Incorporating Systematic Uncertainties in Supervised Classification: a Review

Tommaso Dorigo

INFN Padova

Abstract & info

I will discuss the impact of nuisance parameters on the effectiveness of supervised classification in high-energy physics problems, and techniques that may mitigate or remove their effect in the search for optimal selection criteria and variable transformations.

The approaches discussed include nuisance-parametrized models, modified or adversary losses, semi-supervised learning approaches and inference-aware techniques.

- Please interrupt if you feel you need more detail or if something is unclear
- Slides with a (*) in the title will be likely omitted due to time constraints; they are left for reference
- References are marked in green [xx] and listed at the end of this document

Introduction

Systematic uncertainties affect the precision of measurements in HEP (==high-energy particle physics). With Machine Learning we may reduce their impact.

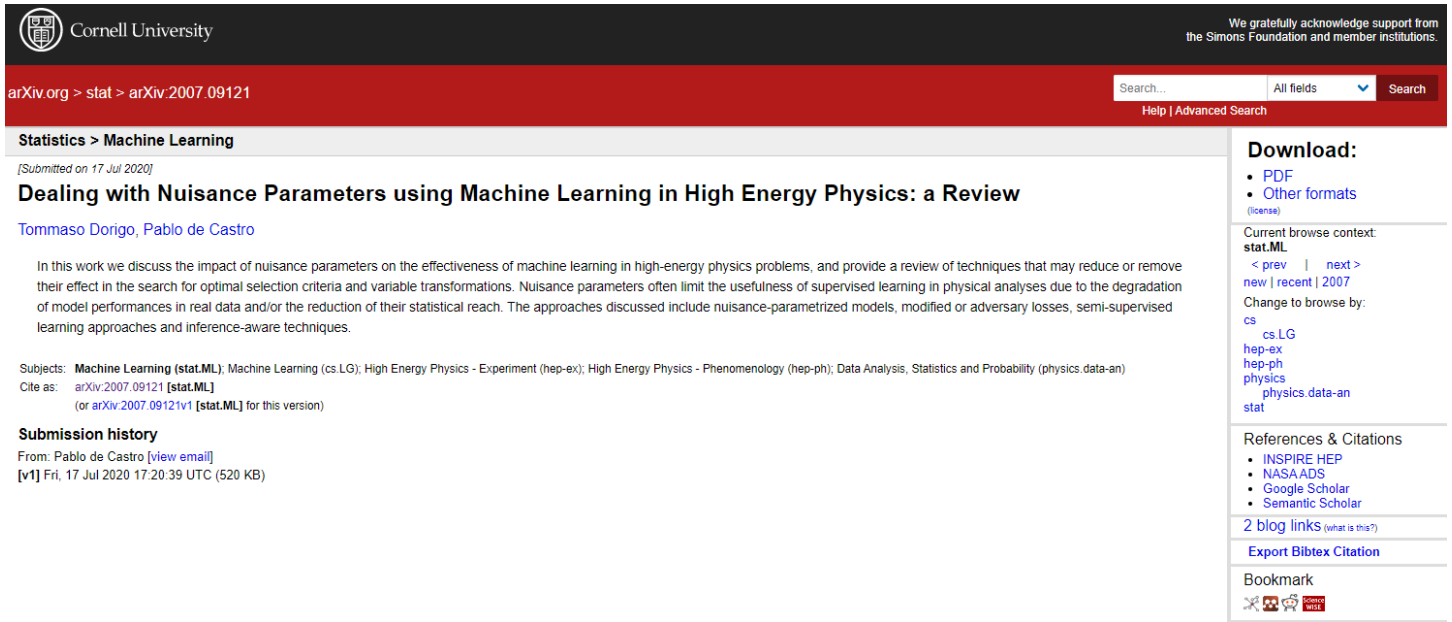
We will focus today on supervised classification, which is by far the most common use case

- Much of the discussion also applies to supervised regression

The contents match well with those of Chapter 7.2 of a book on ML for HEP, which will be published by World Scientific early in 2021.

A preprint of that chapter, titled "*Dealing with Nuisance Parameters Using Machine Learning in HEP Analysis – A Review*" and authored by Pablo de Castro Manzano and myself is available at **https://arxiv.org/abs/2007.09121** **[59]**

- Credits to Pablo de Castro Manzano for part of the material



Contents

0. Particle physics in 7 slides

We are going to discuss how ML can help handling systematic uncertainties, using particle physics problems as benchmarks

→ I need first to explain the general framework of these problems

- I claim I will say all you need to know about this before you manage to fall asleep

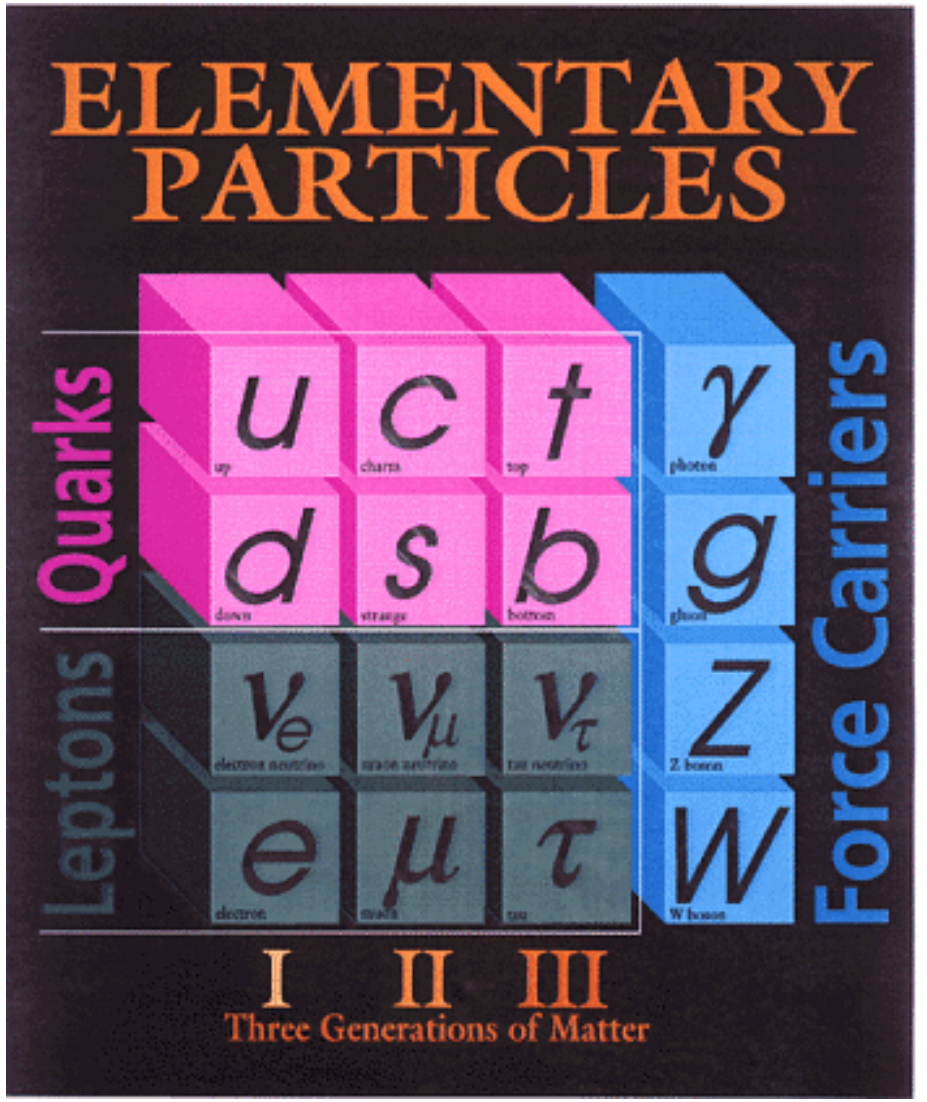
"Particles, particles, particles."

The Standard Model

A misnomer – it is not a model but a full-blown theory which allows us to compute the result of subatomic processes with high precision

- Three families of **quarks**, and three families of **leptons**, are the matter constituents
- Strong interactions between quarks are mediated by 8 **gluons**, g
- Electromagnetic interactions between charged particles are mediated by the photon, **γ**
- The weak force is mediated by **W and Z** bosons

The **Higgs boson** is an additional peculiar particle that gives mathematical consistency to the whole construction

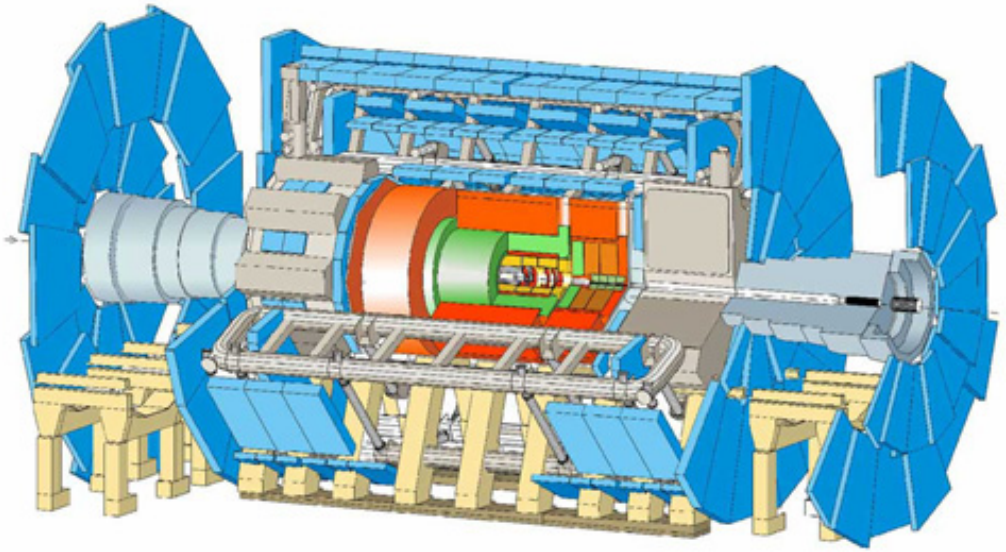


The LHC

LHC is the **world's largest and most powerful** particle accelerator, built to investigate matter at the shortest length scales

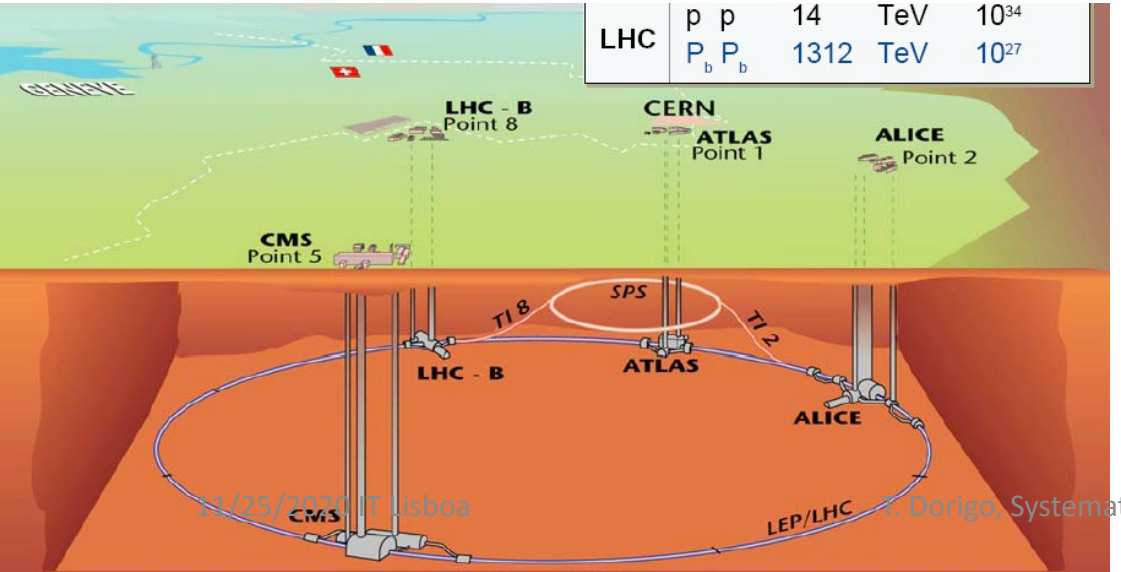
It resides in a 27km long tunnel 100 meters underground near Geneva

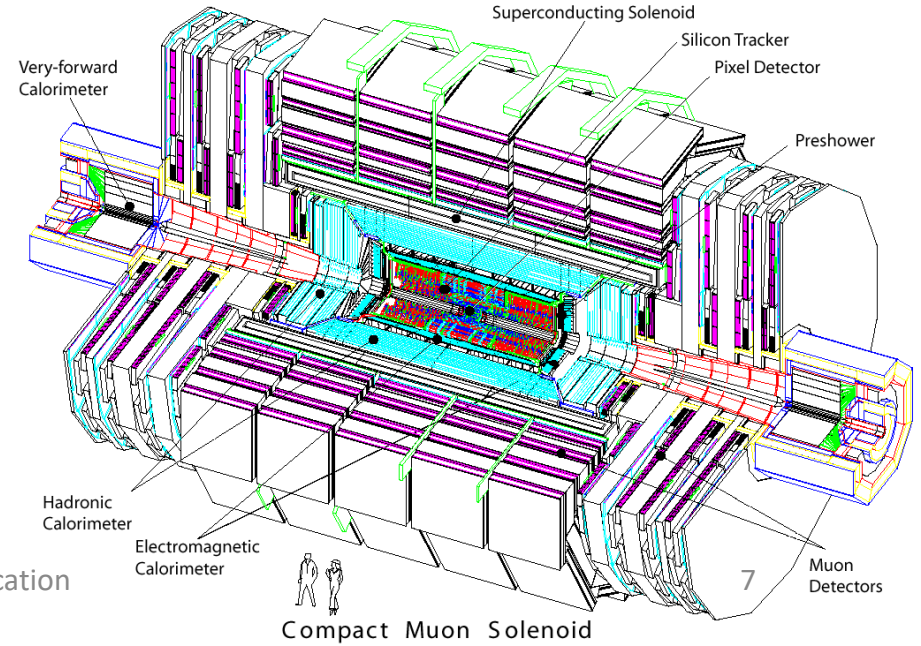
Collisions between protons are created where the beams intersect: 4 caverns are equipped with huge detectors. Two of these (ATLAS and CMS) are multi-purpose «electronic eyes» that try to detect everything that comes out of the collision



ATLAS

CMS

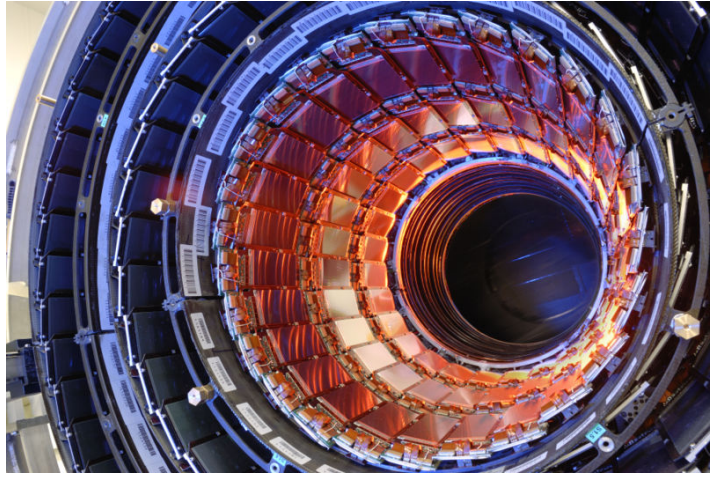


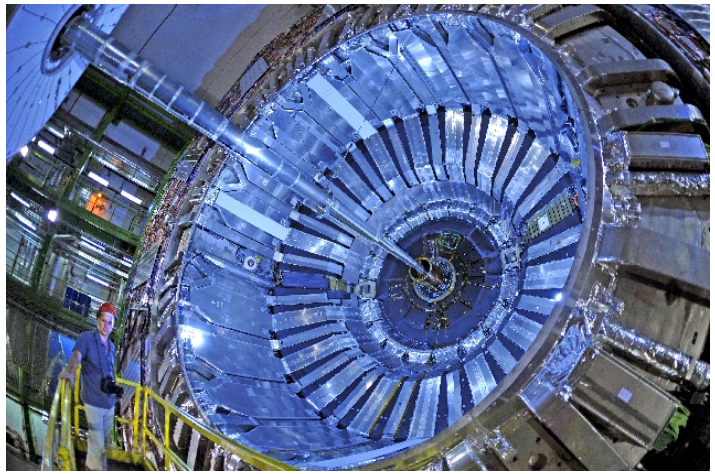


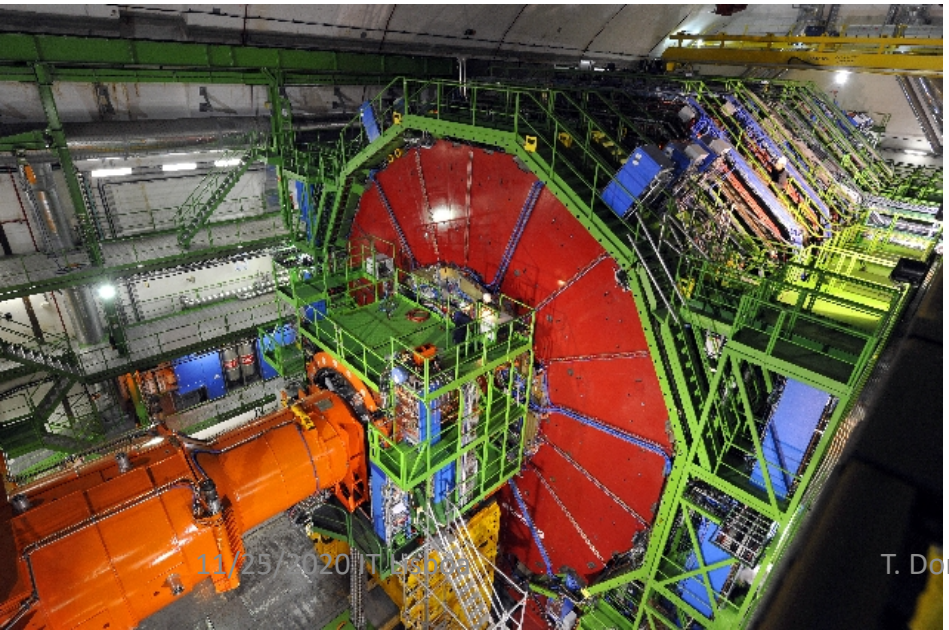
CMS

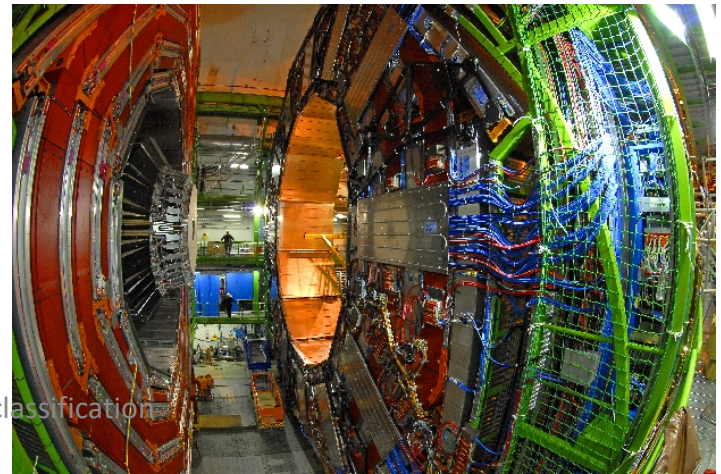
CMS (Compact Muon Solenoid) was built with the specific goal of finding the Higgs boson

- Along with ATLAS, it is arguably one of the most complex machines ever built by mankind
- Hundreds of millions collisions take place every second in its core, and each produces signals in tens of millions of electronic channels. These data are read out in real time and stored for offline analysis









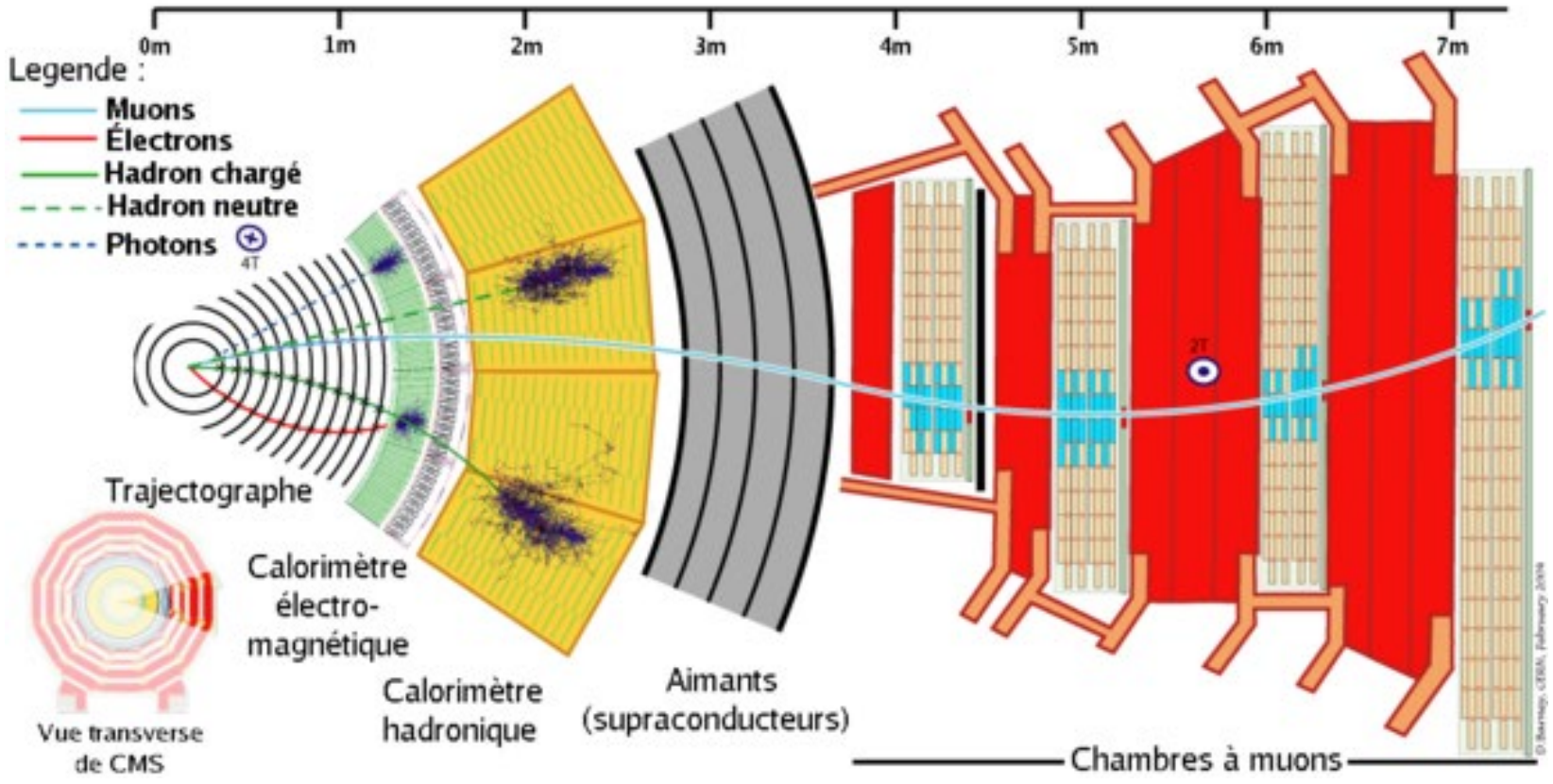
How we detect particles

Charged particles are detected in the **tracker**, through the ionization they leave in silicon;

a powerful **magnet** bends their trajectories, allowing a measurement of their momentum

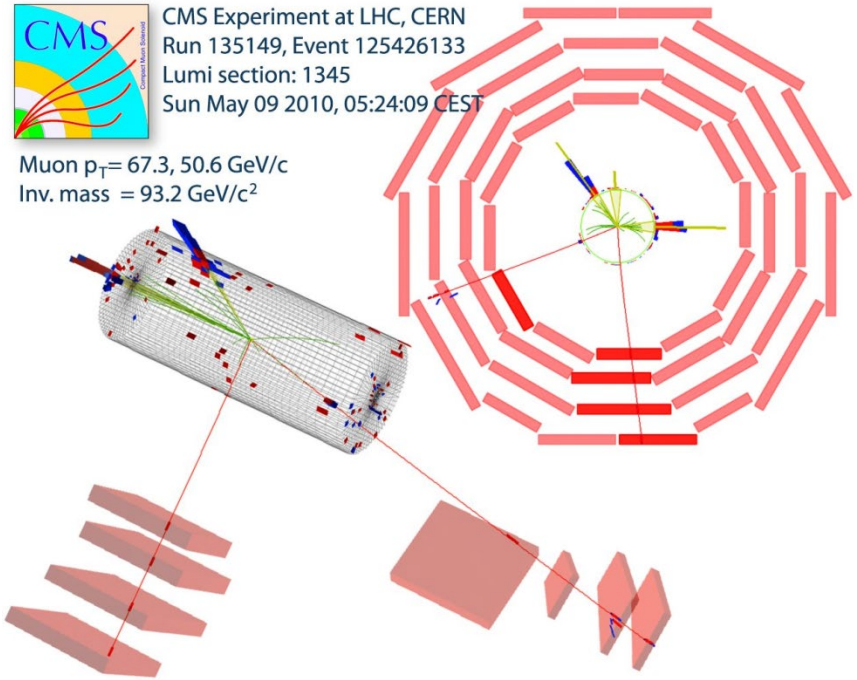
Then **calorimeters** destroy both charged and neutral ones, measuring their energy

Muons are the only particles that can traverse the dense material and get tracked outside

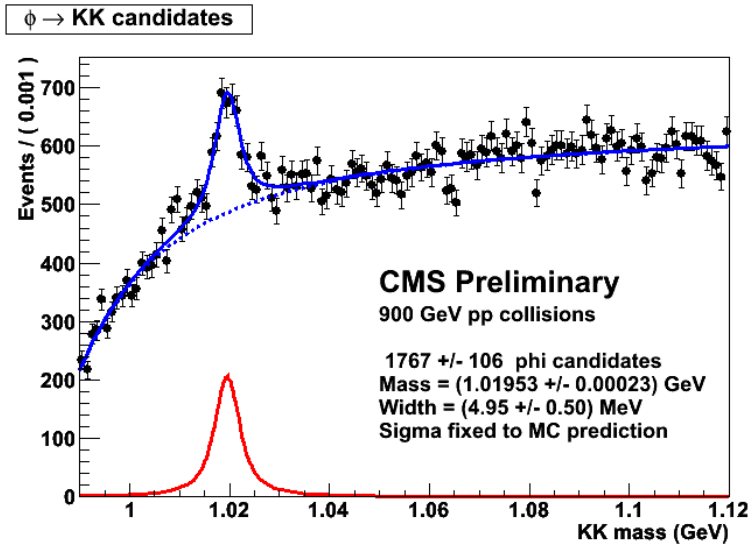


How we see a collision

A reconstruction of the O(10M) electronic signals provides us a «view» of the created objects: using their characteristics we build O(100) high-level variables which we compare to theoretical models after a further compression (usually into a 1-dim test statistic) → then we do measurements and inference



This is a huge dimensionality reduction…



What we do with it

We have a theory that allows us to calculate predicted probabilities for the possible physics processes, to extreme accuracy– but **we believe it is incomplete and to some extent unsatisfactory**.

So we look for new physics processes: things that the Standard Model does not include

- New matter particles
- New force carriers

We also measure with precision known processes, in the attempt of finding a significant difference with model calculations

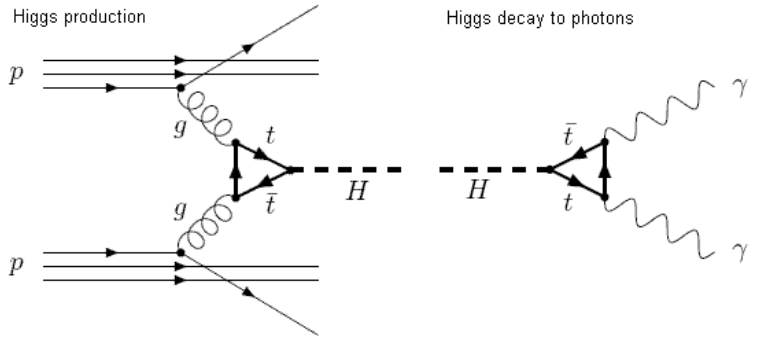
- We thus make extensive use of
  - Hypothesis testing
  - Point and interval estimation

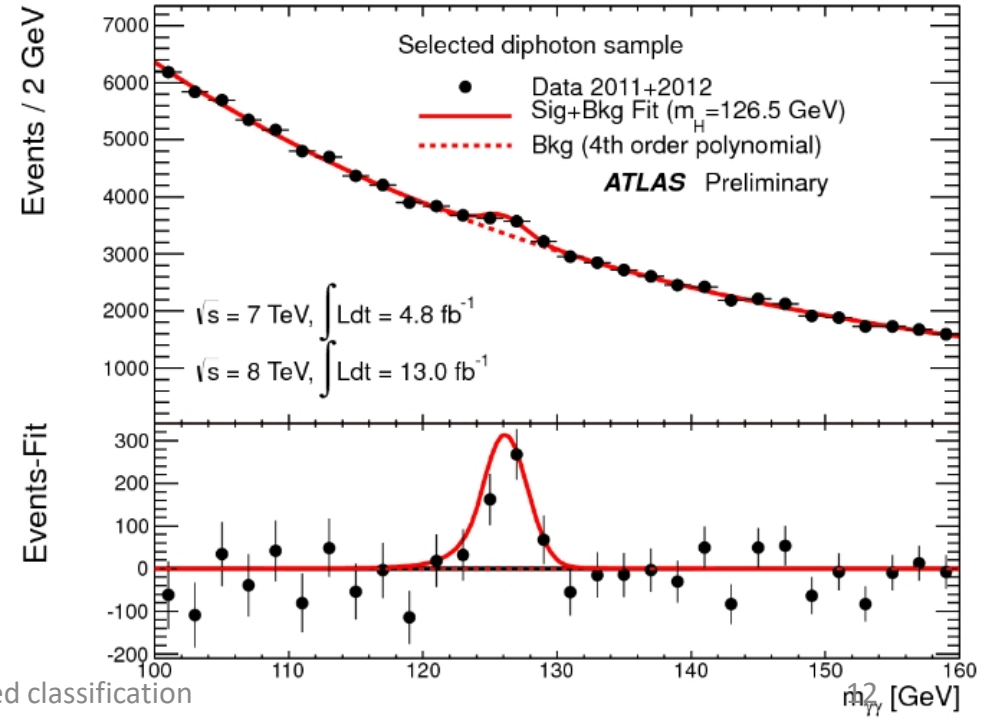
Example: new particle searches

The typical search for a new particle involves a model which predicts it

- Monte Carlo generators use the model to produce **simulated datasets** that **teach us how the signal looks like**

- A data selection isolates a sample where we try to evidence the particle
- Typically we attempt to reconstruct the **particle mass** from the measured features. As mass is a unique attribute of the particle, a histogram may then display a narrow bump on a smooth background
- A test of hypotheses allows to derive **$p(\text{data}|H_0)$**
  - More on that later





1. Problem statement

Supervised classification is used to construct low-dimensional event summaries: summary statistics

- Summary statistics can be employed to carry out statistical inference on parameters of interest $\boldsymbol{\theta}$

- *E.g.* we may use a NN to reduce features $\mathbf{y}$ into a single-dimensional output x, which according to our model distributes with a PDF $f(x|\boldsymbol{\theta})$

The implied compression is informed by simulated observations produced by a generative model (MC). The fidelity of the latter is limited by

- Imperfections in the model (*e.g.* "NLO accuracy")
- Imprecise simulation of detector (calibration constants, etc.)
- Uncertainty in fundamental parameters (top mass?)
- Finiteness of available data samples

- The above are referred to as "nuisance parameters"

Nuisance parameters

To account for the above imperfections, described by nuisance parameters $\boldsymbol{\alpha}$, we would need to enlarge our model to $p(x|\boldsymbol{\theta},\boldsymbol{\alpha})$. The latter can then be used in the construction of a likelihood function, a surrogate of L, or whatever other estimator we need.

- This allows us to account for the variability of the nuisances in our inference
- The inclusion of nuisances usually enlarges the resulting confidence intervals on θ
- A similar effect occurs if we use the model in hypothesis testing → power reduction

→ The presence of nuisance parameters limits the precision and the discovery reach in HEP analyses

→ The problem is however much more general and applies to any inference procedure ("*all models are wrong*")

2. Nuisance parameters in statistical inference

It is useful to recall how nuisance parameters may be "profiled away" from a likelihood function in the extraction of confidence intervals

- In statistical parlance, our measurements constitute a problem of parameter estimation, whose solution is provided by specifying a statistical model. In the model, nuisance parameters may be free and their PDF may be unknown.

We solve the measurement problem by constructing estimators through the likelihood function. Let

- x_i, i=1…N be our observations: random i.i.d. variables
- θ be the parameters of interest
- α be the nuisance parameters

We may write the joint PDF as p(x,θ,α) and with it the likelihood,

$$\mathcal{L}(\theta,\alpha) = \prod_{i=1}^{N} p(x_i,\theta,\alpha)$$

Profiling and marginalizing

If there are no nuisance parameters, the estimation problem is solved by constructing estimators as

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\theta)$$

When nuisances α are present, the profile-likelihood method consists in first obtaining the profile PL(θ) by maximizing over nuisances,

$$PL(\theta) = sup_{\alpha}\mathcal{L}(\theta, \alpha)$$

and then proceeding as above[5][6]. MIGRAD[7] can do this for you, as other more recent packages. However, this assumes that L be differentiable, and can become an impractical solution for large-dimensionality of parameters.

Similar issues affect the main alternative, a Bayesian solution, which marginalizes by integration in the nuisance space over the nuisance prior p(α):

$$\mathcal{L}_m(\theta) = \int \mathcal{L}(\theta, \alpha)p(\alpha)d\alpha$$

Of course, knowledge (or assumption) of the PDF p(α) is necessary, and that is not always given. But even then, nuisance parameters affect the inference by widening our confidence intervals!

3. Toward fully sufficient statistic summaries

ML applications usually focus on the goal of minimizing the *statistical* uncertainty on the estimates of parameters of interest θ.

- The summary statistic they provide should enable the extraction of the highest amount of information on θ, *conditional to* the validity of the underlying model used to generate the samples, as well as of the assumptions made on the value of nuisance parameters α.
- *The conditionality above is hard to get rid of*! as, *e.g.*
  - Problems are complex and high-dimensional
  - Nuisance parameters have unknown PDF
  - Effect of nuisances on the default model is not easy to parametrize

The above imply that the summary statistic is not usually sufficient: being oblivious of a part of feature space, it does not retain all the information relevant to the parameter estimation task – it can be outperformed.

Statistical sufficiency: Fisher-Neyman

An all-important concept in statistical inference!

Fisher-Neyman factorization criterion: a summary statistic for a set of n i.i.d. observations $D=\{x_i, i=1...n\}$ is sufficient WRT a statistical model and a set of parameters θ iff the generating probability density function of the data, $p(x|\theta)$, can be factorized as

$$p(x|\theta) = q(x)\ r[s(x)|\theta]$$

where q is a non-negative function not dependent on θ, and r[] is another non-negative function for which dependence on x occurs only through the summary statistic s(x).

s() then contains all the information provided by D needed to estimate model parameters θ, and no other statistic may add any information from D.

→note: x itself is a sufficient statistic – but it is not a meaningful summary! (it does not reduce the dimensionality).

If we do not know p(x) in closed form, we cannot solve the problem analytically! However, in 2-mixture models where the signal fraction is the only parameter, the density ratio $\mathbf{s(x)=p_s(x)/p_b(x)}$ is a sufficient summary. Hence the advantage of probabilistic classification to approximate density ratios.

Optimize at your own risk

How many of you never chanced to read the word “optimize” declined somehow in a physics article?

To those who raised their hand: you have not done enough reading as of late. The word is used quite liberally, usually in connection with incremental advances of the employed analysis technique

Typically, evidence in support of an optimization task is offered in the form of a peak of the AUC (*area under the curve*) - the integral of the Receiver Output Characteristic (ROC); or on signal acceptance at fixed purity, *e.g.*

→ Those named above are reasonably good proxies to the measurement precision: their maxima approximately track the minima of the statistical uncertainty on intermediate parameters of interest, such as signal fraction….

Yet they are blind to the more general, ultimate goal of, *e.g.*, extracting the cross section of the signal, once all non-stochastic uncertainties are included

One trivial example

In order to be all on the same page, let us consider a fully analytic, trivial toy example of classification task.

Let

$$S(x) = \frac{e^x}{e-1}$$

$$B(x) = \frac{\alpha e^{-\alpha x}}{1-e^{-\alpha}}$$

be the output of a classifier on events belonging to class S (y=1) and B (y=0), where we have normalized S and B in [0,1] for ease of treatment. The background distribution depends on a nuisance parameter, α. Note that by writing B(x) as above, we implicitly assume we know that dependence perfectly.

Let our task in this toy problem be to estimate the signal fraction in data sampled from S and B, based on counting the fraction passing a selection on the output of a classifier trained to distinguish S from B.

TPR, FPR, ROC, AUC

The “true positive rate” (TPR) and the “false positive rate” (FPR) of a data selection criterion x>x* based on the classifier output x can be defined using the S(x) and B(x) PDFs as

$$TPR(x^*) = \int_{x^*}^{1} S(x)dx = \frac{e - e^{x^*}}{e-1},$$

“What are the odds that data with x>x* are signal?”

$$FPR(x^*) = \int_{x^*}^{1} B(x)dx = \frac{e^{-\alpha x^*} - e^{-\alpha}}{1-e^{-\alpha}},$$

“What are the odds that data with x>x* are background?”

and from them we may derive an expression for the **ROC curve**, defined as the functional dependence of TPR on FPR:

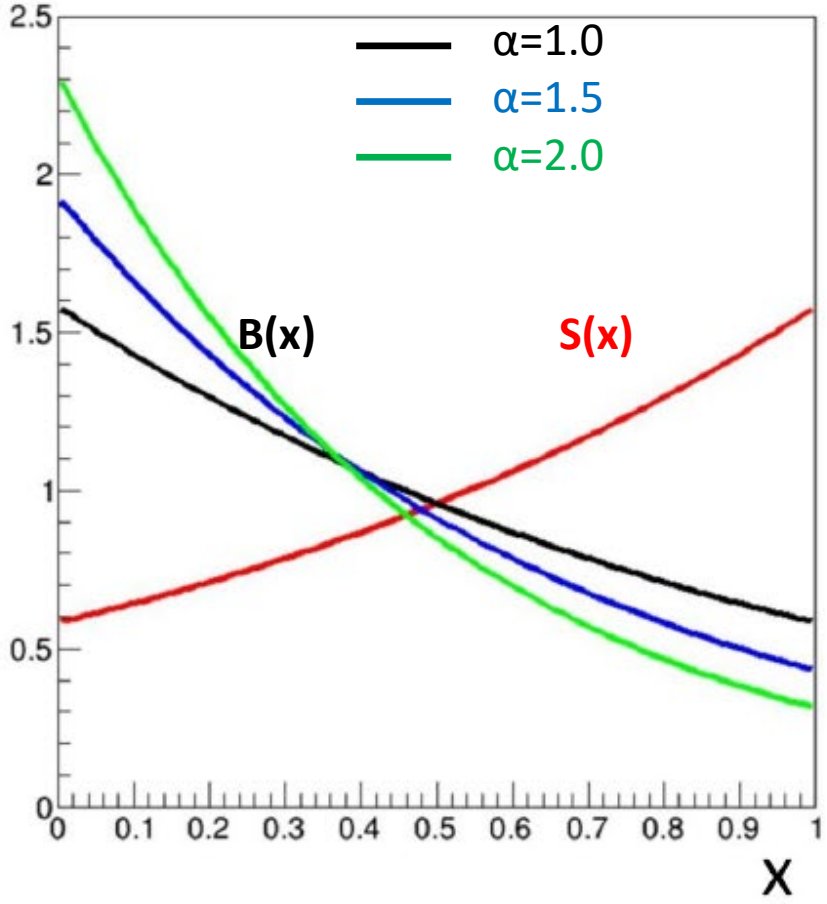
$$TPR(FPR) = \frac{e - [e^{-\alpha} + (1 + e^{-\alpha})FPR]^{-\frac{1}{\alpha}}}{e - 1}$$

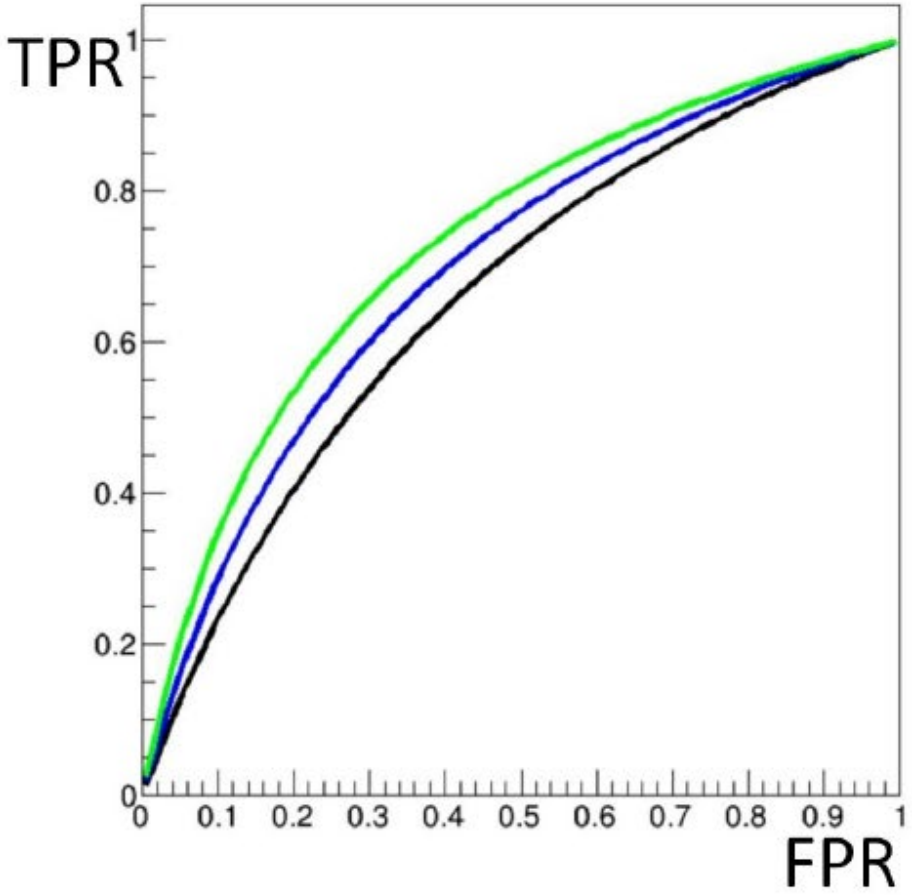
The AUC is then the integral of TPR(FPR) in [0,1].

Systematics at work

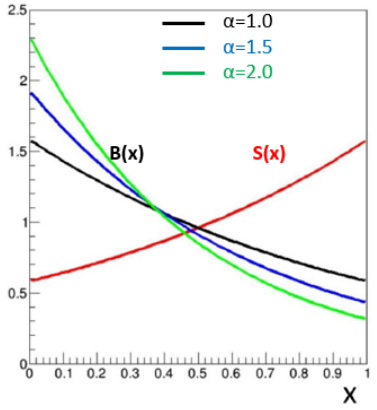
Our toy model allows a visualization of the effect of a nuisance parameter α on our figures of merit.

Again, we have assumed we know the analytic form of B(x|α)…





So what?



Of course, the TPR increases with α for fixed FPR, and so do the ROC and its integral: B(x) becomes steeper at small x, and the signal is more distinguishable.

The first real take-away bit is that if we train a classifier with a given value of α (*e.g.* 1.5 for the blue B(x) curve), the performance is going to be under- or over-estimated if the true value of α is different; the choice of a critical region x>x* corresponding to a pre-defined FPR will similarly be affected, as will the TPR be.

Now, recall that the fraction of data selected in the critical region is our **summary statistic** – our only input to the extraction of the signal fraction. That number is affected by α, but its value alone does not allow us to extract the full information on the true signal fraction: it is not a sufficient statistic. The whole distribution would be one such statistic, but it would not summarize our data well enough (in terms of dimensional reduction).

Taking a decision: enter the AMS

While we may handwavingly say that the higher our ROC (or its AUC), the better, we must define a prescription to decide on the critical region, i.e. the value of x* (or a given TPR value).

In order to have grounds to claim we are optimizing x*, we may try to maximize a figure of merit called "*approximate median significance*" (AMS):

$$AMS = \sqrt{2\left[(N_s + N_b + N_r)\ln\left(1 + \frac{N_s}{N_b + N_r}\right) - N_s\right]}$$

The AMS is a robust surrogate of the significance of an excess of observed events if a signal of mean N_s contributes to a dataset assumed to only contain background sampled from a Poisson of mean N_b. N_r is a regularization avoiding low-count divergences; N_r=10 is a sensible choice.

What happens to our toy problem ? Let us e.g. consider N_s=20, N_b=400 and see what happens.

Where is the AMS maximum?

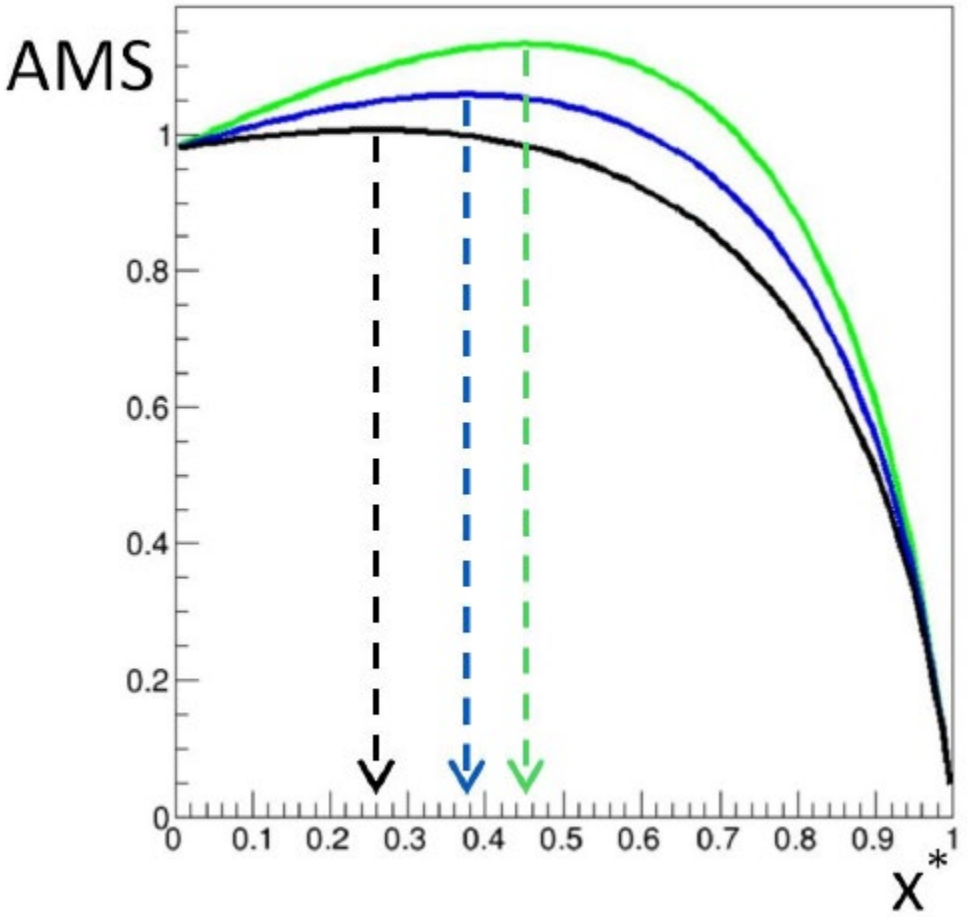
The AMS computed for the three exemplary values of α is shown on the right. As we fully expected, the values reached when α is larger are higher.

However, if we do not know what α is, we cannot "optimize" our critical region, as the optimal choice of x* strongly depends on the value of α, which we do not know.

Nuisance parameters affect the optimal working point, as well the performance of the classifier and the relative merits of different classifiers (which produce different summaries x)

→ Standard supervised classification techniques may not reach optimality unless they address the conditionality issue discussed *supra*.

Note: there is often a misalignment between the specification of the classification task and the true objective of an analysis. E.g., maximizing the AMS (even with no nuisances) may not be the correct way to optimize for an upper limit on N_s



4. Nuisance-Parametrized models

A straightforward attempt at accounting for nuisance parameters is to parametrize their effect on the observable features

→ this requires injecting a priori *knowledge* of their PDF

In low-dimensional cases, a fully analytical solution may be sought, when the parametrization of the nuisance allows to "decorrelate" its effect on the salient features of the events.

An example was proposed in a study of the n-subjettiness ratio τ_{12} [9]

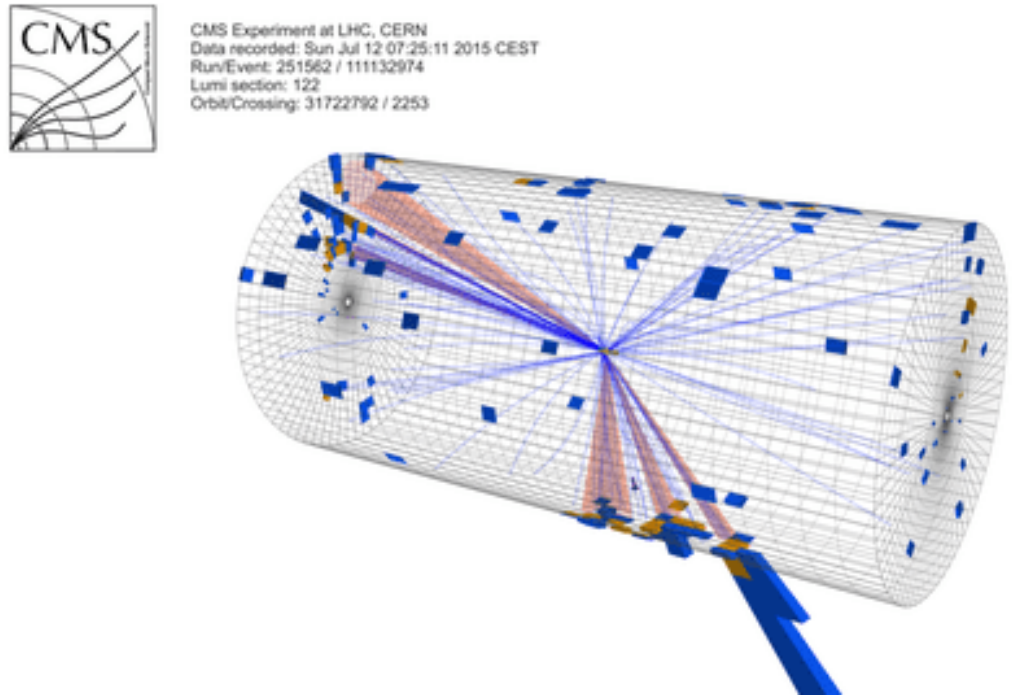
→ see next slide

(Background: boosted decays and fat jets)

Energetic LHC collisions may produce heavy objects with large momentum (top quarks, or W, Z, H bosons). When these decays, they usually yield a collimated stream of particles – a single hadron jet.

A number of techniques allow the extraction of features sensitive to the heavy object decay

N-subjettiness τ_n, ratios τ_n/τ_{n-1}, and soft-drop mass M_{sd} are some of the tools HEP analysts use to distinguish heavy resonances from QCD jets.



Above: a top-pair decay produces two fat jets, where the individual subjects are visible

Example: N-subjettiness in boosted decays

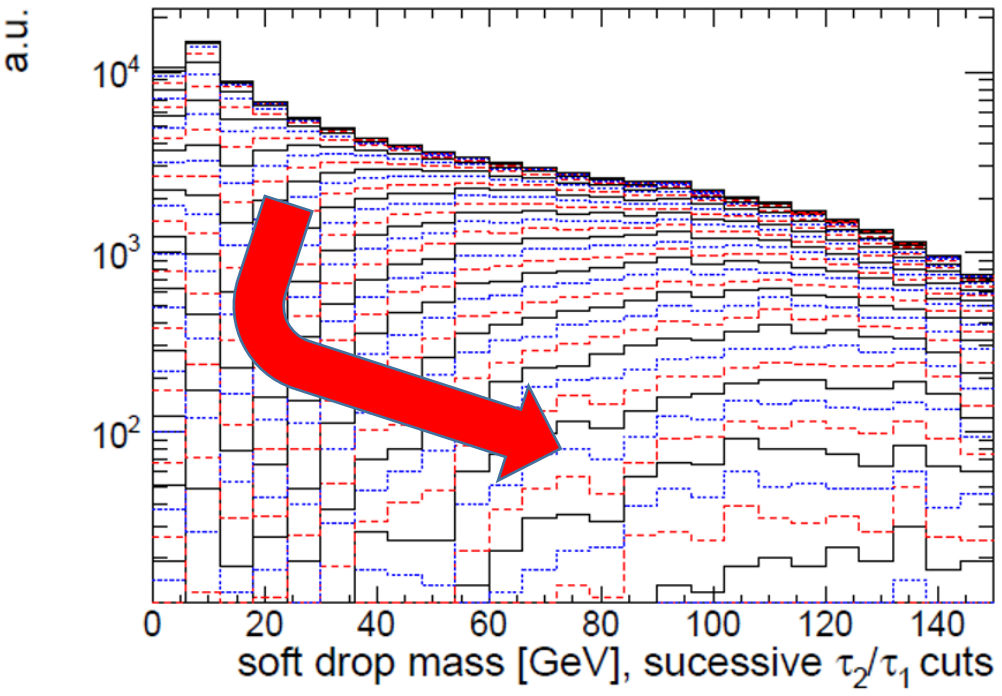
A number of observable features of fat energetic jets have been constructed to separate the hadronic 2- and 3-prong decay of heavy objects (W,Z,H,t) from QCD jets.

Useful variable to discriminate two-body decays: $\tau_{21}=\tau_2/\tau_1$, where taus are functions of the energy distribution within subjets

The problem is that the "soft-drop" mass M that can be constructed with the two subjets is correlated with τ_{21}: a cut on the latter increases S/B but distorts the distribution of M, because of the mutual dependence on jet p_T.

In statistical parlance we may consider p_T a nuisance parameter – it reshapes the variable we want to use for inference.

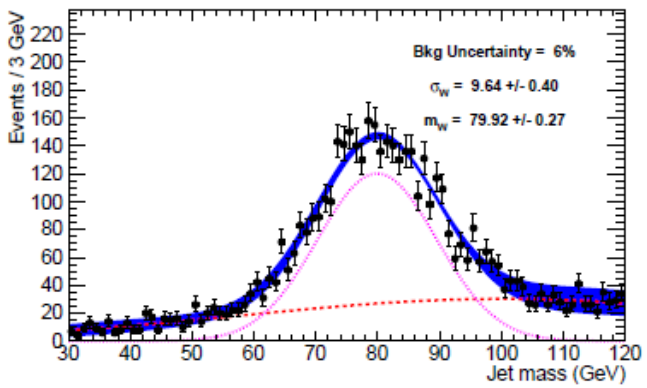
Dolen *et al.*[10] use an analytical parametrization of the nuisance to decorrelate its effect in the variable of interest

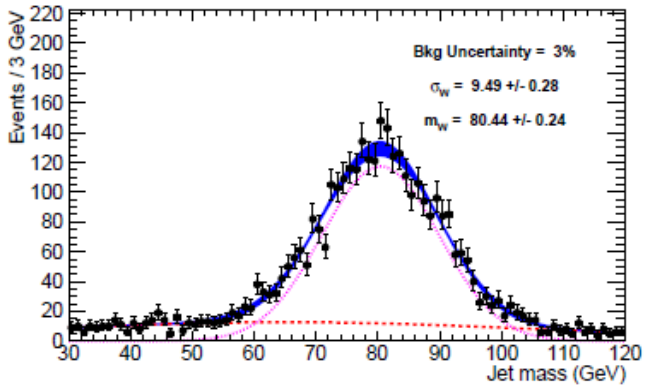


Correcting for the nuisance

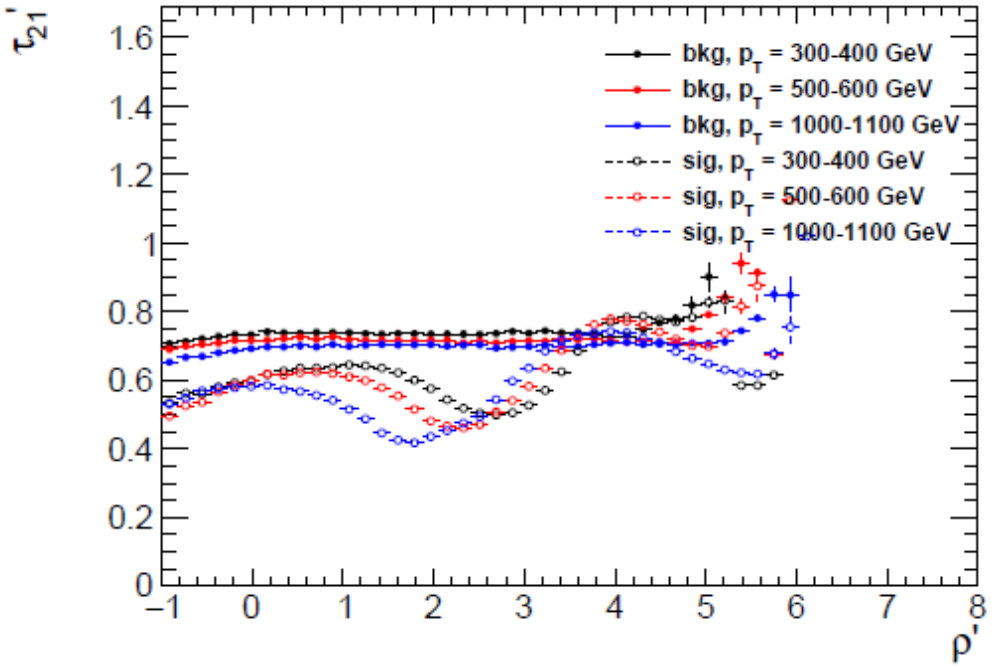
- Define $\rho = \log(m^2/p_T^2)$, an appropriate scaling variable for QCD jets
- Compute average of τ_{21} as f(ρ): it shows linear behaviour
- Define $\rho' = \rho + \log(p_T/\text{GeV})$; then define $\tau'_{21} = \tau_2/\tau_1 - M\rho'$
- Observe that new variable has flat behaviour for QCD

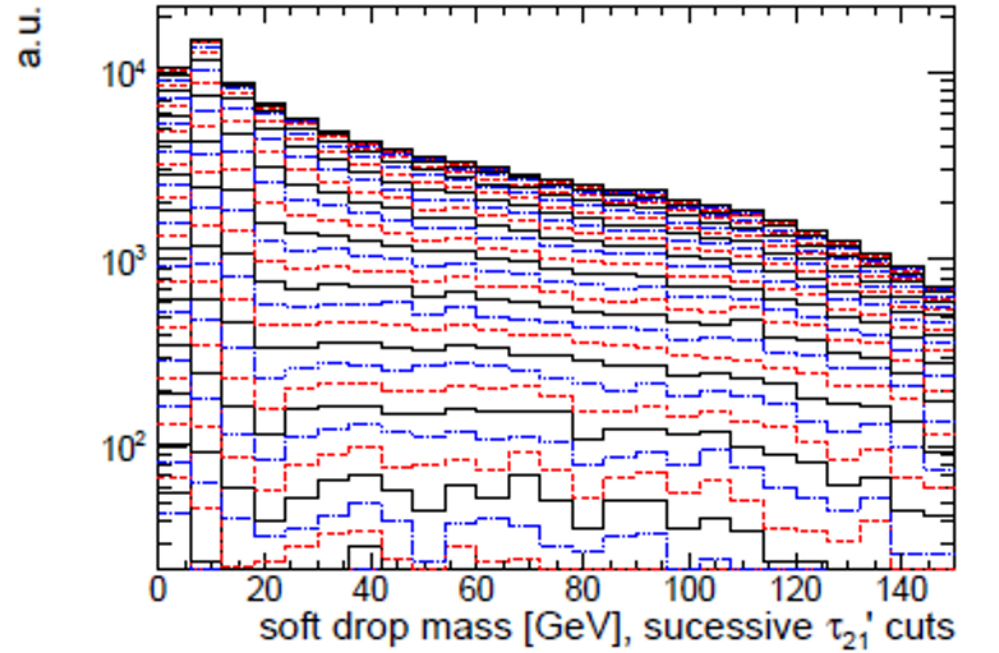
→ τ'_{21} decorrelates the p_T dependence on mass, allowing a selection that preserves ability to use sidebands, etc.





Same signal efficiency, better behaviour for decorrelated tagger (right)





Looking into the matter

In several cases of HEP interest, the data contain information on the nuisance parameter, and one may then try to exploit it to construct robust estimators.

Consider the problem of extracting a sufficient summary statistic for the signal fraction θ with a binary classifier, in presence of a nuisance α that modifies the $P_s(x,\alpha)$ and $P_b(x,\alpha)$ density functions. The likelihood

$$L(\theta,\alpha) = \prod_{i=1}^{N} [\theta p_s(x_i,\alpha) + (1-\theta) p_b(x_i,\alpha)]$$

may be rewritten as

$$L(\theta,\alpha) = \left[\prod_{i=1}^{N} p_b(x_i,\alpha)\right] \cdot \prod_{i=1}^{N} \left[\theta \frac{p_s(x_i,\alpha)}{p_b(x_i,\alpha)} + (1-\theta)\right]$$

Observe that the first term is not a constant: its omission would throw information away (background events carry constraining power on the nuisance!).

→this shows that the task of learning the density ratio p_s/p_b usually performed by binary classifiers is no longer sufficient.

What if we have *no* prior ?

Absence of information on a nuisance parameter is more common in HEP. We can still solve the problem by a parametrization of its effect.

Consider a search for a new particle of unknown mass M: usually, M influences in a smooth manner the observable event features x. If a classifier assumes a value $M_1=M+\alpha$ in training, its performance will degrade as α deviates from zero. M is thus in earnest a nuisance parameter.

One may train n classifiers using data simulated at n different M values, but the solution is sub-optimal (1/n use of total available data)

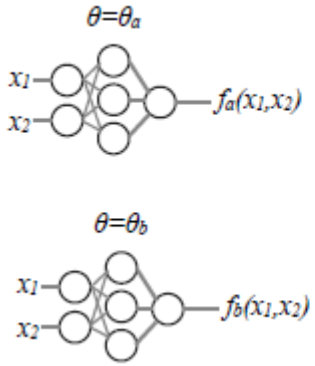
Better solution: parametrize the effect of M in the classifier [15]. The training data may be constructed as a mixture of different M hypotheses if M is included among the features.

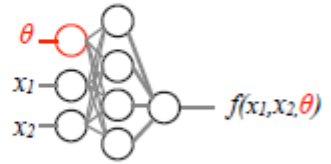
→ note that one must decide *what to do with the background* (for which M is undefined).

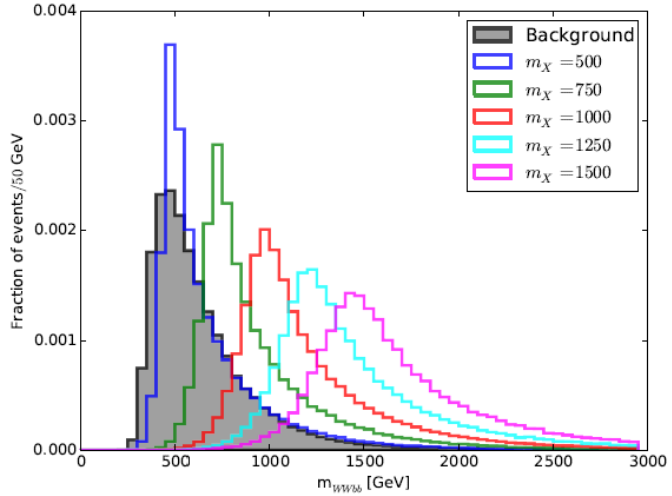
→ also note: this is not a Bayesian technique – the chosen admixture is not a prior on M, and it only affects the power of the classifier.

The advantage is that the network may meaningfully classify events for M* values not seen during training. An interpolation of the score for different mass hypotheses is also possible.

Example: X→tt in ATLAS





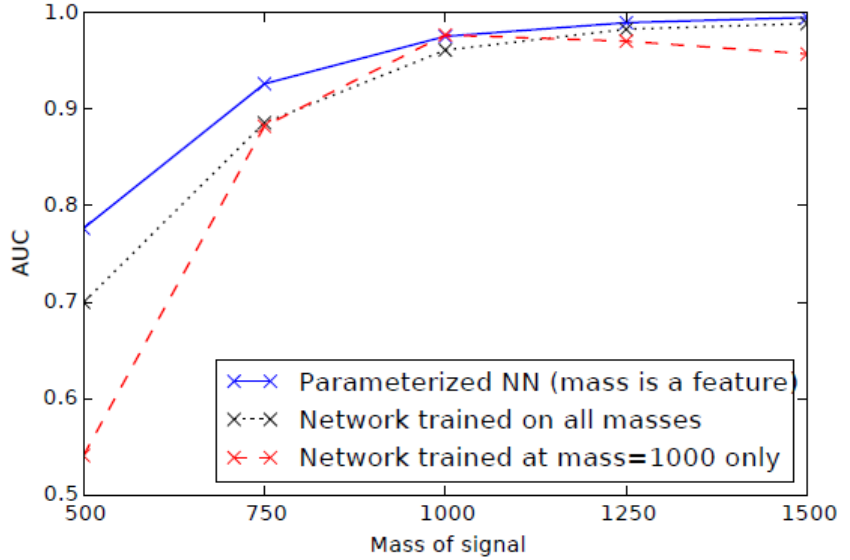


The technique proposed by Baldi *et al.* in[15] was tested with DELPHES[16] simulations, searching for a ttbar resonance within non-resonant ttbar backgrounds.

Random M values were used for the background in the training.

The network provides different scores for same features x, depending on the value of the nuisance M.

The parametrized network was shown to perform as well as individual NN on the mass points at which the latter were trained, but better than a NN trained on a mixture.



Calibrated discriminative classifiers (*)

Cranmer *et al.*[17] have produced a general proof of the viability of parametrized classification, by studying approximations of the likelihood ratio.

They showed how the **likelihood ratio of two PDFs can be kept invariant under dimensionality reductions**, if the transformation is itself monotonic with the LR. *I.e.*,

$$LR(x,\theta_0,\theta_1) = \prod_x \frac{p_X(x|\theta_0)}{p_X(x|\theta_1)} = \prod_x \frac{p_U(u=s(x)|\theta_0)}{p_U(u=s(x)|\theta_1)}$$

provided that U=s(X) is based on a parametrized function s which is monotonic with the ratio $p_X(x|\theta_0)/p_X(x|\theta_1)$.

Since the Neyman-Pearson lemma proves that the LR is the most powerful test statistic for tests of simple hypotheses, one tries to construct classifiers that approximate it, and the above equivalence proves quite effective as one may construct s(x) as a discriminative classifier (NNs will work well, as they provide smooth variation of s(x) as x varies).

The map s is one-dimensional, so evaluating $p(s(x)|\theta)$ is a much simpler task than evaluating $p(x|\theta)$. Histograms or KDE algorithms can be used. A calibration is necessary to approximate the probability ratio given the estimate of s(x); examples show this can be done and the technique is effective.

5. Decorrelation, penalization, adversaries

When a direct parametrization of nuisances proves impractical to implement, there are several alternatives. We can broadly lump them into three classes:

1) Techniques that operate a preprocessing of training data to reduce or remove the dependence of classifier score on the nuisance parameters
2) Construction of a robust optimization objective for the classification task, by penalizing the loss such that it becomes insensitive to α
3) Use adversarial setups to achieve the above result

In what follows we look at a few examples of these methods, to gauge their applicability and merits

Mass decorrelation

The most important use case of the first technique addresses the issue of keeping the background mass distribution unbiased, in signal searches in a mass spectrum.

The simplest way to avoid a reshaping of the mass PDF of background events is called *planing*[18,19]. One may implement it by pre-selecting training samples for S and B such that they have the same PDF on the variable to be planed. But this is sub-optimal.

Better strategy for planing: weight each training event with w(M) as follows:

For signal, $w(M_{rec}) = 1/p_S(M_{rec})$
For background, $w(M_{rec}) = 1/p_B(M_{rec})$

The weights enter the calculation of the loss during training, but are not used in validation or testing.

Planing is more effective than its simplicity would suggest! (some evidence is shown later)

Limitations occur when other event features indirectly inform the classifier on the value of the planed variable, when it carries discriminating power.

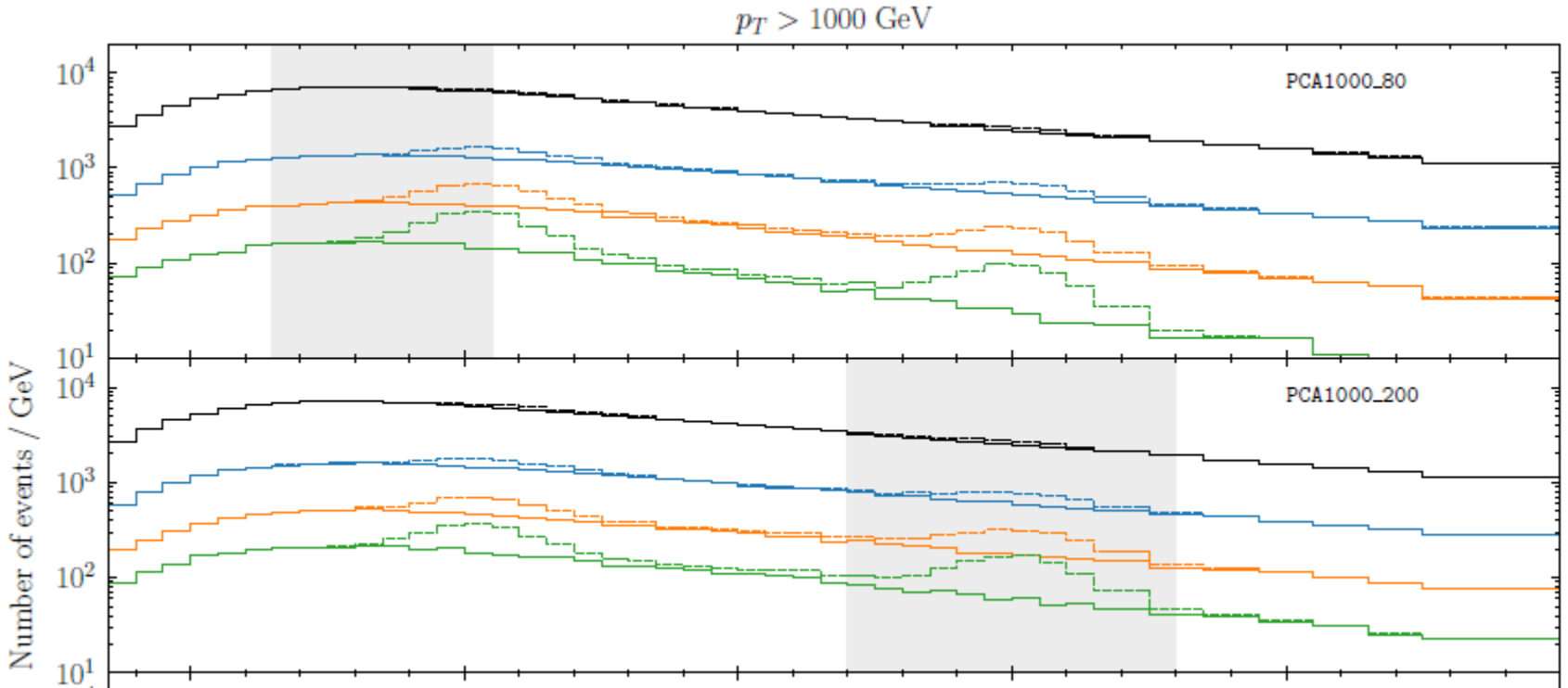
Mass decorrelation in boosting, reprise

Another way to preprocess the data before a NN, to decorrelate the classifier output from the jet mass, is to use PCA on the NN inputs[18].

In the considered case the 17 inputs were a basis set of n-subjettiness variables, and the data was binned in jet mass and p_T, PCA acting on each bin separately.

The technique was tested on searches for H→AA→bbbb and H→WW →qq'qq' with several mass hypotheses.

The discriminants are shown to preserve the mass distribution and are effective also outside of the range of masses where they are trained (grey areas)



Modified boosting and penalized methods

Searches for low-mass resonances in Dalitz plots provide a 2D use case: striving for **uniform selection efficiency** in the plane.

Uboost[22] relies on BDTs to improve signal purity, and a boosting technique to avert biases built on AdaBoost[23], a standard technique for boosting based on increasing weight of misclassified events during the DT generation sequence.

Uboost has the following rule:

$$w_i^n = c_i^n u_i^n w_i^{n-1}$$

where

$$c_i^n = exp(-\gamma_i p_i^{n-1})$$

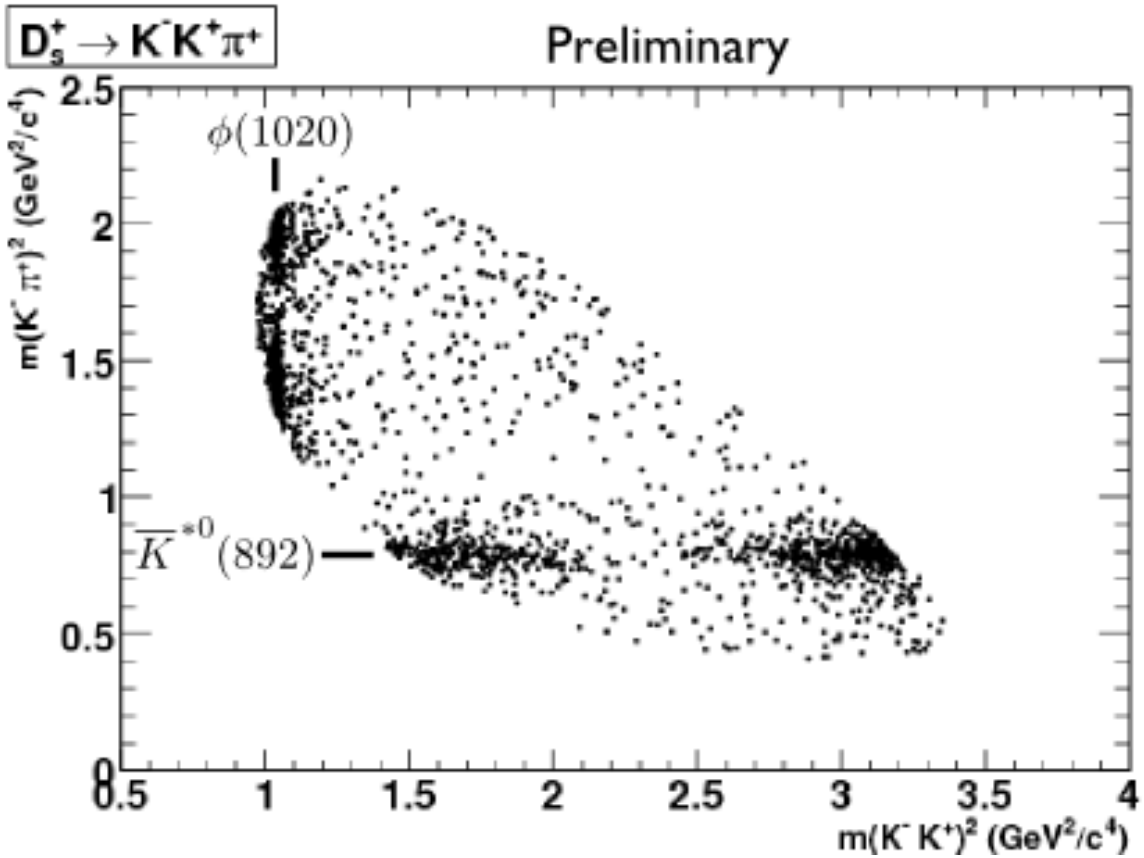
is the AdaBoost weight (γ=+-1 for signal and background, and p is the prediction of the previous iteration), and u is the inverse of the density of signal in proximity of the tested event, computed with kNN (u=1 for background).

(Background: what is a Dalitz Plot?)

When studying decay reactions such as A→B+C+D, one may construct pairs of independent kinematical variables, *e.g.* connected to the pairs (BC), (BD)

The presence of structure in the scatterplots indicate that the reaction proceeds through an intermediate state, A→XD→BCD

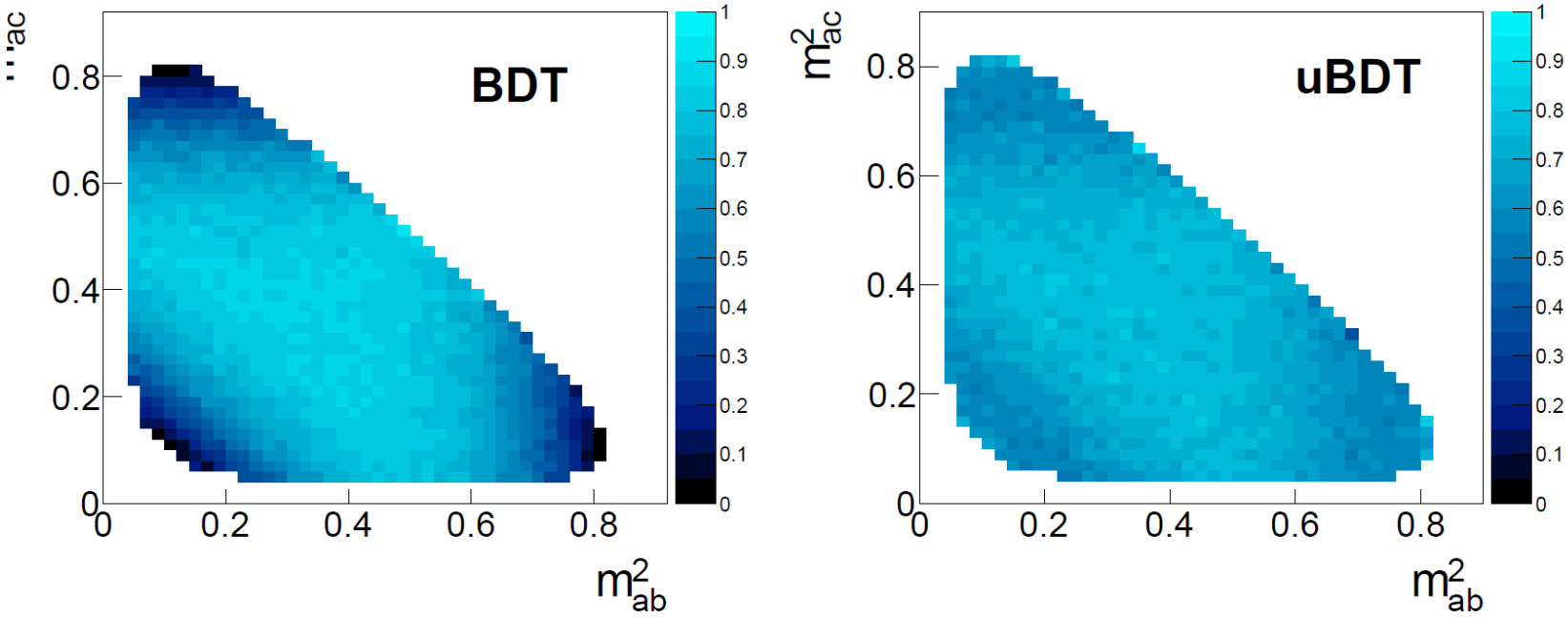
The analysis of a Dalitz plot requires a good uniformity of the «non resonant» background events to put in evidence the structures



Uboost performance

Uboost was shown to work well in the considered application, despite the considerable CPU cost (due to use of kNN method and consideration of different efficiency values in constructing the BDT score).

E.g., here are the maps for average selection efficiency of 70% obtained with a regular BDT and with Uboost.
Almost no loss in performance was seen, while achieving the wanted uniformity.



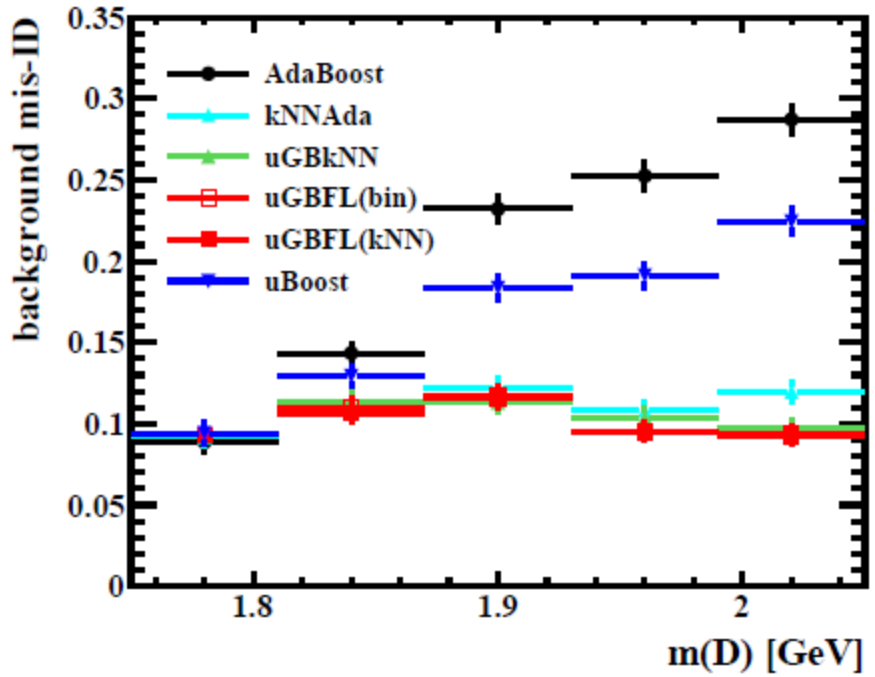
Building on Uboost (*)

The thread started by Uboost was followed by a number of other studies. An alternative also targeting Dalitz analysis is kNNAdaBoost[24], which achieves the uniformity of acceptance in the Dalitz variables by modifying the weights, including information on the classification probability of k nearest neighbors to each event:

$$w_i^n = w_i^{n-1} exp\left[-\gamma_i \sum_j a_{ij} p_j\right]$$

The a_{ij} matrix contains information on the density of events of the same class around event i by setting a_{ij}=1/k for them, 0 otherwise.

Rogozhnikov *et al*.[24] have investigated a number of variants of this concept for an application targeting the $D_s \rightarrow \pi^+\pi^-\pi^-$ signal, finding good results and better generalization vs training in different regions of the mass of the D candidate



DisCo

Kasieczka and Shih recently published[25] a method to decorrelate a nuisance parameter by incorporating a «distance-correlation»-inspired regularization term in the loss of a NN.

One first defines a distance covariance

$$dCov^2(X,Y) = \langle |X-X'||Y-Y'| \rangle + \langle |X-X'| \rangle \langle |Y-Y'| \rangle - 2\langle |X-X'||Y-Y''| \rangle$$

where |.| is the Euclidean norm, and (X,Y), (X',Y'), (X'',Y'') are i.i.d. pairs from the joint PDF. The so-named distance correlation, defined as

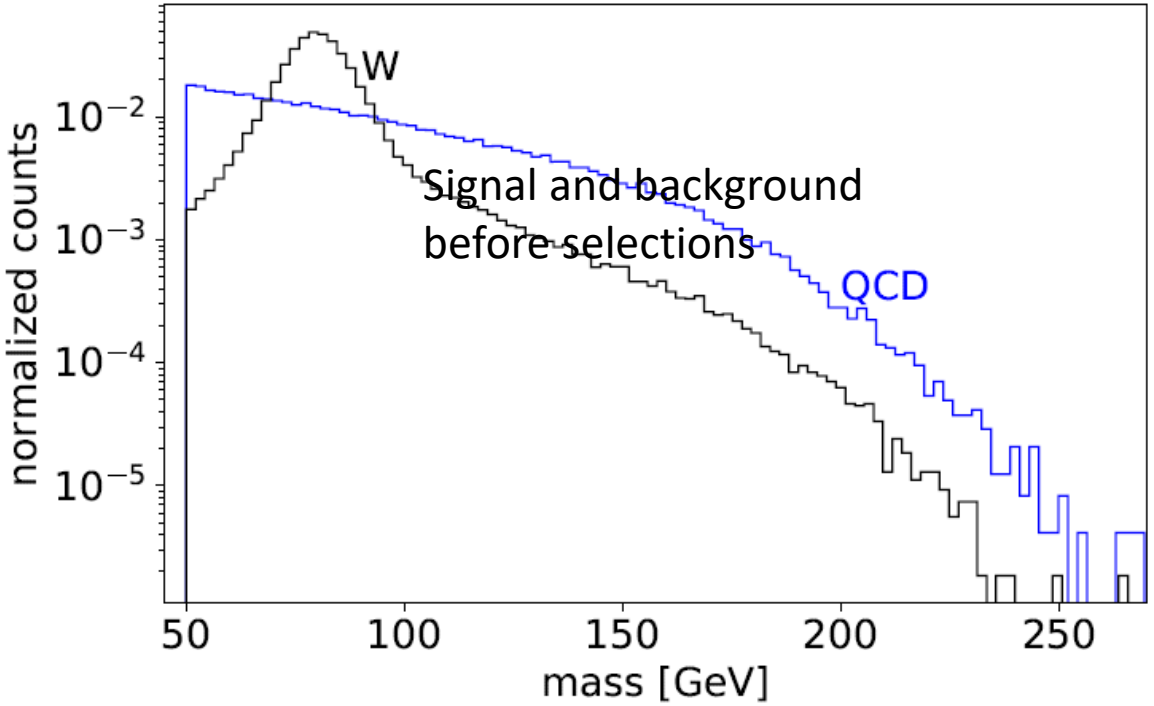
$$dCorr^2(X,Y) = \frac{dCov^2(X,Y)}{dCov(X,X)dCov(Y,Y)}$$

is then a [0,1] measure, null only if x,y are fully independent. Crucially, it is differentiable and computable with data samples, so it can be included in the loss function (for label y and mass m) with a penalty regularization factor λ

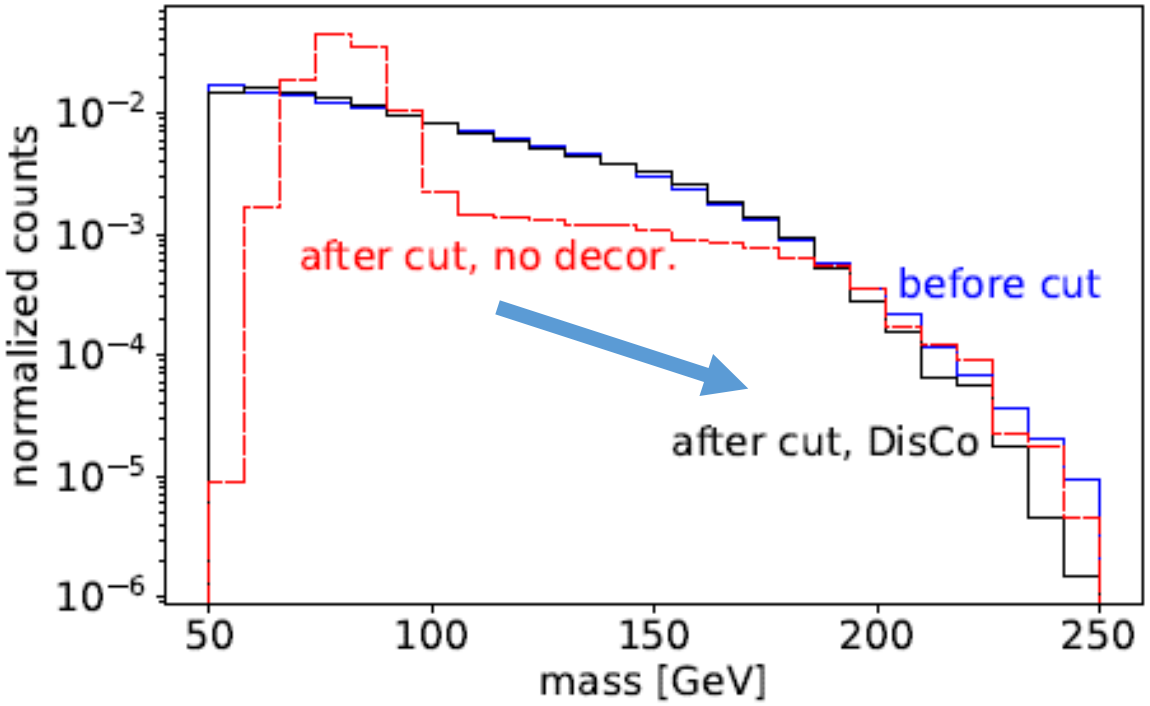
$$L = L_{class}(y, y_{true}) + \lambda\, dCorr^2(m, y)$$

DisCo action

Kasieczka and Shih test DisCo on W-boson tagging in simulated ATLAS data, reweighted to have a flat p_T distribution. They show that a NN discrimination of W-like jet images produces a biased mass distribution for QCD backgrounds, while DisCo preserves the QCD mass shape (bottom left).



ies in

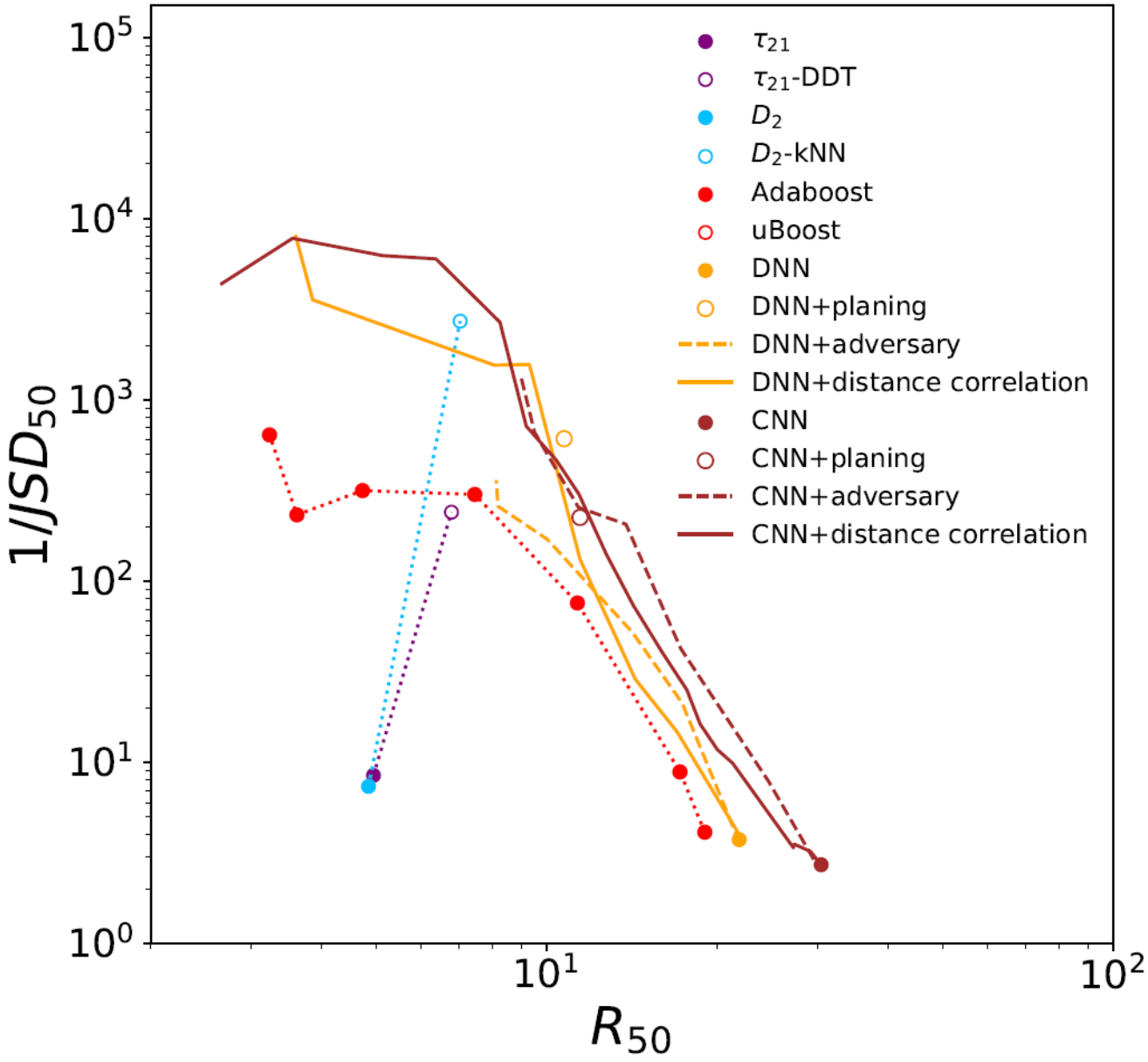


Comparisons

A comparison of the background rejection (x axis) of different W taggers, retrofitted with decorrelation methods (planing, adversarial NN, and DisCo regularization) shows that DisCo performs well. Surprisingly, also planing seems to do a decent job in this particular task.

DisCo regularization works well with complex image-based CNN setups, too.

More studies in other setups are advisable…



Above: a measure of decorrelation (inverse of Jensen-Shannon divergence between QCD bgr before and after 50%TPR selection) as a function of background rejection at 50%TPR.

Another penalization scheme

Wunsch *et al.*[26] more directly target situations when the nuisance parameter is a systematic source liable to modify the PDF of the observable features of data. They create a differentiable model of the NN-transformed features x by Gaussian smoothing of their histograms:

$$\mathcal{N}_k(f(x)) = \sum_b \mathcal{G}_k(f(x))$$

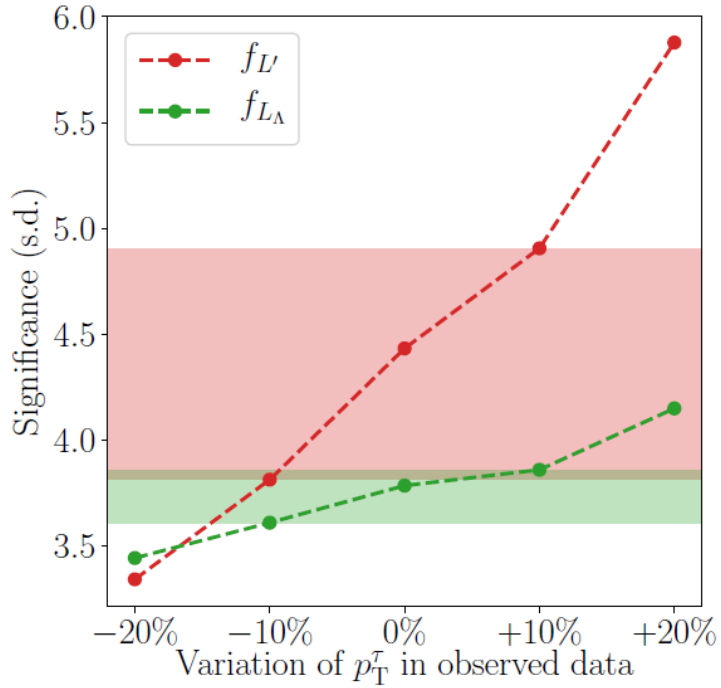
Above, k is the bin index, and the sum runs on events in a batch; f(x) is the NN output for x. The loss of a classifier may then be complemented with a term describing the difference between smoothed PDF of NN output for x and x+α:

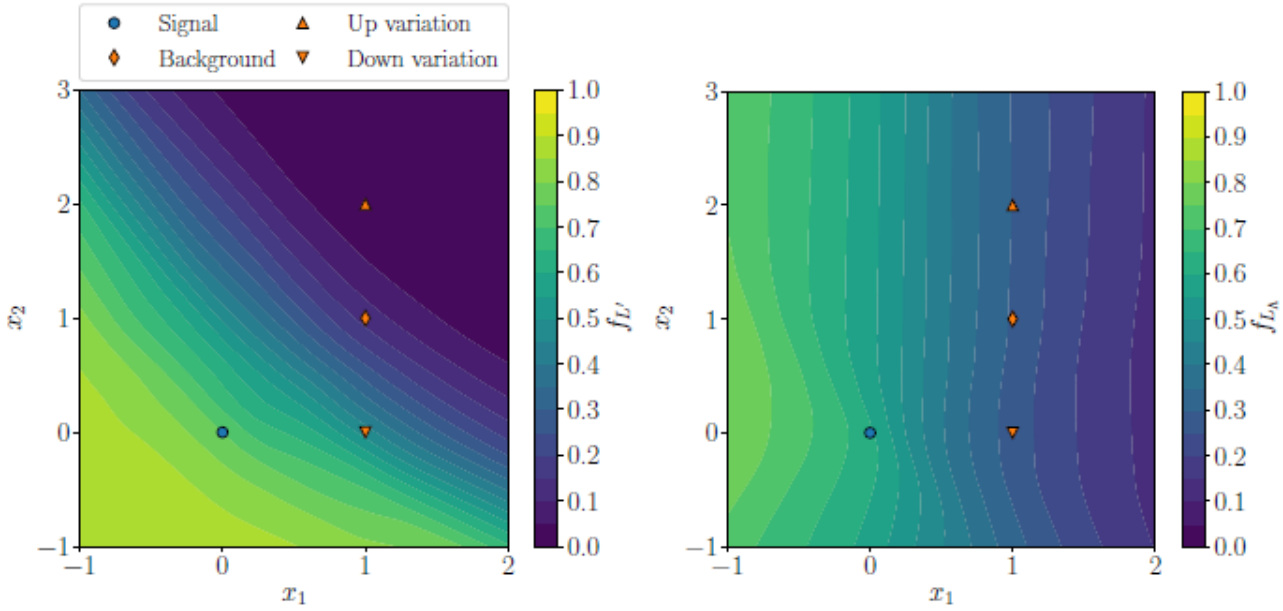
$$L(\lambda) = L_0 + \lambda \frac{1}{n} \sum_k \left(\frac{\mathcal{N}_k(f(x)) - \mathcal{N}_k(f(x+\alpha))}{\mathcal{N}_k(f(x))} \right)^2$$

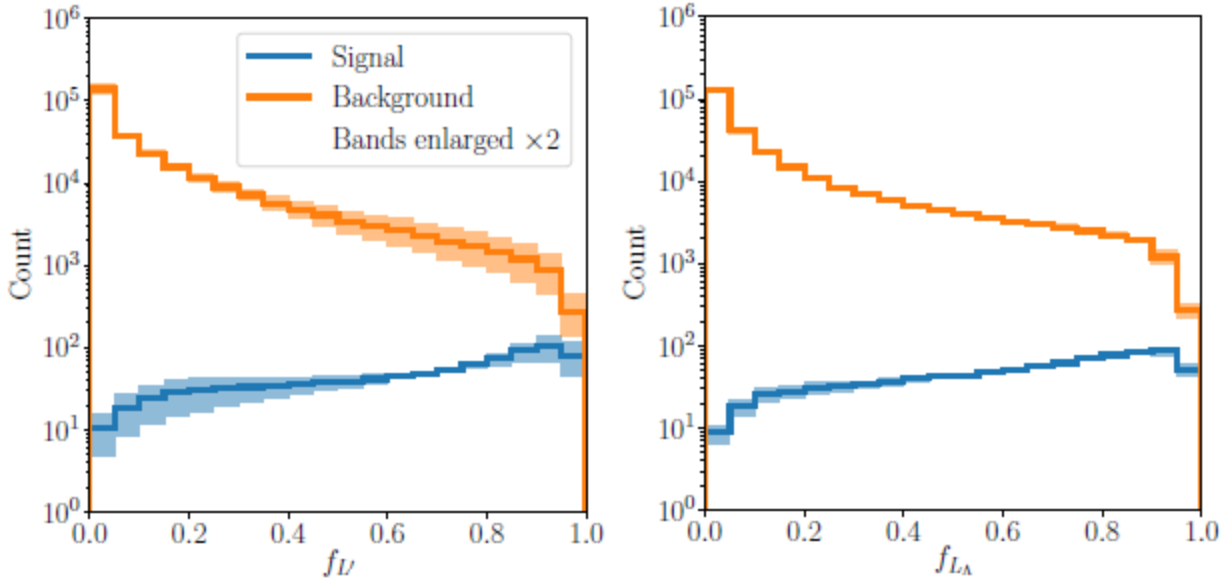
Examples

The method is proven to effectively decouple the classifier output from the systematic source in a synthetic 2D Gaussian example (left and right, bottom). It also works in Higgs Kaggle challenge data, where the systematic is the momentum scale of the tau leptons (right).

For the latter, the penalization reduces the dependence on significance on systematic source, but it does not increase the significance of the observable signal (top), so the advantage of the technique should still be proven in other situations.







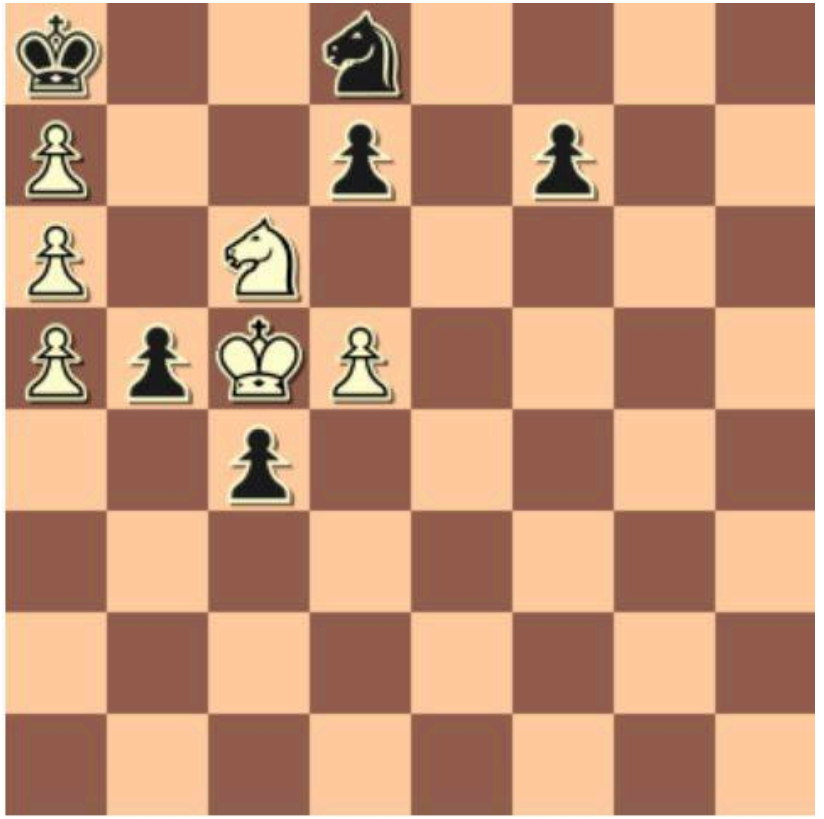
Adversarial setups

Adversarial setups attacking the decorrelation problem should be considered an extension, if not the logical next step, of the penalized loss methods seen above.

- The loss is still the combination of two parts – a BCE term and a penalization contributed by the adversary, modulated by a hyperparameter.

However, in adversarial settings there is a symmetry between the two competing tasks.

An issue introduced is the convexity of the objective, which is guaranteed with positive penalties, while in ANN one searches a saddle point of the two competing losses.



Above: minimax routines (originally coming from Nash theory of equilibrium) were specialized to teach machines play chess.
In the shown position (T. Dorigo, 1989) white to move wins. If you don't want to follow the lecture you may try solving it, but bear in mind that you will need to apply backward propagation for that!

Where the idea comes from

Adversarial architectures were investigated in computer science to achieve **domain adaptation** of discriminative classifiers[28][29] much before they were adopted in HEP.

- General issue: training and test data are not drawn from same PDF

may arise when they come from different domains, or if the simulation (used for training) is imperfect model of (real) test data.

- Other common situation in DA is that problem is semi-supervised (labels not available for all test data) → let's leave this for later

Solutions usually involve finding a data representation that is maximally insensitive to their source

→ task an ANN to learn such representation, while competing with the one that tries to separate labelled classes of training data[30].

Learning to pivot

The first use of ANNs to achieve robustness to systematics in HEP comes from the work of Cranmer, Kagan and Louppe[31] who sought pivotal[32] classification scores $f(x;\theta_f)$: ones independent on nuisance α (θ_f are the classifier parameters).

The adversary, with parameters θ_r , tries to guess α from $f(x;\theta_f)$, and the loss is defined globally as

$$E(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \mathcal{L}_r(\theta_f, \theta_r)$$

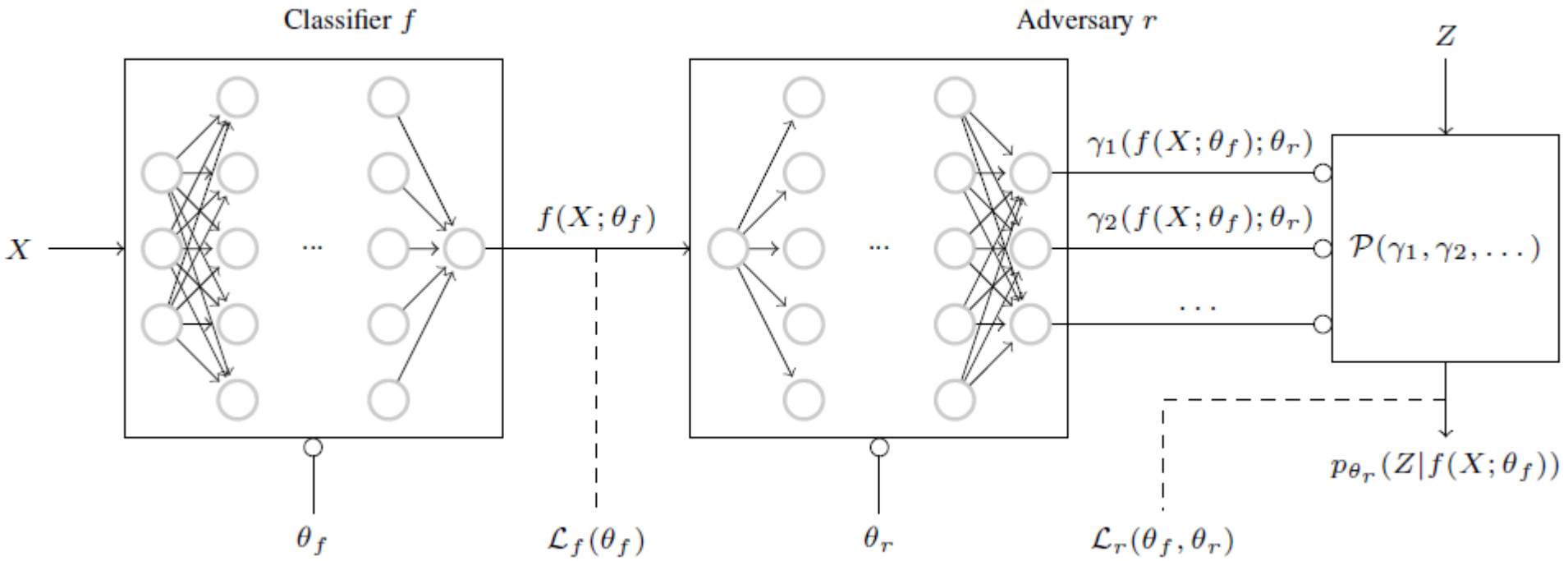
The minimax solution of this problem is reached for

$$\hat{\theta_f}, \hat{\theta_r} = argmin_{\theta_f} max_{\theta_r} E(\theta_f, \theta_r)$$

A convergence of the above constrained problem cannot be guaranteed in general; a hyperparameter λ can be used to tune the adversary term.

Pivoting ANN architecture

The architecture is a series of two discriminative classifiers: the adversary tries to model p(z|f(x)), and the global loss forces this toward the unconditional prior p(z). When this happens, f is independent on z.

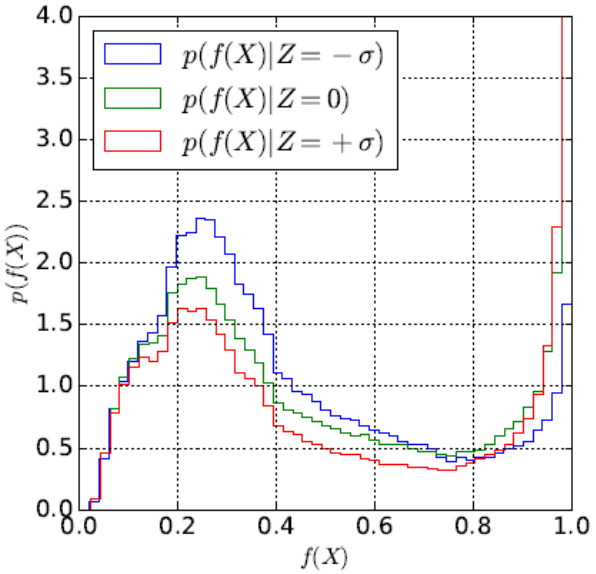


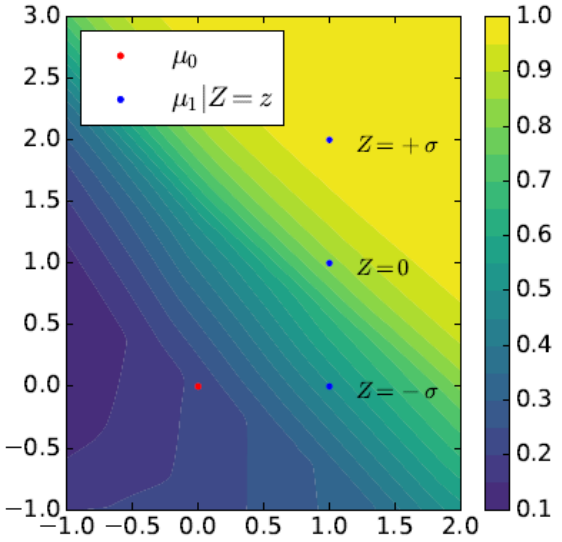
Experiments with the pivot

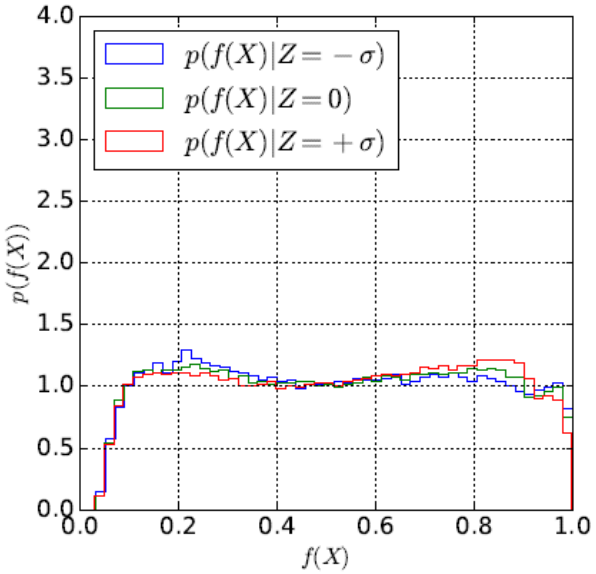
The graphs below describe the same synthetic example we saw earlier from Wunsch *et al*., although the publication order is inverted.

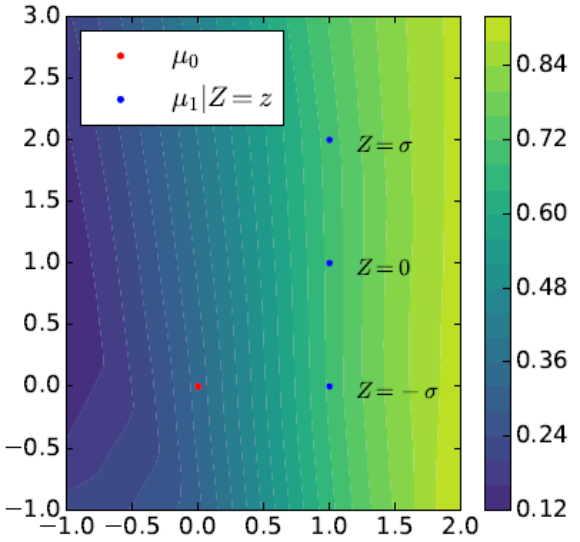
Left: a standard NN produces f(x) depending on nuisance Z (the vertical location of the signal 2D Gaussian PDF)

Right: the pivoting setup makes f(x) independent on Z.







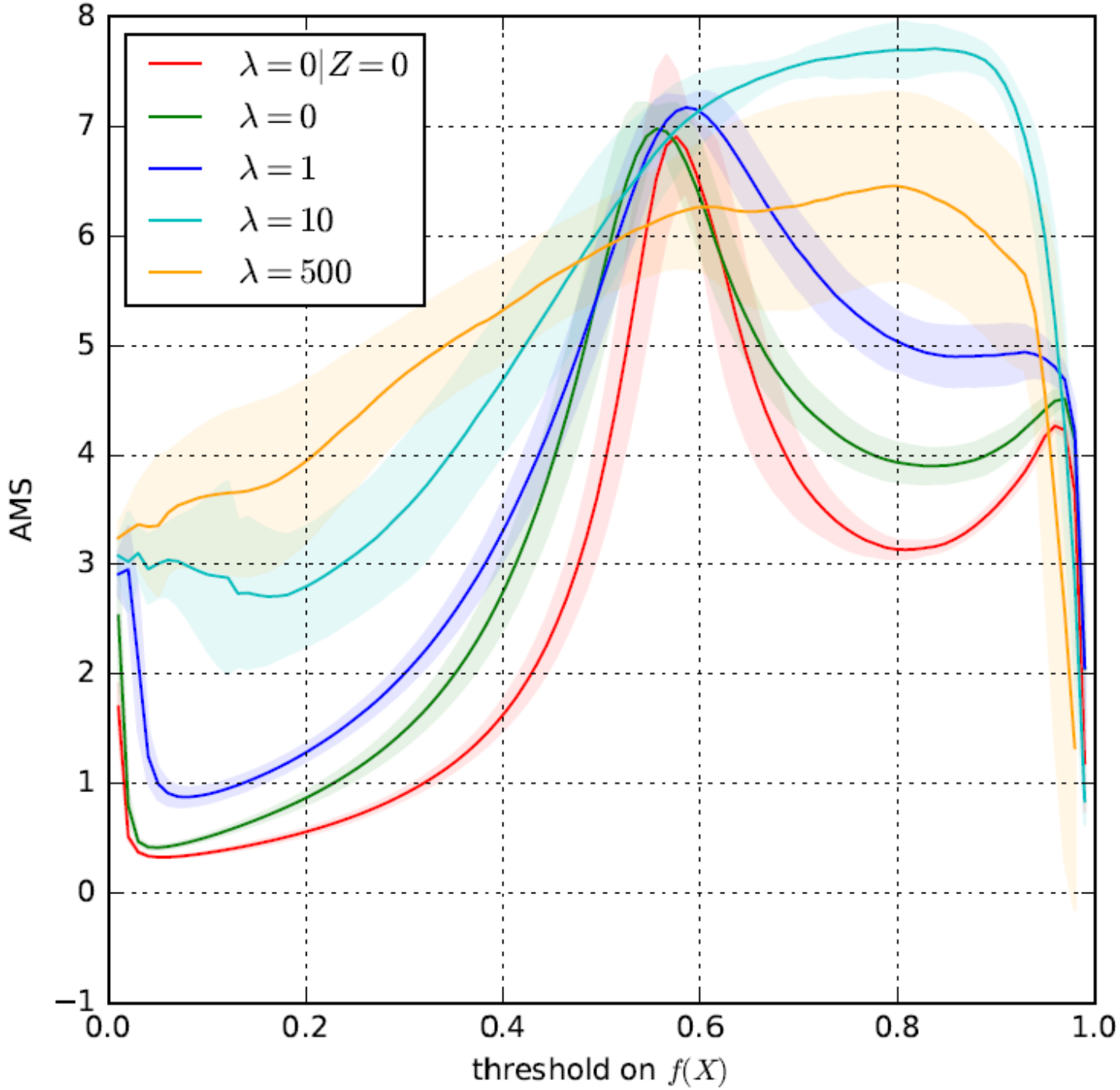


LHC example

The technique is demonstrated on boosted W tagging in ATLAS, with pile-up being the nuisance (Z=0 no pileup, Z=1 PU-50 conditions).

As in this case finding a f that is pivotal while minimizing the loss L is probably not possible, one must optimize a suitable objective (AMS) WRT the hyperparameter.

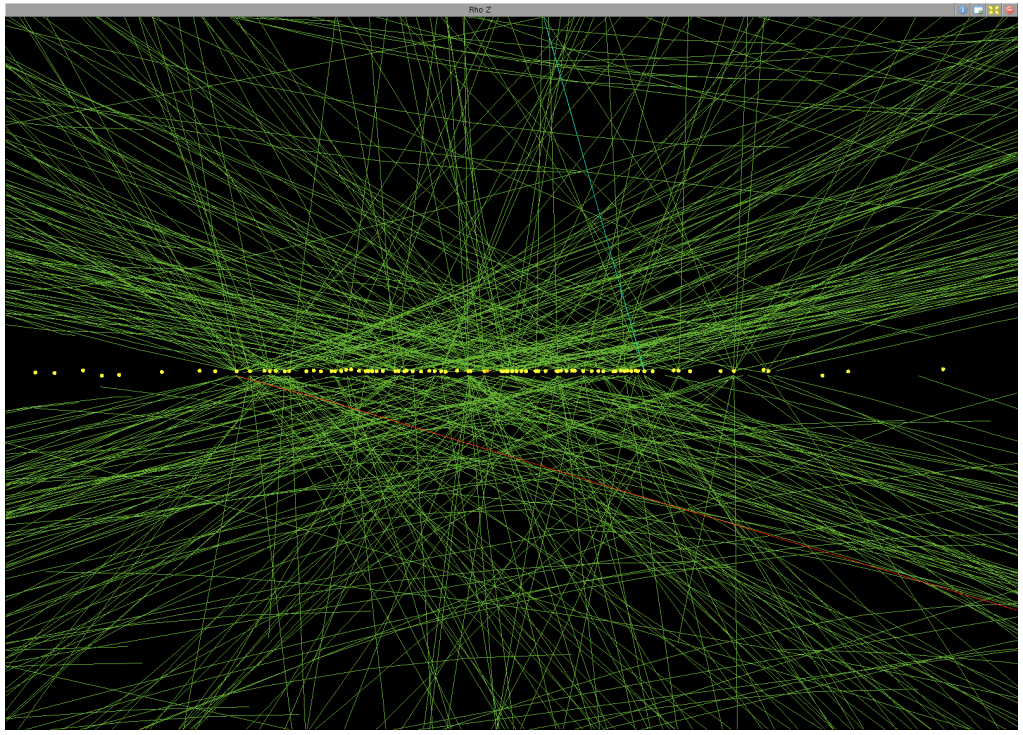
For a suitable choice of λ (10) the AMS reaches a higher maximum



(Background: Pile-up at the LHC)

Pile-up is the effect of having dense packets of protons colliding every 25ns in the core of ATLAS and CMS: one gets tens of independent proton-proton collisions producing overlapping signals.

When one of these collisions is interesting, all others degrade the extractable information → complex pattern recognition and regression issues

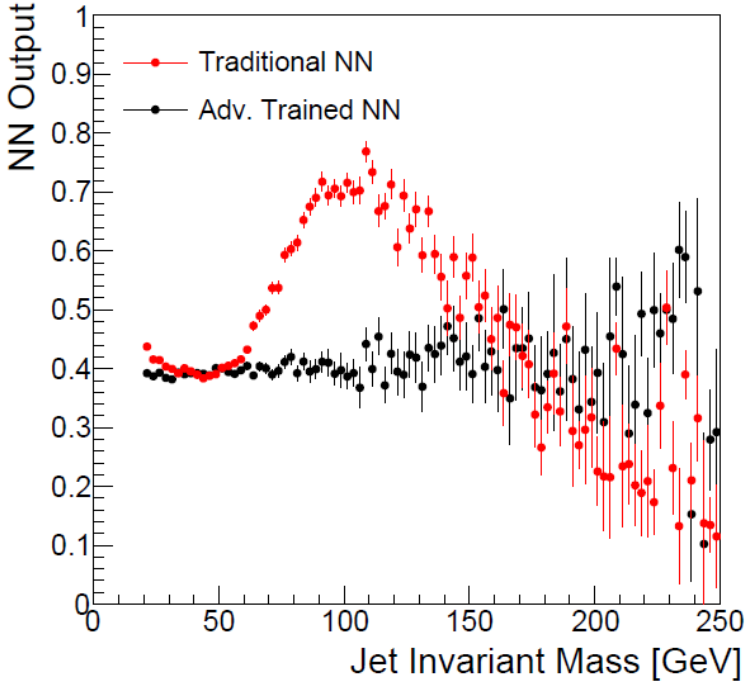


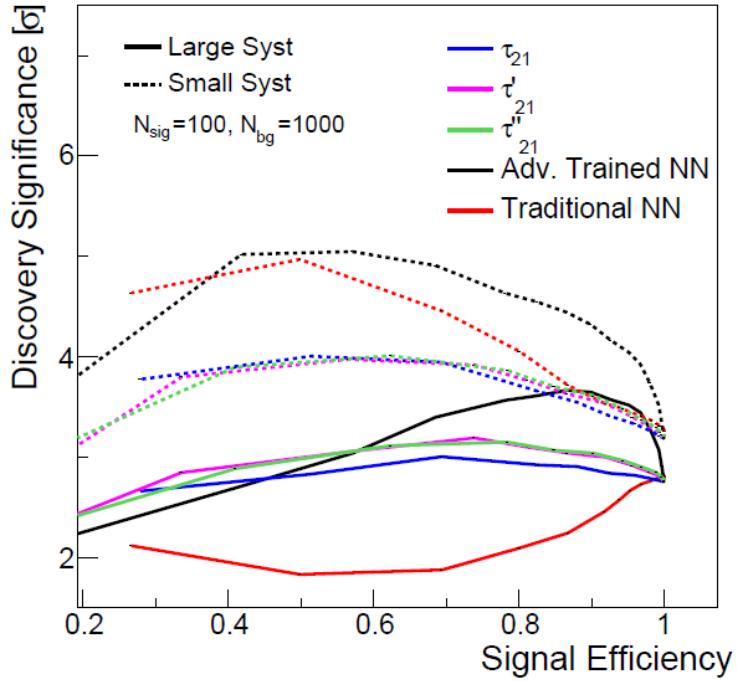
Above: identified collision vertices (yellow) during a bunch crossing in CMS. In green are (some of) the measured charged particle tracks

Further developments / 1

The adversarial setup seen above has been tested and compared with other decorrelation techniques[33] in boosted jet tagging. The dependence on jet mass is effectively absorbed by the pivotal properties of the ANN (top).

Authors further consider the full effect of the reduced mass dependence on background shape deformations, extracting the resulting significance (bottom) for a NN, the ANN, and subjettiness selection with and without the analytical decorrelation already discussed before, for different levels of systematic uncertainties on the QCD background shape.





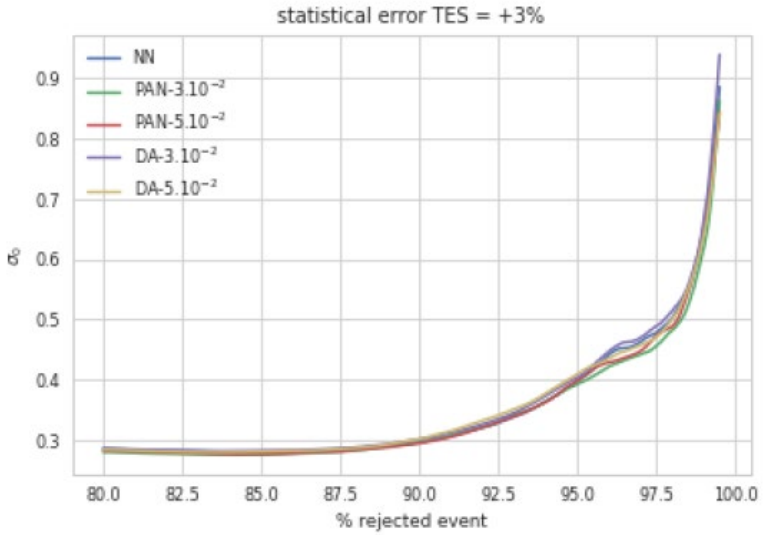
Further developments / 2 (*)

Estrade *et al.*[34] compared the pivot ANN technique of Louppe *et al*. to a data augmentation technique and to a tangent propagation method, using the H→ττ Kaggle challenge data with a systematic uncertainty added on the tau jet energy scale, similarly to[28].

The systematic is sampled from a Gaussian (1%,3%,5% nuisance) in training data, but is fixed in test data.

- Data augmentation: the training data is mixed at different values of the nuisance – rationale is that with sufficient data the NN should be able to learn the manifold
- Tangent propagation: consider the systematic as a coherent geometrical transformation, f(x,α), differentiable, and the model is regularized by partial derivative of NN score WRT nuisance parameter.

Results are not conclusive, only seem to favor ANN over plain NN



Further developments /3 (*)

- Blance, Spannowsky, and Waite[36] use ANN as a preliminary step to use of autoencoders (AE) for unsupervised classification, to reduce impact of systematics in the task. Their goal is to search for new physics without being dependent on Monte Carlo (MC) simulation mismodeling.
- They smear MC reconstructed objects and use ANN to desensitize AE response to smearing; use case: X→tt resonances. Show that ANN+AE provide reasonable solution

- Englert *et al.*[37] consider theoretical uncertainties, which affect the data in a more coherent way than other nuisances do. Large theoretical scales uncertainties affecting H+jet production at high p_T are considered. Sensitivity to the process can be retained by training an ANN to make robust classification WRT the scale.

6. Semi-supervised approaches

Weakly-supervised and semi-supervised learning techniques have been proposed to close the gap between learning from simulated and real data:

- Simulated data are fully labeled, but they are often an imperfect model
- Real data are unlabelled or only partly labelled

The approaches strive to learn useful models from partial, non-standard, or noisy label information. They are thus potentially useful for reduction of the impact of certain systematic uncertainties in HEP problems.

The challenge is that these methods typically rely on assumptions (known fractions, independence of PDF of features) that are hardly met in practice. We see a few examples in what follows.

LLP (Learning from Label Proportions)

LLP is a weak-supervision approach that may allow the training on real data (Dery *et al.*[38]).

While in a full supervision setup one tries to find a score fulfilling

$$f_{\text{full}} = \text{argmin}_{f':\mathbb{R}^n \to \{0,1\}} \sum_{i=1}^{N} \ell\left(f'(x_i) - t_i\right)$$

(l is a loss, *e.g*. mean squared error, and t is the target label), in LLP one only exploits knowledge of the fraction of each label in training data:

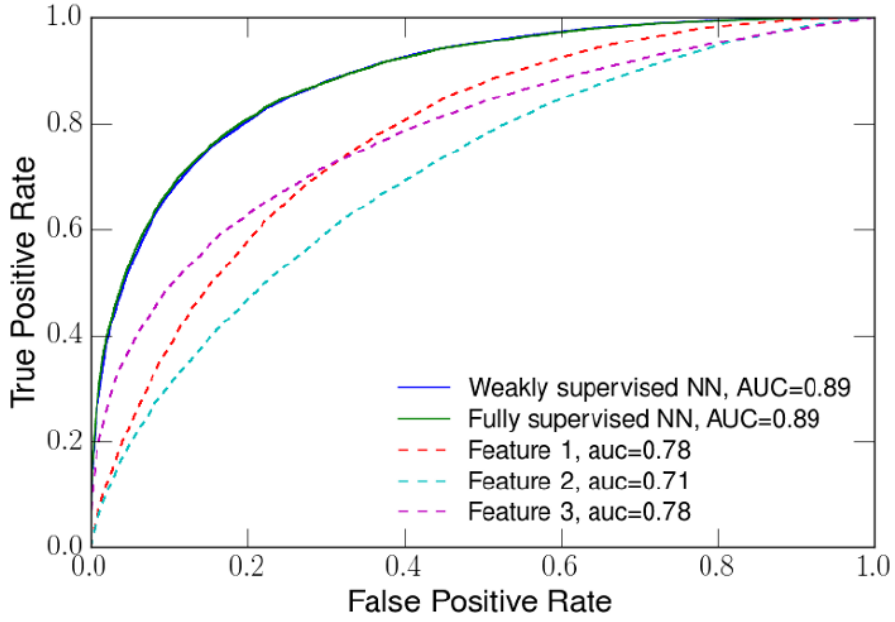
$$f_{\text{weak}} = \text{argmin}_{f':\mathbb{R}^n \to [0,1]} \ell\left(\sum_{i=1}^{N} \frac{f'(x_i)}{N} - y\right)$$

(y=average label)

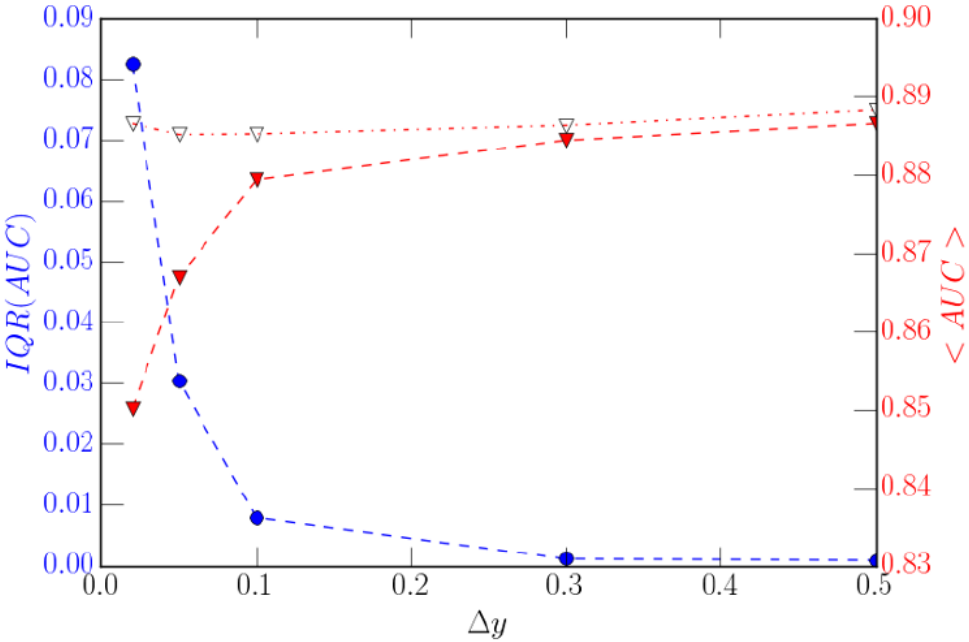
The problem is thus very ill-constrained, but a minimization of the loss can still be performed with batches of data of different proportions, *as long as the PDFs of the features do not change in the batches*.

LLP proof of concept

Using a 3-layer NN and synthetic data with class proportions between 0.2 and 0.4, with three features, allows LLP to perform equally as well as a fully supervised method.



The range of performances, due to randomness of the inputs, decreases when training data has wider range of class proportions (Δy on x axis, bottom). The overall performance also increases with the diversity of input samples.

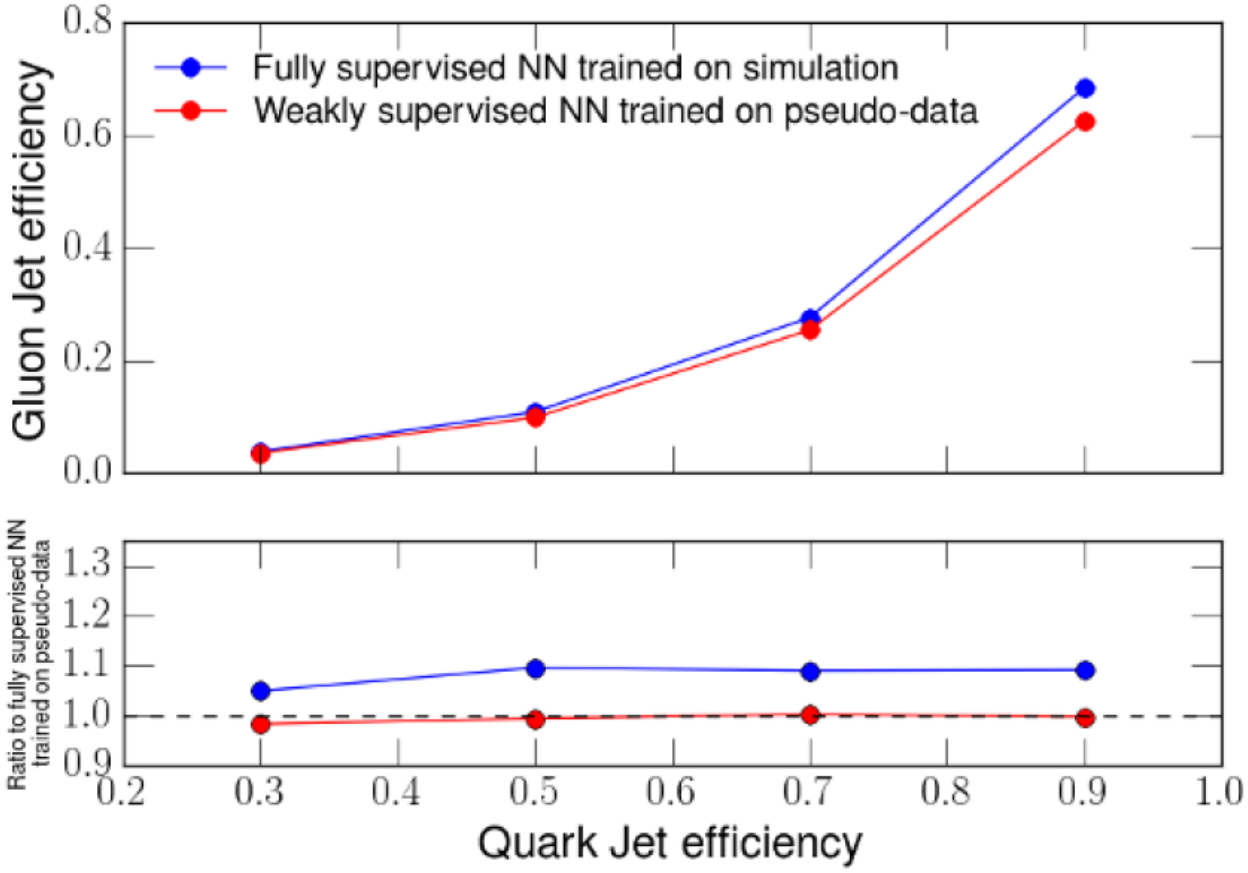


Test of LLP on Q/G discrimination

Authors of [38] argue that quark/gluon jet separation lends well to this method, as *a priori* fractions in different physical processes are «well estimated» from QCD+PDF theory – while shapes may be harder to simulate (→systematic from modeling).

12 different samples are obtained by binning in dijet pseudorapidity difference (quark fractions vary from 0.21 to 0.32).

Distortion of real data is mimicked by modeling previous studies; then a comparison with a full supervised classifier shows 10% advantages (lower G efficiency for given Q efficiency)



CWoLa – classification without labels

One limitation of LLP is the need to precisely know the class labels of training subsets. A technique by Metodiev *et al.*[39], CWoLa, overcomes this by using as labels the identifiers of the different mixed samples.

CWoLa is based on the fact that the optimal binary classifier is a function of the density ratio between the components, so the discrimination of the two mixed samples works also for pure classes:

Theorem 1. *Given mixed samples M_1 and M_2 defined in terms of pure samples S and B using eqs. (2.3) and (2.4) with signal fractions $f_1 > f_2$, an optimal classifier trained to distinguish M_1 from M_2 is also optimal for distinguishing S from B.*

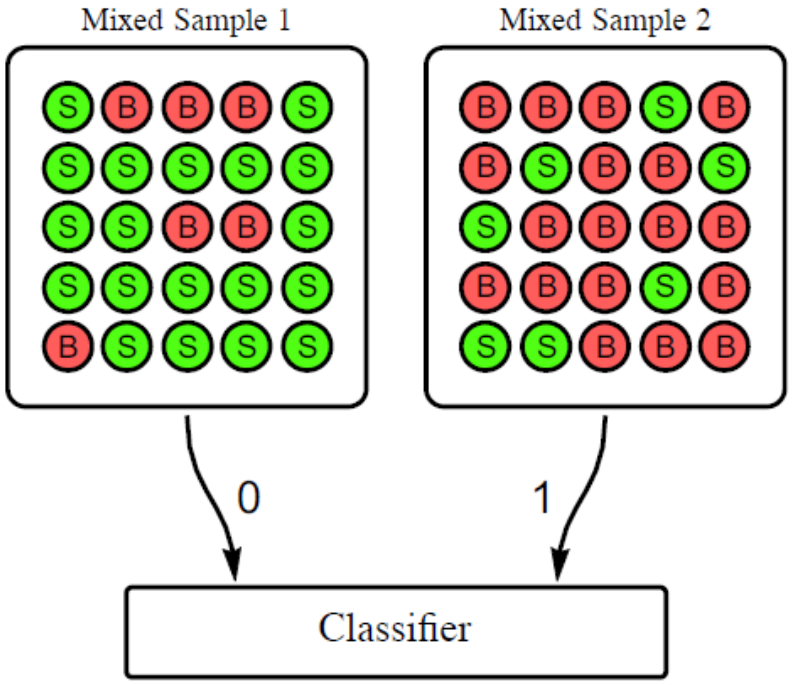
Proof. The optimal classifier to distinguish examples drawn from p_{M_1} and p_{M_2} is the likelihood ratio $L_{M_1/M_2}(\vec{x}) = p_{M_1}(\vec{x})/p_{M_2}(\vec{x})$. Similarly, the optimal classifier to distinguish examples drawn from p_S and p_B is the likelihood ratio $L_{S/B}(\vec{x}) = p_S(\vec{x})/p_B(\vec{x})$. Where p_B has support, we can relate these two likelihood ratios algebraically:

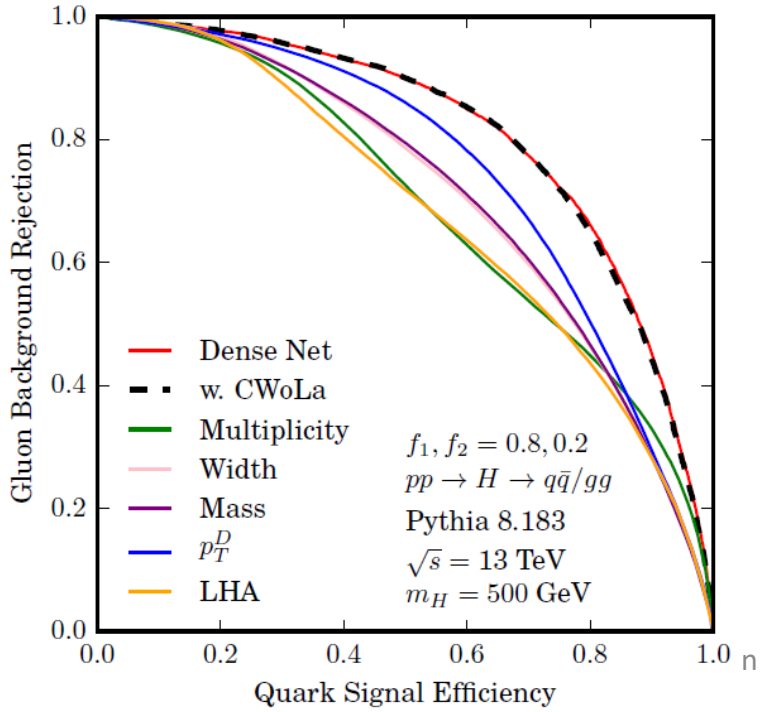
$$L_{M_1/M_2} = \frac{p_{M_1}}{p_{M_2}} = \frac{f_1\, p_S + (1-f_1)\, p_B}{f_2\, p_S + (1-f_2)\, p_B} = \frac{f_1\, L_{S/B} + (1-f_1)}{f_2\, L_{S/B} + (1-f_2)}, \tag{2.6}$$

which is a monotonically increasing rescaling of the likelihood $L_{S/B}$ as long as $f_1 > f_2$, since $\partial_{L_{S/B}} L_{M_1/M_2} = (f_1 - f_2)/(f_2 L_{S/B} - f_2 + 1)^2 > 0$. If $f_1 < f_2$, then one obtains the reversed classifier. Therefore, $L_{S/B}$ and L_{M_1/M_2} define the same classifier. □

CWoLa at work

Tested on the same problem of Q/G discrimination, and with a NN as classifier, the CWoLa concept was shown to perform as well as a NN working on pure classes if trained on classes with 80%-20% class proportion split





Of course the algorithm still requires labelled data for tests of performance and choice of operating point, but the proof of principle is encouraging.

7. Inference-aware approaches

What we have seen so far are ways to cope with the imperfect knowledge of the generative model of our data, which affects the power of our simulation-based classification tasks.

There are now solutions that try to move away from the proxy classification task, and address directly the optimization of simulation-based statistical inference.

→ **This realigns task and objective**

The area of research[42] is sometimes called «Likelihood-free inference»

Here we discuss how some of these inference-aware approaches may be used to tame nuisance parameters in HEP.

Estimates of the likelihood ratio

As discussed earlier, a reparametrization and approximation of the likelihood ratio for all possible pairs of relevant parameters θ_0,θ_1 of a generative model $p(x|\theta)$ may allow[17] to efficiently solve the problem of inference in the presence of nuisances.

The method may be too CPU intensive to be practical in high-dimensional cases, as large datasets are required to approximate the LR.

A number of techniques were published by Brehmer *et al*.[44-46] to evaluate the LR in a data-effective manner, using information from the simulator to augment the training data.

These techniques may collectively be addressed as «learning efficiently from the simulator».

- A meaningful discussion of the wealth of ideas deployed for this would require a couple of lectures by itself

Inference-aware summary statistics

A complementary family of techniques to "likelihood-free inference methods" tries to construct summaries that are better aligned with inference goal, once nuisance parameters are accounted for.

Typical procedure in HEP:

1. "optimize" a classifier f(x) to best distinguish S(θ) from B (e.g., θ=σ, cross section of signal process), e.g. focusing on maximizing AUC or other figures of merit connected to observability of S (pseudo-significances)

2a. Choose operating point (*e.g.* cut on f), maybe accounting for variability of S and B PDFs, and perform a counting experiment on data above f cut;

2b. Parametrize shape and fit for signal fraction, accounting for nuisances as shape variations

→ In both cases, the optimization target (discrimination of S/B in absence of nuisances) is different from the true objective of the analysis (minimize uncertainty on parameter of interest)

For a realignment, we must inform the classifier of the effect of nuisances on the final measurement goal

INFERNO (Inference-aware neural optimization)

Idea of P. de Castro and TD[49]: make the loss of a NN aware of what we really want to make of the NN output, and simultaneously inject in it a parametrization of nuisances, so that a loss minimization perfectly matches the (stat+syst) variance minimization of the final measurement.

The NN constructs summaries that are differentiable WRT the nuisances, and this property is propagated to the inference step, such that a global minimization can be performed.

NN parameters are optimized by SGD within an AutoDiff framework (in TensorFlow); *a PyTorch implementation is in final phase of development by G. Strong.*

Problem 1: need to produce differentiable map of nuisance effect on features
→ Calls for custom solutions in HEP problems of different complexity

Problem 2: how to estimate the final variance on the parameter of interest?
→ Use the inverse of the Hessian matrix of a likelihood constructed with the summary statistic provided by the NN

INFERNO structure

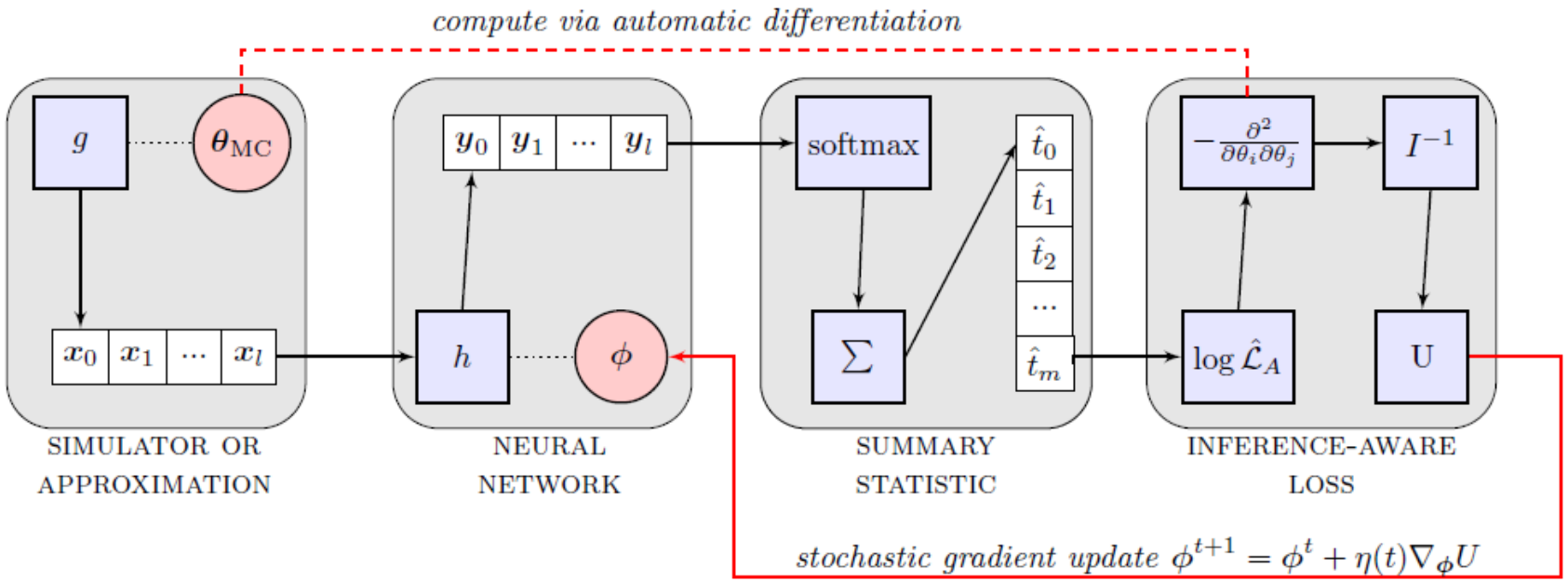
Block 1: A simulator or an approximation of it is used to sample observations given parameters θ

Block 2: NN with parameters ϕ produces outputs y

Block 3: A one-dimensional summary statistic, as a smoothed version of y, is produced by softmax

Block 4: An Asimov likelihood is constructed with the summary (e.g. a histogram of Poisson counts), and used to get Hessian matrix, yielding expected variance on parameter of interest

Autodiff allows to update the NN parameters given the value of the variance, to navigate with SGD to the optimal solution



INFERNO details

Given a sample of data D, the output of the NN (of parameters ϕ) is a set $f_i(x|\phi)$, with which we may construct a non-parametric binned likelihood by simply counting how often the data have maximum output on the i^{th} node:

$$t_i(D;\boldsymbol{\phi}) = \sum_{x\in D} \begin{cases} 1 & i = \underset{j=\{0,\dots,b\}}{argmax}\,(f_j(\boldsymbol{x};\boldsymbol{\phi})) \\ 0 & i \neq \underset{j=\{0,\dots,b\}}{argmax}\,(f_j(\boldsymbol{x};\boldsymbol{\phi})) \end{cases}$$

and using the summary t to write $L(D|\varphi) = \prod Pois[t(D|\varphi)|t(G_{MC};\varphi)]$

where G_{MC} is the generated simulation used for calibration.

The argmax is non-differentiable, so we can approximate the summary with the softmax operator:

$$\hat{t}_i(D;\boldsymbol{\phi}) = \sum_{x\in D} \frac{e^{f_i(\boldsymbol{x};\boldsymbol{\phi})/\tau}}{\sum_{j=0}^{m} e^{f_j(\boldsymbol{x};\boldsymbol{\phi})/\tau}}$$

where τ is a temperature HP.

We may finally construct an Asimov likelihood, whose maximization will provide the true parameter as MLE:

$$\hat{\mathcal{L}}_A(\boldsymbol{\theta};\boldsymbol{\phi}) = \prod_{i=0}^{m} \text{Pois}\left(\left(\frac{n}{l}\right)\hat{t}_i(G_{\text{MC}};\boldsymbol{\phi}) \mid \left(\frac{n}{l}\right)\hat{t}_i(G_{\text{MC}};\boldsymbol{\phi})\right)$$

(n/l factors account for different fractions of S and B simulation data)

INFERNO details / 2

The Asimov likelihood we have written, $$\hat{\mathcal{L}}_A(\boldsymbol{\theta};\boldsymbol{\phi}) = \prod_{i=0}^{m} \mathrm{Pois}\left(\left(\frac{n}{l}\right)\hat{t}_i(G_{\mathrm{MC}};\boldsymbol{\phi}) \mid \left(\frac{n}{l}\right)\hat{t}_i(G_{\mathrm{MC}};\boldsymbol{\phi})\right)$$

is maximized by the value of simulation parameters θ_{MC} used to generate the data G_{MC}.

We may then take the second derivative, expanded in θ around θ_{MC}, of the Asimov likelihood and interpret it as the Fisher information matrix,

$$I(\boldsymbol{\theta})_{ij} = \mathbb{E}\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}\left(-\log\hat{\mathcal{L}}_A(\boldsymbol{\theta};\boldsymbol{\phi})\right)\right]$$

whose inverse, by the Cramer-Rao lower bound, is a lower limit of the covariance: we may then use it as an estimator of the variances of our parameters of interest in the loss function, *i.e.*

$$U = I^{-1}_{kk}(\boldsymbol{\theta}_{\mathrm{MC}})$$

INFERNO pseudo-code (*)

Algorithm 1 Inference-Aware Neural Optimisation.

Input 1: differentiable simulator or variational approximation $g(\boldsymbol{\theta})$.
Input 2: initial parameter values $\boldsymbol{\theta}_{\mathrm{MC}}$.
Input 3: parameter of interest $\omega_0 = \theta_k$.
Output: learned summary statistic $\boldsymbol{t}(D; \boldsymbol{\phi})$.

1: **for** $i = 1$ to N (number of SGD iterations) **do**
2: Sample a representative mini-batch G_{MC} from $g(\boldsymbol{\theta}_{\mathrm{MC}})$.
3: Compute differentiable summary statistic $\hat{\boldsymbol{t}}(G_{\mathrm{MC}}; \boldsymbol{\phi})$.
4: Construct Asimov likelihood $\mathcal{L}_A(\boldsymbol{\theta}, \boldsymbol{\phi})$.
5: Get information matrix inverse $I(\boldsymbol{\theta})^{-1} = \boldsymbol{H}_{\boldsymbol{\theta}}^{-1}(\log \mathcal{L}_A(\boldsymbol{\theta}, \boldsymbol{\phi}))$.
6: Obtain loss $U = I_{kk}^{-1}(\boldsymbol{\theta}_{\mathrm{MC}})$.
7: Update network parameters $\boldsymbol{\phi} \rightarrow \mathrm{SGD}(\nabla_{\boldsymbol{\phi}} U)$.
8: **end for**

INFERNO synthetic example

In [49] a simple example with 3 nuisances affecting the background of a 2-component mixture problem is considered:

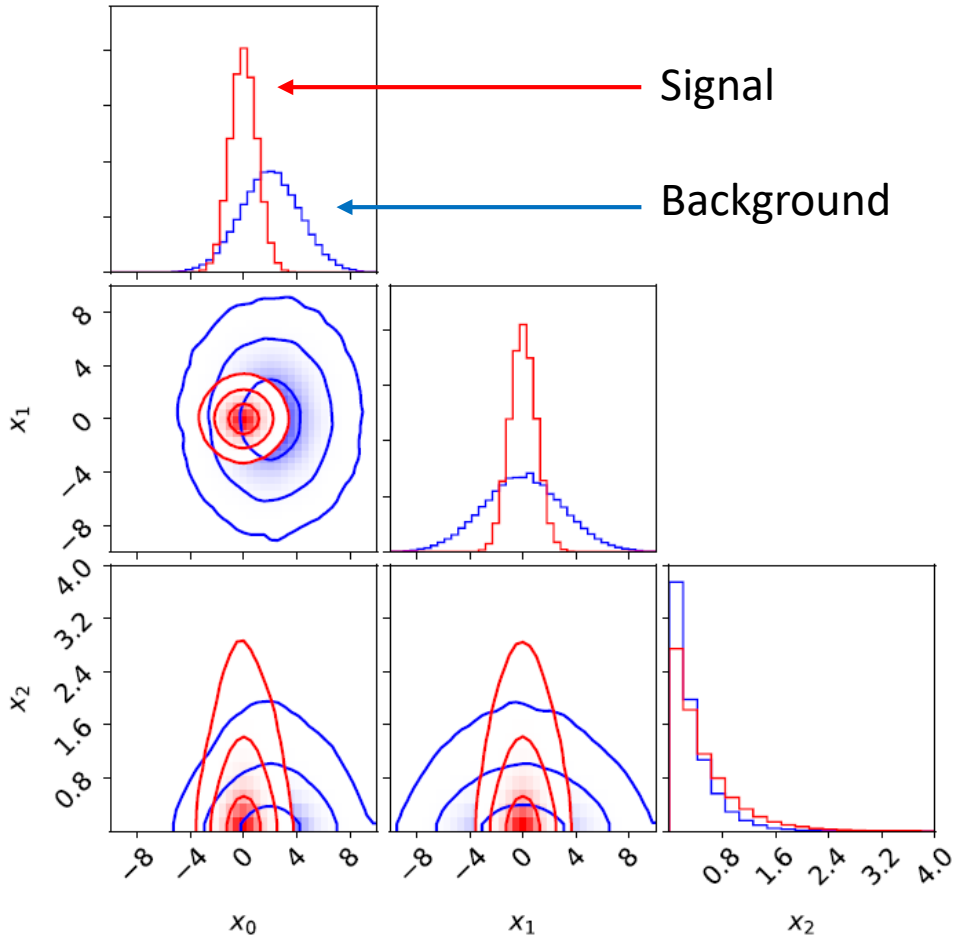
$$f_b(\boldsymbol{x}|r,\lambda) = \mathcal{N}\left((x_0,x_1)\,\middle|\,(2+r,0),\begin{bmatrix}5 & 0\\ 0 & 9\end{bmatrix}\right) Exp(x_2|\lambda)$$

$$f_s(\boldsymbol{x}) = \mathcal{N}\left((x_0,x_1)\,\middle|\,(1,1),\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\right) Exp(x_2|2)$$

r shifts the background mean, λ changes the slope, and b is background normalization.
The model is then

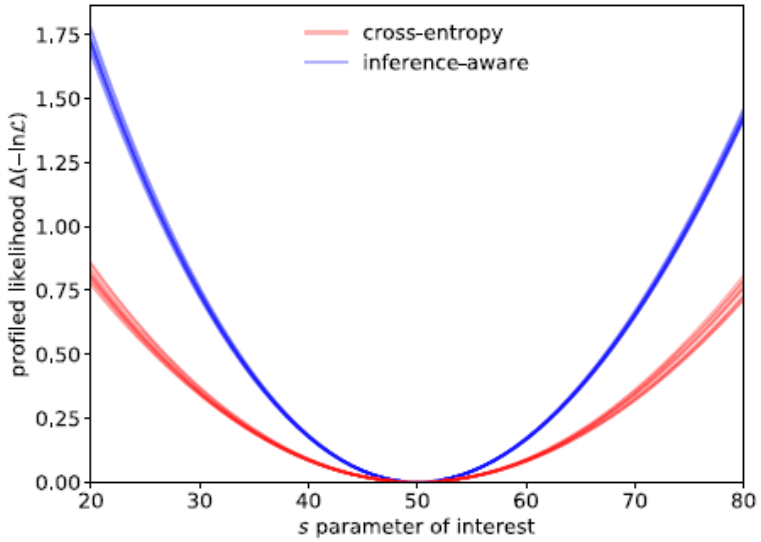
$$p(\boldsymbol{x}|\mu,r,\lambda) = (1-\mu)f_b(\boldsymbol{x}|r,\lambda) + \mu f_s(\boldsymbol{x})$$

and the optimization of the NN is tested with several benchmarks, releasing nuisances (see below)



INFERNO results

	Benchmark 0	Benchmark 1	Benchmark 2	Benchmark 3	Benchmark 4
Interest pars	1 (s)	1 (s)	1 (s)	1 (s)	1 (s)
Nuisance pars	0 (all fixed)	1 (r)	2 (r and λ)	2 (r and λ)	3 (r, λ and b)
r (bkg shift)	0.0 (fixed)	free (init 0.0)	free (init 0.0)	$\mathcal{N}(\lambda\|0.0, 0.4)$	$\mathcal{N}(\lambda\|0.0, 4.0)$
λ (bkg exp rate)	3.0 (fixed)	3.0 (fixed)	free (init 3.0)	$\mathcal{N}(\lambda\|3.0, 1.0)$	$\mathcal{N}(\lambda\|3.0, 1.0)$
b (bkg normalisation)	1000 (fixed)	1000 (fixed)	1000 (fixed)	1000 (fixed)	$\mathcal{N}(b\|1000, 100)$



INFERNO consistently outperforms the NN and has performance which approaches that of the analytical likelihood result.

	Benchmark 0	Benchmark 1	Benchmark 2	Benchmark 3	Benchmark 4
NN classifier	$14.99^{+0.02}_{-0.00}$	$18.94^{+0.11}_{-0.05}$	$23.94^{+0.52}_{-0.17}$	$21.54^{+0.27}_{-0.05}$	$26.71^{+0.56}_{-0.11}$
INFERNO 0	$\mathbf{15.51^{+0.09}_{-0.02}}$	$18.34^{+5.17}_{-0.51}$	$23.24^{+6.54}_{-1.22}$	$21.38^{+3.15}_{-0.69}$	$26.38^{+7.63}_{-1.36}$
INFERNO 1	$15.80^{+0.14}_{-0.04}$	$\mathbf{16.79^{+0.17}_{-0.05}}$	$21.41^{+2.00}_{-0.53}$	$20.29^{+1.20}_{-0.39}$	$24.26^{+2.35}_{-0.71}$
INFERNO 2	$15.71^{+0.15}_{-0.04}$	$16.87^{+0.19}_{-0.06}$	$\mathbf{16.95^{+0.18}_{-0.04}}$	$16.88^{+0.17}_{-0.03}$	$18.67^{+0.25}_{-0.05}$
INFERNO 3	$15.70^{+0.21}_{-0.04}$	$16.91^{+0.20}_{-0.05}$	$16.97^{+0.21}_{-0.04}$	$\mathbf{16.89^{+0.18}_{-0.03}}$	$18.69^{+0.27}_{-0.04}$
INFERNO 4	$15.71^{+0.32}_{-0.06}$	$16.89^{+0.30}_{-0.07}$	$16.95^{+0.38}_{-0.05}$	$16.88^{+0.40}_{-0.05}$	$\mathbf{18.68^{+0.58}_{-0.07}}$
Optimal classifier	14.97	19.12	24.93	22.13	27.98
Analytical likelihood	14.71	15.52	15.65	15.62	16.89

INFERNO challenges and status

The structure of INFERNO is complex, but the minimization of the loss is relatively straightforward

Main issue: how to model HEP nuisances and effect on observations: must *e.g.* transform input features, interpolating simulated observation weights, or interpolate histogram counts (last ditch).

An application to a real HEP analysis is underway through the work of Lukas Layer (INFN-PD) on CMS open data (a Run 1 top cross section measurement)

Recent developments

Two recent works have built on the idea of INFERNO for HEP applications.

1. Wunsch *et al.*[52] use a single-output NN to construct a Poisson-count likelihood instead of a softmax, and make the histogram differentiable by smoothing it with a Gaussian kernel.
2. Heinrich and Simpson[53] use "fixed-point differentiation" to compute gradients of a profile likelihood, aiming at directly minimize the expected upper limits on sought processes with CLs. Also in their work (NEOS) the modelling of the nuisances is restricted to histogram interpolation.

In addition there have been

- a proposal to use the AMS in a single bin counting experiment including a single systematic in the loss function[54]
- A variation of BDT training (QBDT) targets directly signal significance with an approximate model of nuisances[55].

The field is in rapid evolution and new ideas are possible. The bottomline is that if one can realign the MVA target to be the final desired goal, results will be close to optimal, in the sense of maximizing the use of the available information.

8. Summary

A wide arsenal of techniques that try to remove the impact of systematic uncertainties in supervised classification for HEP problems has been developed in recent years

- The focus in many cases is achieving a decorrelation of salient features (jet mass), to maximize discovery significance
  - Here, several methods successfully achieve the desired goal, with minor performance loss

The real issue is however how to minimize the effect of systematic uncertainties whatever their origin, with tools of more general applicability

- Important steps have been made but the topic is still an active area of research in ML

Thank you for your attention!

References

[1] P. De Castro Manzano. Statistical Learning and Inference at Particle Collider Experiments. PhD thesis (2019). URL https://cds.cern.ch/record/2701341.

[2] K. Cranmer, G. Lewis, L. Moneta, A. Shibata, and W. Verkerke, HistFactory: A tool for creating statistical models for use with RooFit and RooStats (June 2012).

[3] K. Cranmer. Practical Statistics for the LHC. In 2011 European School of High-Energy Physics, pp. 267{308 (2014). doi: 10.5170/CERN-2014-003.267.

[4] L. Lista. Practical Statistics for Particle Physicists. In 2016 European School of High-Energy Physics, pp. 213{258 (2017). doi: 10.23730/CYRSP-2017-005.213.

[5] W. M. Pateeld, On the maximized likelihood function, Sankhya: The Indian Journal of Statistics, Series B (1960-2002). 39(1), 92{96 (1977). URL http://www.jstor.org/stable/25052054.

[6] D. R. Cox and O. E. Barndor-Nielsen, Inference and Asymptotics. CRC Press (Mar., 1994). URL https://play.google.com/store/books/details?id=KxYeBQAAQBAJ.

[7] F. James and M. Roos, Minuit - a system for function minimization and analysis of the parameter errors and correlations, Comput. Phys. Commun. 10 (6), 343{367 (Dec., 1975). URL http://www.sciencedirect.com/science/article/pii/0010465575900399.

[8] G. Cowan, K. Cranmer, E. Gross, and O. Vitells, Asymptotic formulae for likelihood-based tests of new physics, Eur. Phys. J. C. 71, 1554 (2011). doi: 10.1140/epjc/s10052-011-1554-0. [Erratum: Eur. Phys. J. C73, 2501 (2013)].

[9] J. Thaler and K. Van Tilburg, Maximizing Boosted Top Identication by Minimizing N-subjettiness, JHEP. 02, 093 (2012). doi: 10.1007/JHEP02(2012)093.

References / 2

[10] J. Dolen, P. Harris, S. Marzani, S. Rappoccio, and N. Tran, Thinking outside the ROCs: Designing Decorrelated Taggers (DDT) for jet substructure, JHEP. 05, 156 (2016). https://doi.org/10.1007/JHEP05(2016)156.

[11] I. Moult, B. Nachman, and D. Neill, Convolved Substructure: Analytically Decorrelating Jet Substructure Observables, JHEP. 05, 002 (2018). doi: 10.1007/JHEP05(2018)002.

[12] R. M. Neal, Computing likelihood functions for high-energy physics experiments when distributions are defined by simulators with nuisance parameters (2008). URL http://cds.cern.ch/record/1099977.

[13] T. Aaltonen et al., Evidence for a particle produced in association with weak bosons and decaying to a bottom-antibottom quark pair in Higgs boson searches at the Tevatron, Phys. Rev. Lett. 109, 071804 (2012). doi: 10.1103/PhysRevLett.109.071804.

[14] S. Chatrchyan et al., Combined results of searches for the standard model Higgs boson in pp collisions at sqrt(s) = 7 TeV, Phys. Lett. B. 710, 26{48 (2012). doi: 10.1016/j.physletb.2012.02.064.

[15] P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson, Parameterized neural networks for high-energy physics, Eur. Phys. J. C. 76(5), 235 (2016). doi: 10.1140/epjc/s10052-016-4099-4.

[16] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaitre,A. Mertens, and M. Selvaggi, DELPHES 3, A modular framework for fast simulation of a generic collider experiment, JHEP. 02, 057 (2014). doi: 10.1007/JHEP02(2014)057.

[17] K. Cranmer, J. Pavez, and G. Louppe, Approximating likelihood ratios with calibrated discriminative classifiers (June, 2015). URL http://arxiv.org/abs/1506.02169.

[18] J. Aguilar-Saavedra, J. H. Collins, and R. K. Mishra, A generic anti-QCD jet tagger, JHEP. 11, 163 (2017). doi: 10.1007/JHEP11(2017)163.

References / 3

[19] S. Chang, T. Cohen, and B. Ostdiek, What is the Machine Learning?, Phys. Rev. D. 97(5), 056009 (2018). doi: 10.1103/PhysRevD.97.056009.

[20] K. Datta and A. Larkoski, How Much Information is in a Jet?, JHEP. 06,073 (2017). doi: 10.1007/JHEP06(2017)073.

[21] R. Dalitz, On the analysis of tau-meson data and the nature of the tau-meson, Phil. Mag. Ser. 7. 44, 1068{1080 (1953). doi: 10.1080/14786441008520365.

[22] J. Stevens and M. Williams, uBoost: A boosting method for producing uniform selection efficiencies from multivariate classifiers, JINST. 8, P12013 (2013). doi: 10.1088/1748-0221/8/12/P12013.

[23] Y. Freund and R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences. 55(1), 119{139 (Aug., 1997). doi: 10.1006/jcss.1997.1504.

[24] A. Rogozhnikov, A. Bukva, V. Gligorov, A. Ustyuzhanin, and M. Williams, New approaches for boosting to uniformity, JINST. 10(03), T03002 (2015). doi: 10.1088/1748-0221/10/03/T03002.

[25] G. Kasieczka and D. Shih, DisCo Fever: Robust Networks Through Distance Correlation, arXiv:2001.05310 (Jan 2020).

[26] S. Wunsch, S. J□orger, R. Wolf, and G. Quast, Reducing the dependence of the neural network function to systematic uncertainties in the input space (July, 2019). URL http://arxiv.org/abs/1907.11674.

[27] C. Adam-Bourdarios, G. Cowan, C. Germain-Renaud, I. Guyon, B. Kegl, and D. Rousseau, The Higgs Machine Learning Challenge, J. Phys. Conf. Ser. 664(7), 072015 (2015). doi: 10.1088/1742-6596/664/7/072015.

[28] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In eds. B. Scholkopf, J. C. Platt, and T. Homan, Advances in Neural Information Processing Systems 19, pp. 137-144. MIT Press (2007). URL http://papers.nips.cc/paper/2983-analysis-of-representations-for-domain-adaptation.pdf.

References / 4

[29] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, A theory of learning from different domains, Machine Learning. 79 (1-2), 151{175 (Oct., 2009). URL https://doi.org/10.1007/s10994-009-5152-4.

[30] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand, Domain-adversarial neural networks, arXiv:1412.4446 (2014).

[31] G. Louppe, M. Kagan, and K. Cranmer. Learning to pivot with adversarial networks. In eds. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Advances in Neural Information Processing Systems 30, pp. 981{990. Curran Associates, Inc. (2017). URL http://papers.nips.cc/paper/6699-learning-to-pivot-with-adversarial-networks.pdf.

[32] M. H. Degroot and M. J. Schervish. Probability and statistics, Carnegie-Mellon Univ., 1977.

[33] C. Shimmin, P. Sadowski, P. Baldi, E. Weik, D. Whiteson, E. Goul, and A. Sgaard, Decorrelated Jet Substructure Tagging using Adversarial Neural Networks, Phys. Rev. D. 96(7), 074034 (2017). doi: 10.1103/PhysRevD.96. 074034.

[34] V. Estrade, C. Germain, I. Guyon, and D. Rousseau. Adversarial learning to eliminate systematic errors: a case study in high energy physics. In NIPS 2017 (2017).

[35] P. Simard, B. Victorri, Y. LeCun, and J. Denker. Tangent prop - a formalism for specifying selected invariances in an adaptive network. In eds. J. E. Moody, S. J. Hanson, and R. P. Lippmann, Advances in Neural Information Processing Systems 4, pp. 895{ 903. Morgan-Kaufmann (1992). URL http://papers.nips.cc/paper/536-tangent-prop-a-formalism-for-specifying-selected-invariances-in-an-adaptive-network.pdf.

[36] A. Blance, M. Spannowsky, and P. Waite, Adversarially-trained autoencoders for robust unsupervised new physics searches, JHEP. 10, 047 (2019). doi: 10.1007/JHEP10(2019)047.

References / 5

[37] C. Englert, P. Galler, P. Harris, and M. Spannowsky, Machine Learning Uncertainties with Adversarial Neural Networks, Eur. Phys. J. C. 79(1), 4 (2019). doi: 10.1140/epjc/s10052-018-6511-8.

[38] L. M. Dery, B. Nachman, F. Rubbo, and A. Schwartzman, Weakly supervised classification in high energy physics, J. High Energy Phys. 2017(5), 145 (May, 2017). URL https://doi.org/10.1007/JHEP05(2017)145.

[39] E. M. Metodiev, B. Nachman, and J. Thaler, Classication without labels: learning from mixed samples in high energy physics, J. High Energy Phys. 2017(10), 174 (Oct., 2017). URL https://doi.org/10.1007/JHEP10(2017) 174.

[40] T. Cohen, M. Freytsis, and B. Ostdiek, (machine) learning to do more with less, J. High Energy Phys. 2018(2), 34 (Feb., 2018). URL https://doi.org/10.1007/JHEP02(2018)034.

[41] P. T. Komiske, E. M. Metodiev, B. Nachman, and M. D. Schwartz, Learning to classify from impure samples with high-dimensional data, Phys. Rev. D. 98(1), 011502 (July, 2018). URL https://link.aps.org/doi/10.1103/ PhysRevD.98.011502.

[42] K. Cranmer, J. Brehmer, and G. Louppe, The frontier of simulation-based inference, arXiv:1911.01429 (Nov. 2019).

[43] J. Neyman and E. S. Pearson, On the problem of the most efficient testsof statistical hypotheses, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character. 231, 289{337 (1933). ISSN 02643952. URL http://www.jstor.org/stable/91247.

[44] J. Brehmer, G. Louppe, J. Pavez, and K. Cranmer, Mining gold from implicit models to improve likelihood-free inference, arXiv:1805.12244 (2018).

[45] J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez, Constraining Effective Field Theories with Machine Learning, arXiv:1805.00013 (2018).

[46] J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez, A Guide to Constraining Effective Field Theories with Machine Learning, Phys. Rev. D 98, 052004 (2018), DOI: 10.1103/PhysRevD.98.052004.

References / 6

[47] M. Stoye, J. Brehmer, G. Louppe, J. Pavez, and K. Cranmer, Likelihood-free inference with an improved cross-entropy estimator, arXiv:1808.00973 (2018).

[48] J. Brehmer, F. Kling, I. Espejo, and K. Cranmer, MadMiner: Machine learning-based inference for particle physics, Comput. Softw. Big Sci. 4(1), 3 (2020). doi: 10.1007/s41781-020-0035-2.

[49] P. de Castro and T. Dorigo, INFERNO: Inference-Aware neural optimisation, Comput. Phys. Commun. 244, 170-179 (Nov., 2019). URL http://www.sciencedirect.com/science/article/pii/S0010465519301948.

[50] T. Charnock, G. Lavaux, and B. D. Wandelt, Automatic physical inference with information-maximizing neural networks, Phys. Rev. D. 97(8), 083004 (Apr., 2018). URL https://link.aps.org/doi/10.1103/PhysRevD.97.083004.

[51] J. Alsing and B. Wandelt, Nuisance hardened data compression for fast likelihood-free inference, Mon. Not. R. Astron. Soc. 488(4), 5093{5103 (Oct., 2019). URL https://academic.oup.com/mnras/article-abstract/488/4/5093/5530778.

[52] S.Wunsch, S. J□orger, R.Wolf, and G. Quast, Optimal statistical inference in the presence of systematic uncertainties using neural network optimization based on binned poisson likelihoods with nuisance parameters, arXiv:2003.07186 (Mar., 2020). URL http://arxiv.org/abs/2003.07186.

[53] L. Heinrich and N. Simpson. pyhf/neos: initial zenodo release, URL https://doi.org/10.5281/zenodo.3697981 (Mar., 2020).

[54] A. Elwood and D. Kr□ucker, Direct optimisation of the discovery significance when training neural networks to search for new physics in particle colliders, arXiv:1806.00322 (June, 2018). URL http://arxiv.org/abs/1806.00322.

[55] L.-G. Xia, QBDT, a new boosting decision tree method with systematic uncertainties into training for high energy physics, arXiv:1810.08387 (Oct., 2018). URL http://arxiv.org/abs/1810.08387.

References / 7

[56] CMS collaboration, internal.

[57] ATLAS collaboration, Dijet resonance search with weak supervision using sqrt(s)=13 TeV pp collisions in the ATLAS detector, arXiv:2005.02983 (hep-ex), 2020.

[58] F. de Almeida Dias, talk at Anomaly detection mini-workshop, July 16th 2020.

[59] P. de Castro Manzano and T. Dorigo, Dealing with nuisance parameters using machine learning in high-energy physics: a review", https://arxiv.org/abs/2007.09121 (2020).

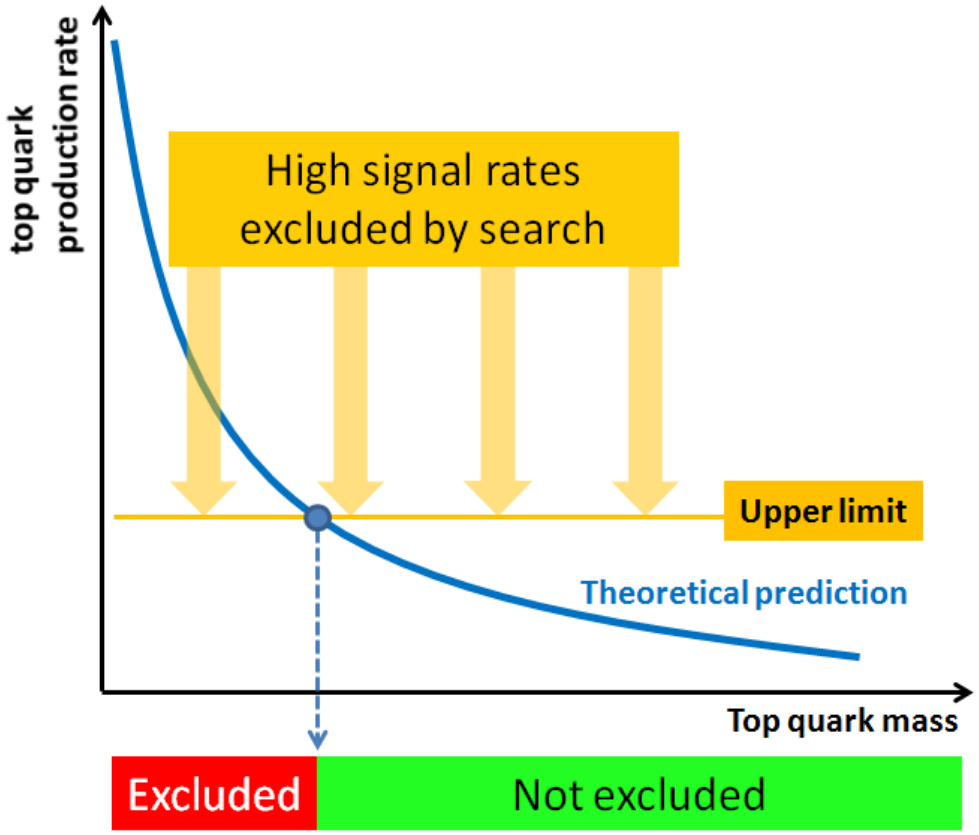
Backup

And what if there is no signal ?

If we do not see a signal we can **exclude the new physics model**

- More often the model is composite, so we exclude a range of values of the relevant nuisance parameter
  - Often this is, again, the mass of the particle

- We may, *e.g.*, derive lower limits on the particle mass from upper limits on the signal strength, by comparing those to a theoretical model



Luckily, even a lower mass limit is useful information, **worth a publication!**

Neal's solution (*)

In [12] Radford Neal proposes a recipe to construct summaries for both the nuisances and the observed features, using *e.g.* a neural network.

One trains the NN to discriminate S from B given values from the nuisances α, taken from their prior. If the NN is constructed to reduce the dimensionality of the inputs, the probability of signal and background as seen by the classifier can then be written

$$P(S|x,\alpha) = F_S(f_1(\alpha),f_2(x))$$
$$P(B|x,\alpha) = F_B(f_1(\alpha),f_2(x))$$

where $S_1 = f_1(\alpha)$, $S_2 = f_2(x)$ are the **two summarized forms of nuisances and event features**.

If the above P can be parametrized, and the mappings f_1, f_2 do not lose too much information in the compression, one may obtain **approximate sufficiency**. What is needed is a regression model r(x) that approximates the nuisances α given data x. One may then obtain $P(S,B) = F_{S,B}(f_1(r(x)), f_2(x))$ if the models are good enough, and use it for classification.

This scheme is untested, but has become the basis of a wide range of studies of likelihood-free inference methods. We discuss them later.

The «Learning to Pivot» algorithm (*)

Algorithm 1 Adversarial training of a classifier f against an adversary r.

Inputs: training data $\{x_i, y_i, z_i\}_{i=1}^N$; *Outputs:* $\hat{\theta}_f, \hat{\theta}_r$.

1: **for** $t = 1$ to T **do**
2: **for** $k = 1$ to K **do**
3: Sample minibatch $\{x_m, z_m, s_m = f(x_m; \theta_f)\}_{m=1}^M$ of size M;
4: With θ_f fixed, update r by ascending its stochastic gradient $\nabla_{\theta_r} E(\theta_f, \theta_r) :=$

$$\nabla_{\theta_r} \sum_{m=1}^{M} \log p_{\theta_r}(z_m|s_m);$$

5: **end for**
6: Sample minibatch $\{x_m, y_m, z_m, s_m = f(x_m; \theta_f)\}_{m=1}^M$ of size M;
7: With θ_r fixed, update f by descending its stochastic gradient $\nabla_{\theta_f} E(\theta_f, \theta_r) :=$

$$\nabla_{\theta_f} \sum_{m=1}^{M} \left[-\log p_{\theta_f}(y_m|x_m) + \log p_{\theta_r}(z_m|s_m) \right],$$

where $p_{\theta_f}(y_m|x_m)$ denotes $\mathbf{1}(y_m = 0)(1 - s_m) + \mathbf{1}(y_m = 1)s_m$;
8: **end for**

CWoLa applications to LHC data (*)

Two recent applications of CWoLa:

1. CMS used it in a recent ttbb measurement[56]
2. ATLAS used in a search for resonances A→BC in dijets[57], where the plane of two fat-jet masses is scanned by weak supervised NN learning where training data are extracted from sidebands in M_{jj} in 8 bins (right, figure from F. de Almeida Dias talk at Anomaly detection mini-workshop[58])