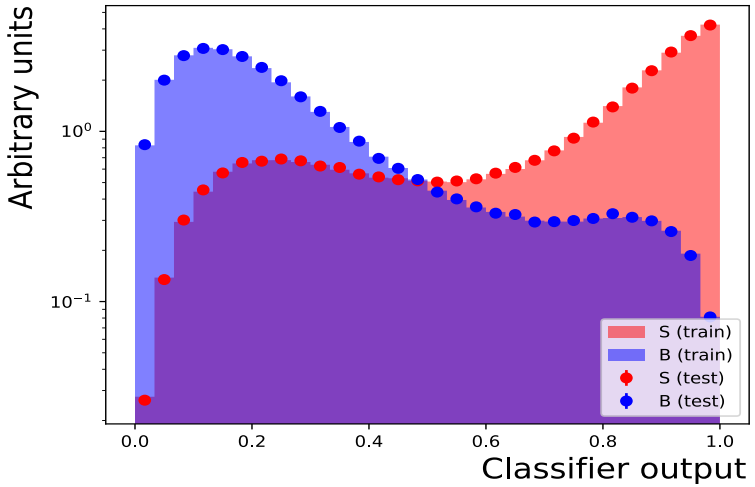
- best $Z_A = 4.8 \pm 0.3$
- **Acc = 92%**
- AUC=0.87

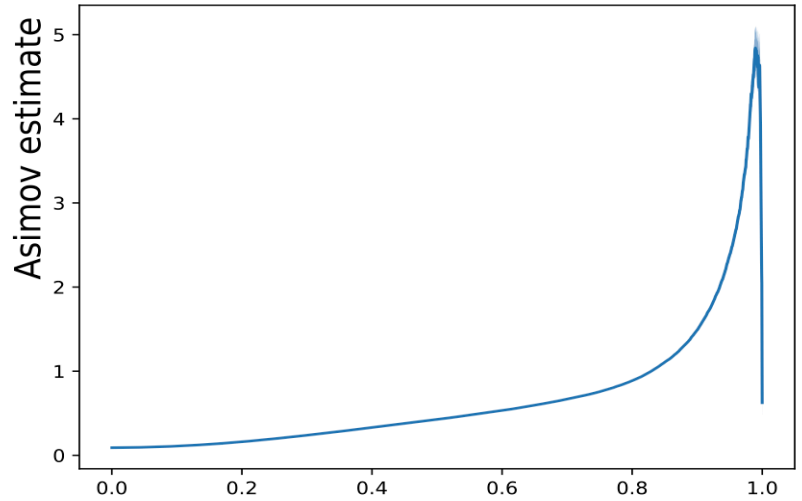
Systematic uncertainty 50%, Differences in $Z_A$ shrink for small systematic uncert.




Asimov score vs. cut on classifier




compressed model point

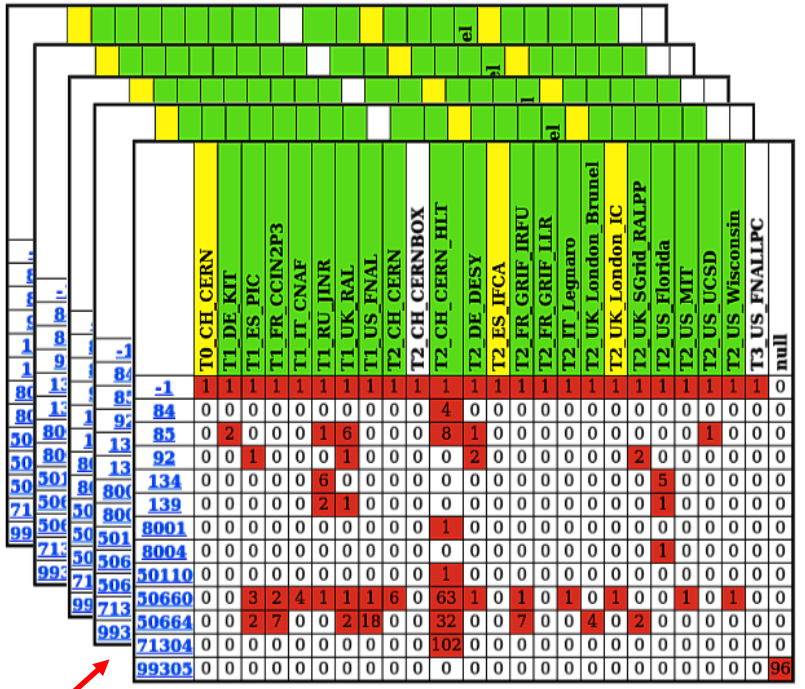
# Computing

# CMS Workflow Failures Recovery Panel Towards AI-assisted Operation

## Christian Contreras et al. CHEP 2018

### Motivation

- CMS central production system
  - ~200K grid cores, ~hundred sites
  - thousands of work flows with thousands of jobs
- Failures are unavoidable
  - Manual operator intervention
  - Can we have an automated system that give **advice how to recover a job**?
- Train a Deep Neural Network



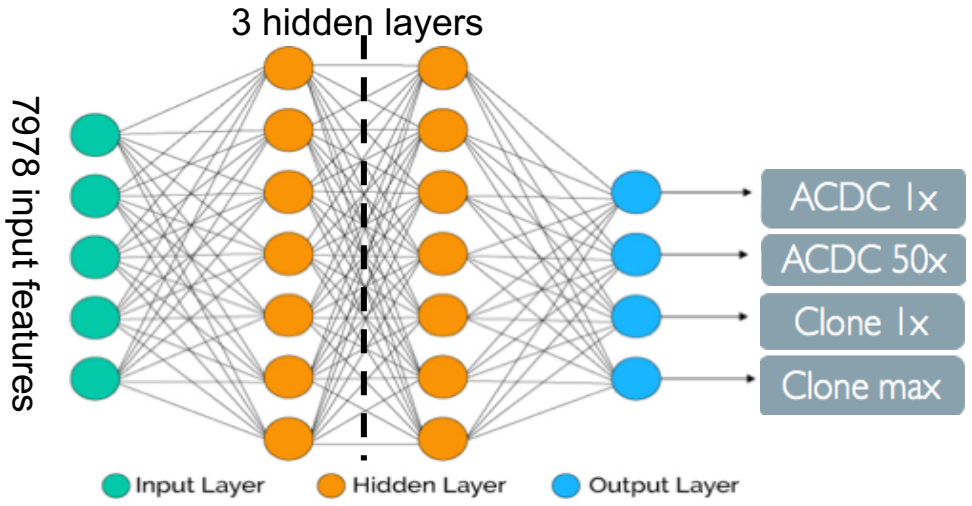

### Trainings data (15,000 workflow tasks)

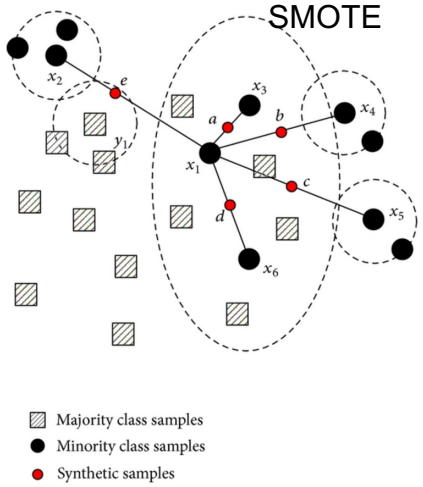
- Logs (json files) pulled from Workflow Team Web Tools which extracts information from Site-Readiness report
  - **Job failure codes and site information**
  - **Operator actions**

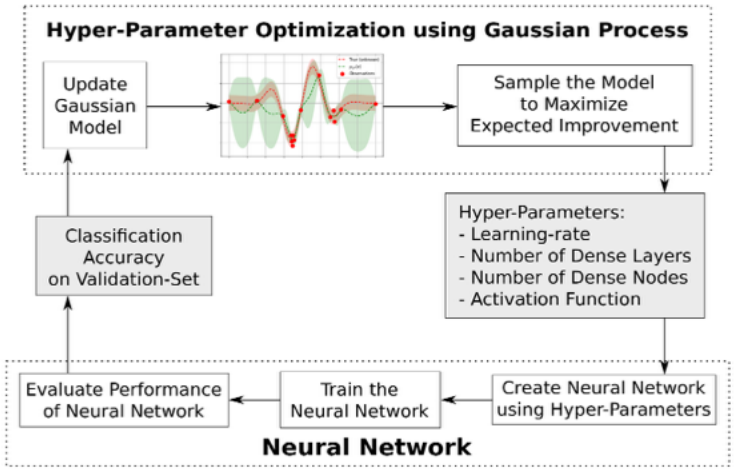
- ACDC (rerun failed)
  - splitted
- Clone (total retry)
  - splitted

# CMS Workflow Failures Recovery Panel Towards AI-assisted Operation

## Christian Contreras et al. CHEP 2018




### Difficulties to handle

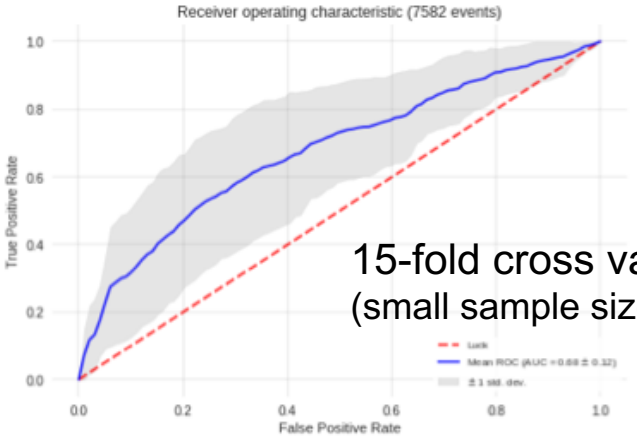
- Input preprocessing
  - Getting the data, how to handle missing data
- Unbalanced data
  - E.g. some failure codes are rare
  - **SMOTE** Synthetic Minority Over-Sampling by interpolation for minority classes
- Multiclass Training
  - 2 stage classifier
    - Binary + multi-class

### Model hyper-parameter tuning by Gaussian Processes (skopt)

- Eg. number of layers, numer of neurons, ..

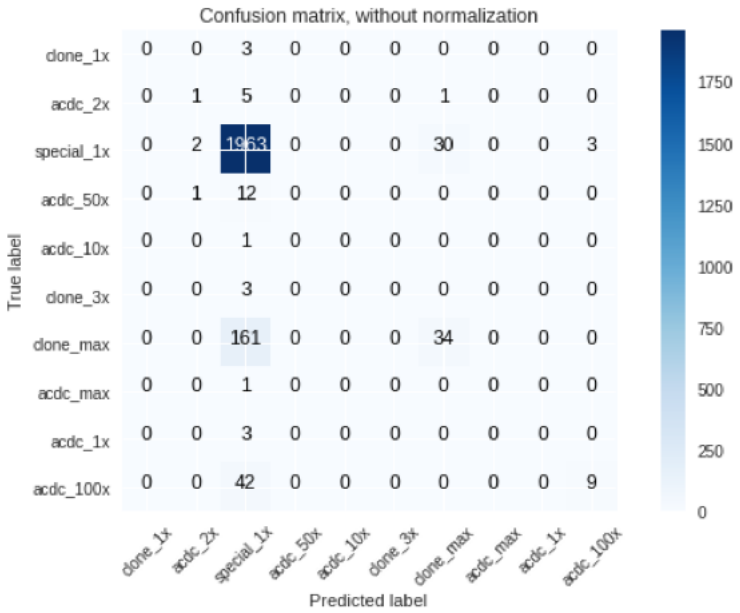
# CMS Workflow Failures Recovery Panel Towards AI-assisted Operation

## Christian Contreras et al. CHEP 2018




binary stage

15-fold cross validation on ROC
(small sample size)




Confusion matrix
for the multi-class stage
70/30% split for train/test

- A first pass for the supervised learning in error handling prediction. The operator's procedure will be automatized further by applying the decisions that are predicted with acceptable confidence.
- **Improve current WTC web interface**
  - To start using Machine Learning Model
    - Include the prediction for recommended action Start recovery from trivial cases
    - Monitor performance for model re-training
  - Add GUI display for diagnostic summary reports

| Model type | accuracy | Recall | Precision |
|---|---|---|---|
| **Binary** | 87% | 87% | 83% |
| **Multi-class** | 87% | 86% | 83% |

# Collecting data routinely

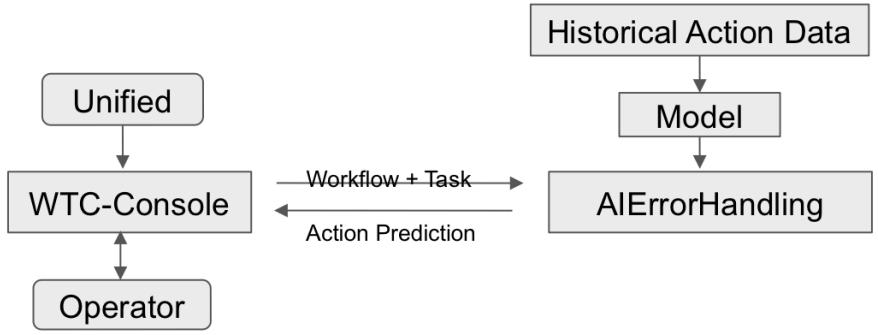
## Ongoing work to implement the tools in the daily workflow for further development

- Operator decisions are “Noisy labels”
- Collecting available data
  - Databases
  - Log files
- Model prediction as additional information to support the operator

# AIErrorHandling

- A machine is dedicated to serve as the “action predictor”
- Useful to keep track of several “trained models” during development phase
- Using workflow/task name, extra information are gathered and the “prediction” is produced

- WTC-Console Tech Stack:
  - React.js
  - Celery Workers
  - Django
    - MongoDB
- AIErrorHandling Tech Stack:
  - Tensorflow
  - Django JSON API
  - CMSCompOps
    - WorkflowWebTools
    - CMSToolBox




20

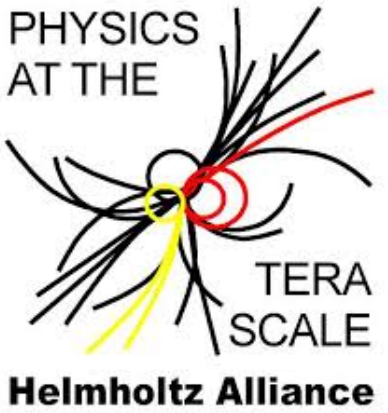
# Education

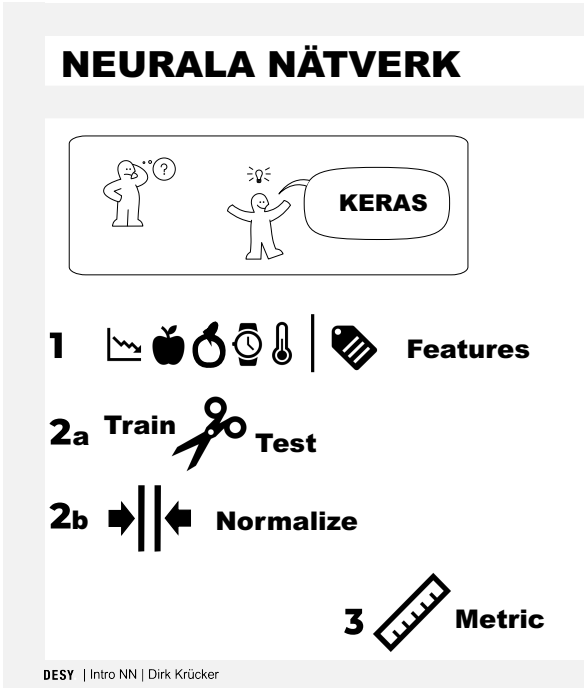
# Education

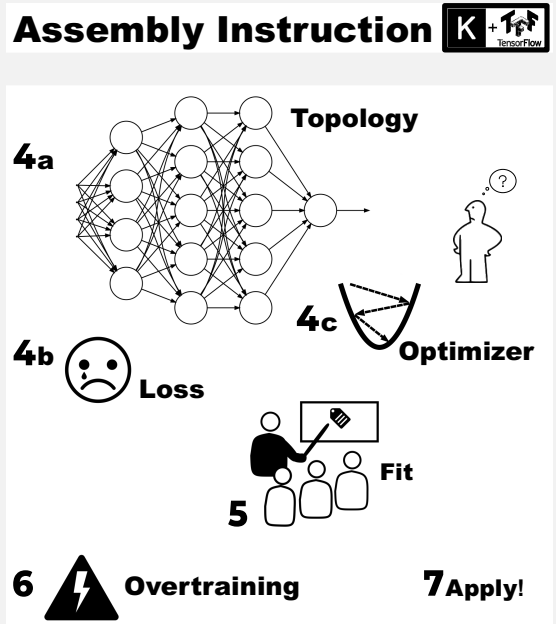
## DESY provides education in the field of Machine Learning

Schools on Machine Learning within the Terascale Alliance and beyond

- 5th Machine Learning in High Energy Physics in cooperation with the Yandex Data Science School, June 2019
- 10 days of lab courses in Wuhan/China, INFIERI summer school, May 2019
- 1st Terascale School on Machine Learning, October 2018



# Summary

## Many CMS ML activities in different areas