ICNFP IX, Kolympari (Crete)

Can Machine Learning Rid us of Systematic Uncertainties ?

Tommaso Dorigo

INFN, Padova

September 5 2020

Foreword

We will discuss the impact of nuisance parameters on HEP analyses and how to reduce it by focusing on supervised classification, which is by far the most common use case.

The contents match well with those of Chapter 7.2 of a book on ML for HEP, which will be published by World Scientific toward the end of the year.

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 as arXiv:2007.09121[stat.ML].

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

Contents

1. Problem statement
2. ~~Nuisance parameters in statistical inference~~
3. Toward fully sufficient statistic summaries
4. Nuisance-parametrized models
5. Feature decorrelation, penalized methods, and adversary losses
6. ~~Semi-supervised approaches~~
7. Inference-aware approaches
8. Summary

1. Problem statement

Supervised classification is used to construct low-dimensional event summaries: we may call them with their name, summary statistics

  - Summary statistics can be employed to carry out statistical inference on parameters of interest **θ**

- E.g. we may use a NN to reduce features **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 or a proper surrogate 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

2. Nuisance parameters in statistical inference

It is useful to recall how nuisance parameters are “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 where nuisance parameters are free and unknown.

We solve the measurement problem by constructing estimators using 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,

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

αnd 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 there is the rub – that is not always given. But even in that case, of course the 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 implies that the summary statistic is **not 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

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|θ), can be factorized as

$$p\,(x|\theta) = q\,(x) * r\,(s(x)|\theta)$$

where q is a non-negative function not depending 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 is a sufficient statistic – but it is not a meaningful summary! (it does not reduce the dimensionality).

If we do not know p(x) analytically, we cannot solve the problem analytically! However, in 2-mixture models where the signal fraction is the only parameter, the density ratio 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 have never read the word “optimize” declined somehow in a physics article?

To those of you who raised their hand, or thought they probably should but didn’t: 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 the evidence in support of an optimization task is offered as the integral of the Receiver Output Characteristic (ROC) curve, or on signal acceptance at fixed purity, e.g.

→ Those named above are reasonably good proxies to the measurement precision: their maxima somehow 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

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?"

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

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

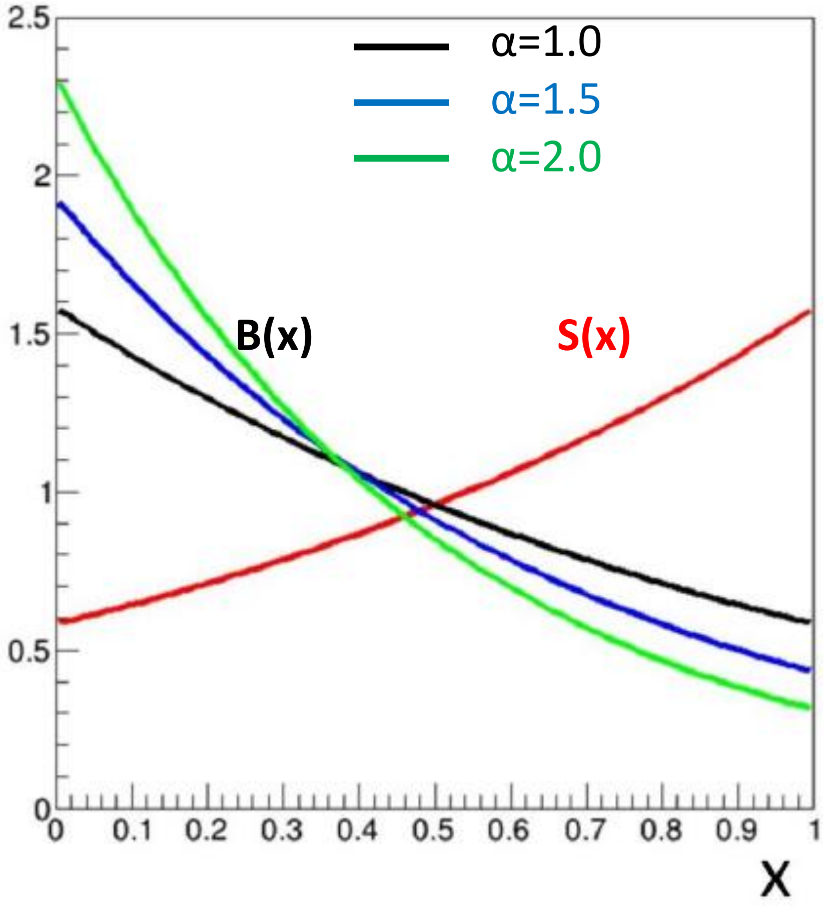
and from them we may derive an expression for the **ROC curve**, defined as the functional dependence of TPR on FPR:

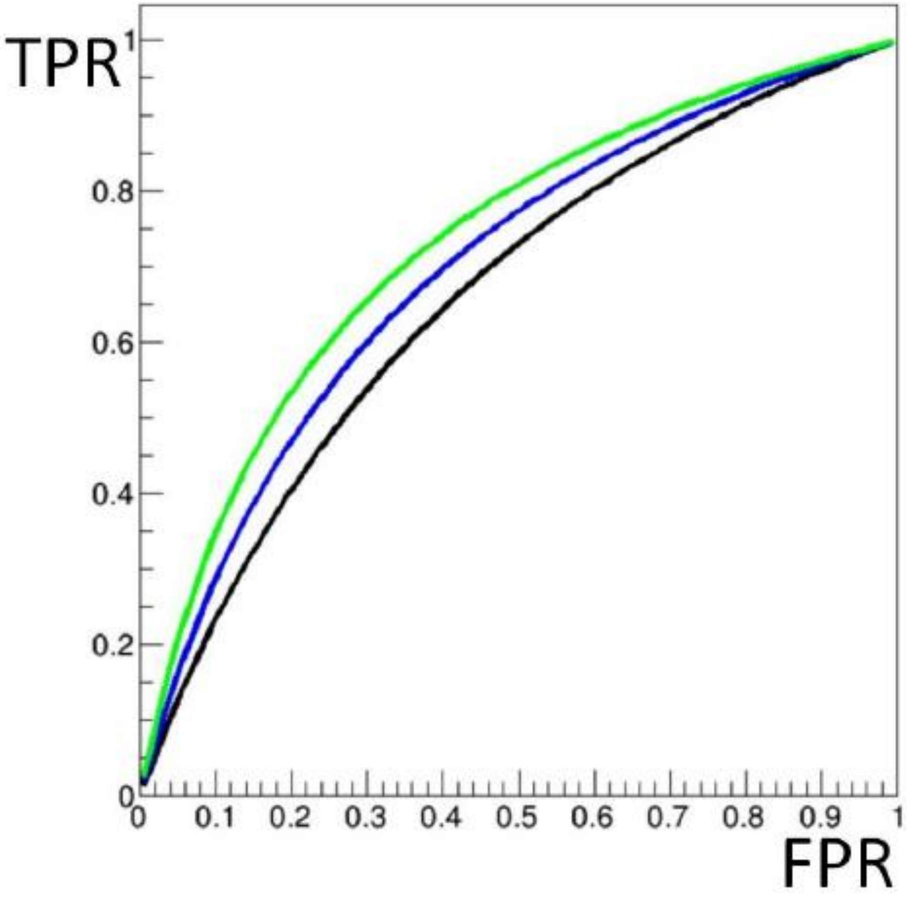
$$TPR(FPR) = \frac{e - \left[e^{-\alpha} + (1 + e^{-\alpha})FPR\right]^{-\frac{1}{\alpha}}}{e-1}$$

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?

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 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 value of TPR.

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, 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?

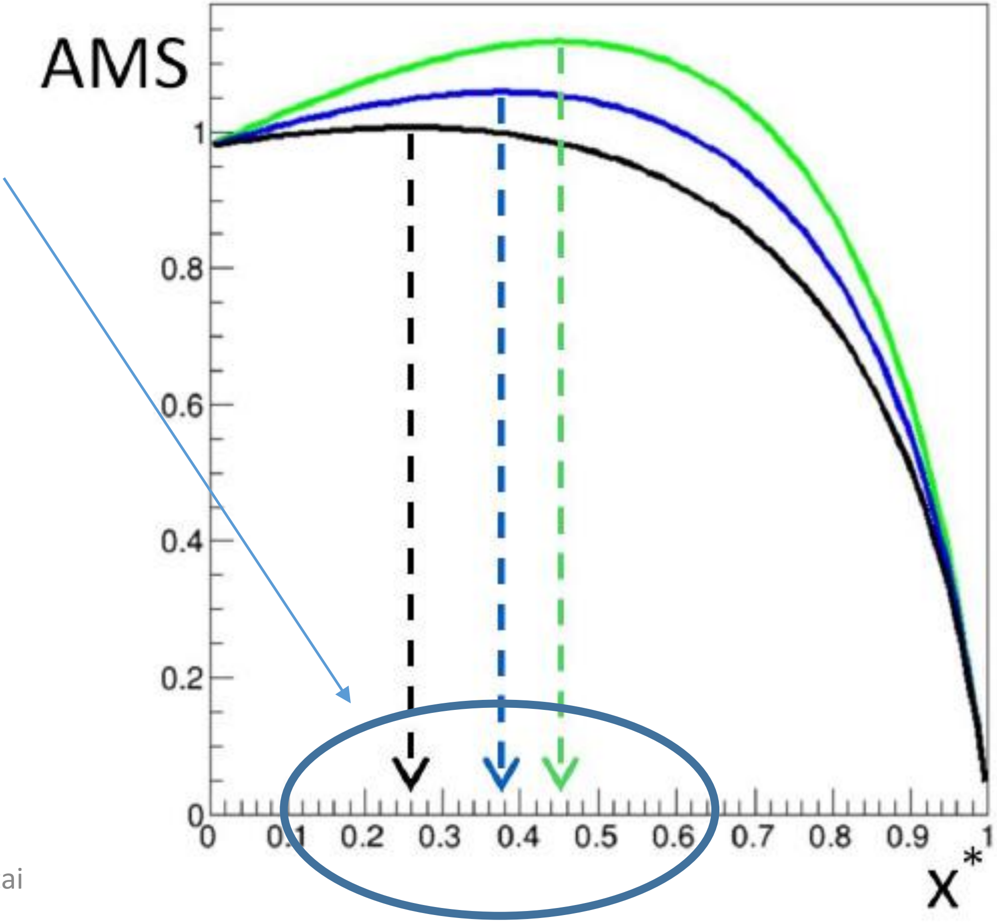
The AMS computed for the three exemplary values of α is shown on the right. As we fully expected, the values reached when α is larger are higher.

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. Maximizing the AMS (even with no nuisances) may not be the correct way to optimize for an upper limit on Ns, e.g.



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

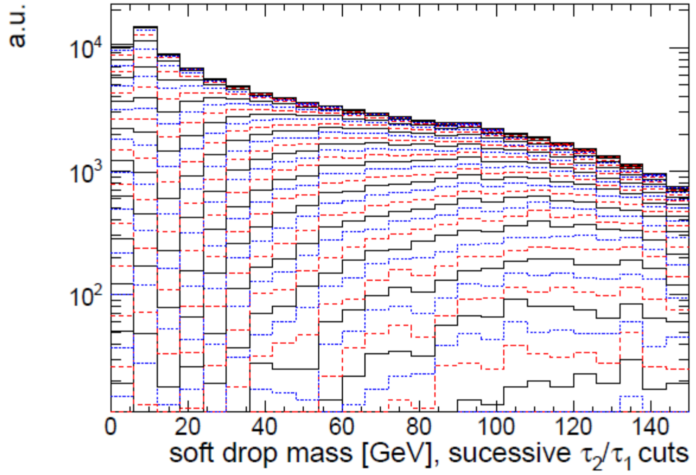
Example: N-subjettiness in boosted decays

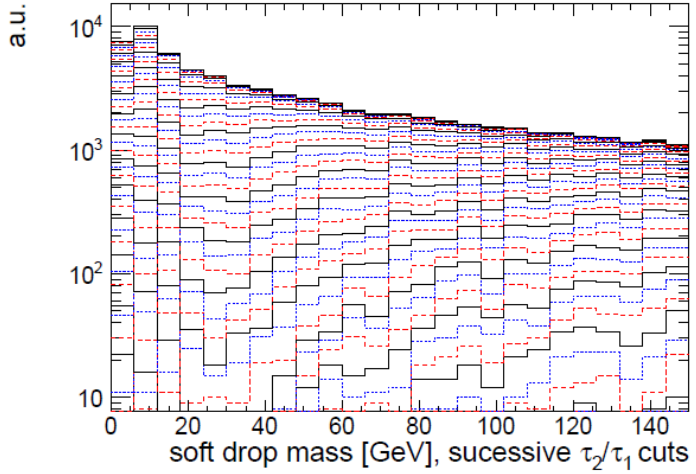
A number of observable features of fat energetic jets have been constructed to separate the hadronic 2- and 3-prong decay of heavy objects (W,Z,H,t) from QCD jets.

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

The "soft-drop" mass M of 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

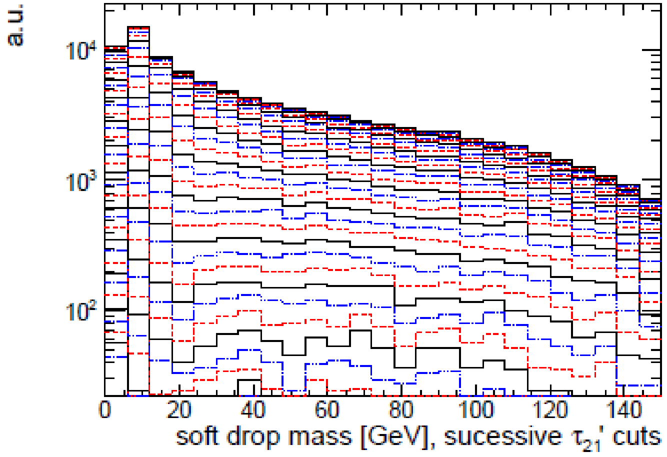


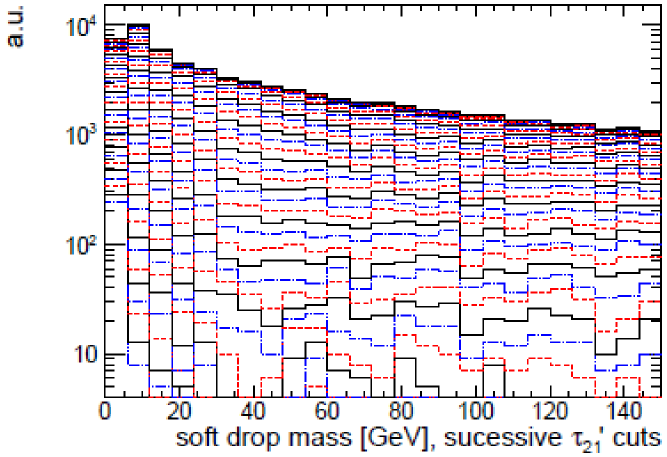


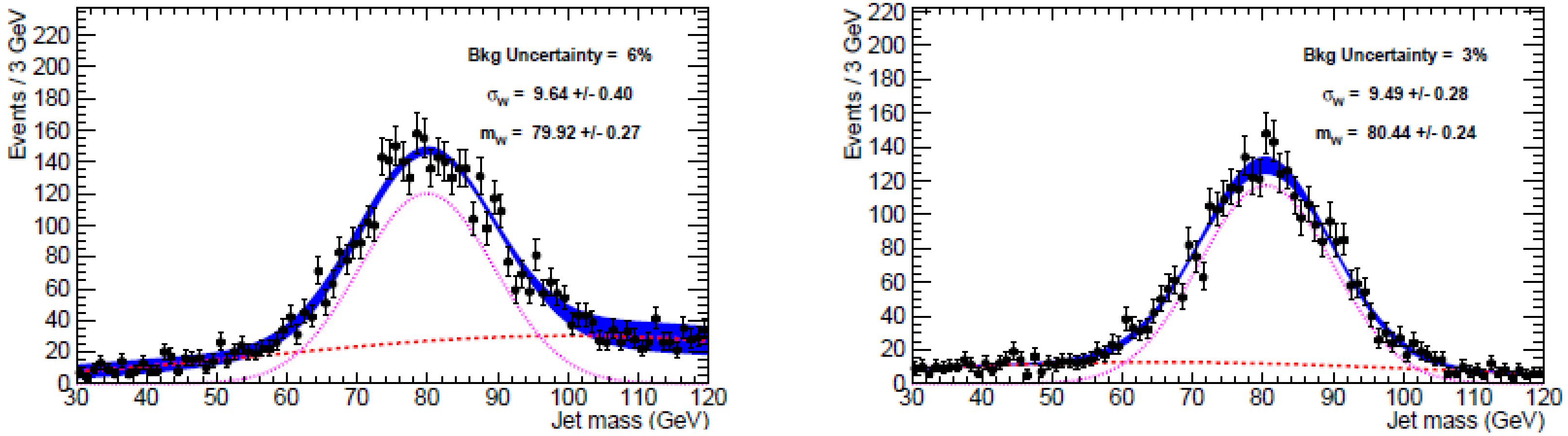
Correcting for the nuisance

- Define $\rho = \log(m^2/p_T^2)$, an appropriate scaling variable for QCD jets
- Compute average of τ_{21} as $f(\rho)$: it shows linear behaviour
- Define $\rho' = \rho + \log(p_T/\mathrm{GeV})$; 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 selection that preserves ability to use sidebands, etc.







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

What if we have no prior ?

The 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 many classifiers using data simulated at different M values, but the solution is sub-optimal (1/N use of total available data)

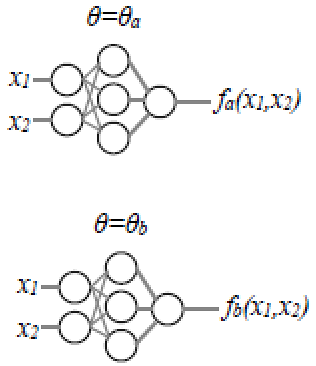
Better solution: parametrize the effect of M in the classifier [15]. The training data may be constructed as a mixture of different M hypotheses if M is included among the features.

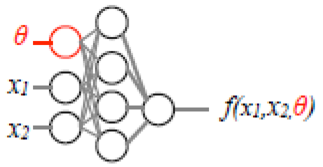
→ note that one must decide what to do with the background (for which M is undefined).

→ 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.

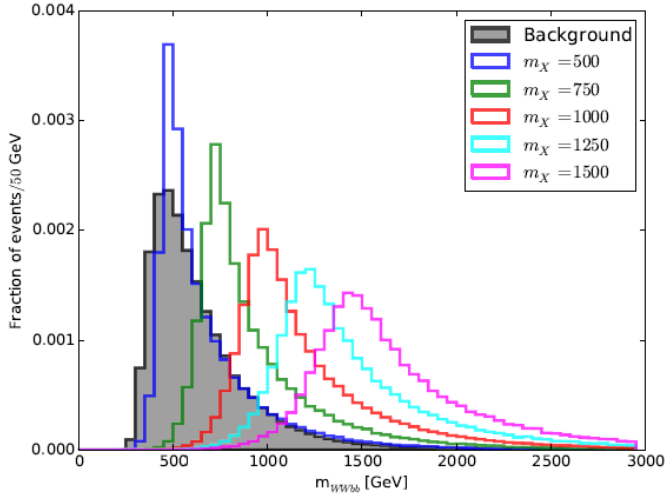
Example: X→tt in ATLAS





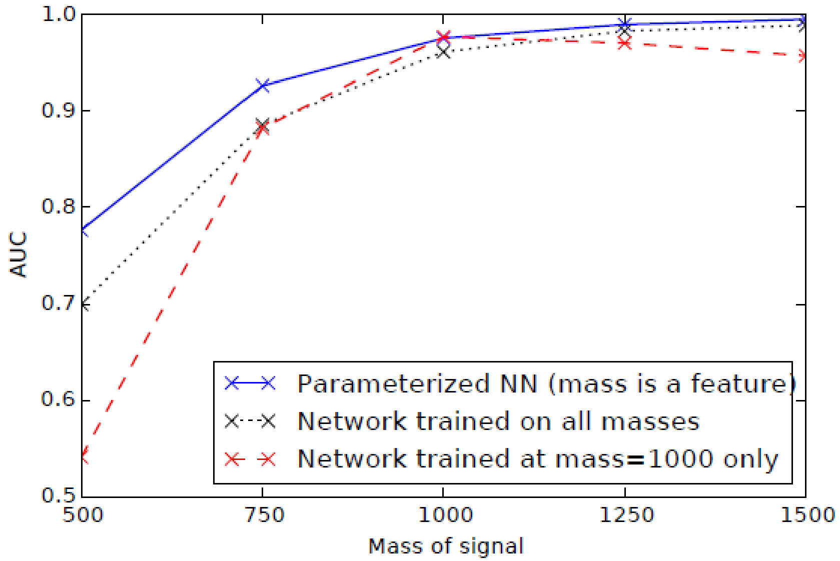
The technique proposed by Baldi et al. in[15] was tested with DELPHES[16] simulations, searching for a ttbar resonance within non-resonant ttbar backgrounds.

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.



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:

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

There are a number of recent algorithms using each of the above approaches. In what follows we look at one representative example for each these methods

Mass decorrelation

The most important use case of the first technique addresses the issue already discussed – keeping the background mass distribution unbiased.

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 of course this is sub-optimal.

Better strategy: 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.

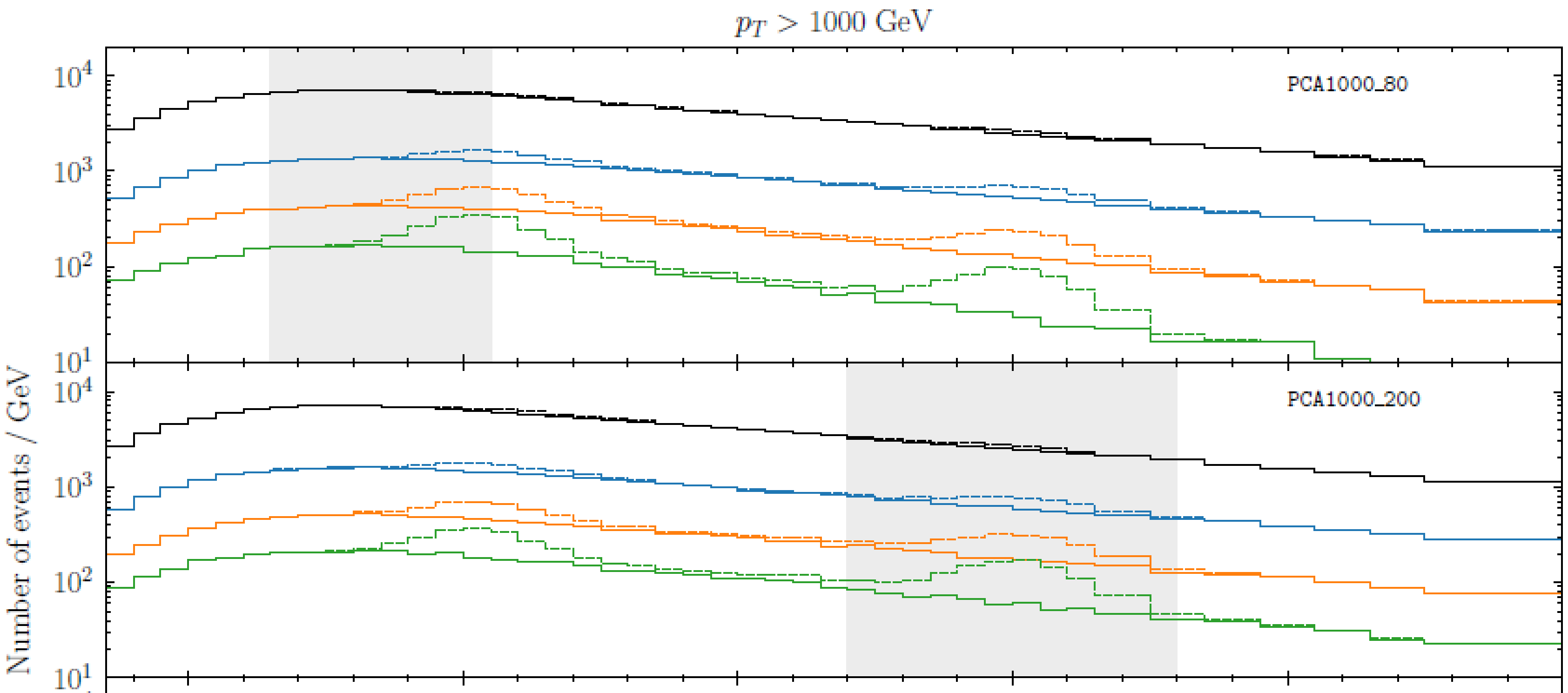
Mass decorrelation in boosting, reprise

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)



Penalizing the loss: 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

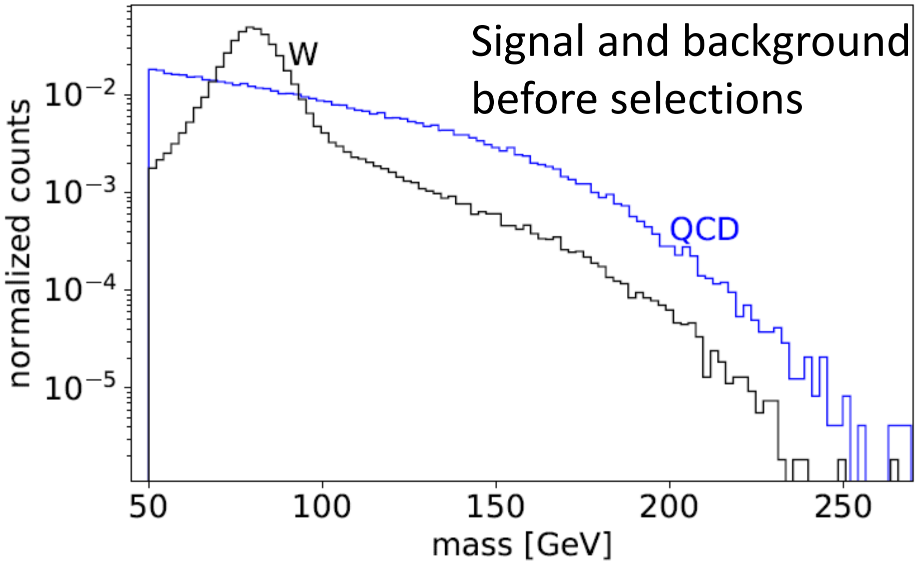
$$dCorr^2(X,Y) = \frac{dCov^2(X,Y)}{dCov(X,Y)dCov(Y,Y)}$$

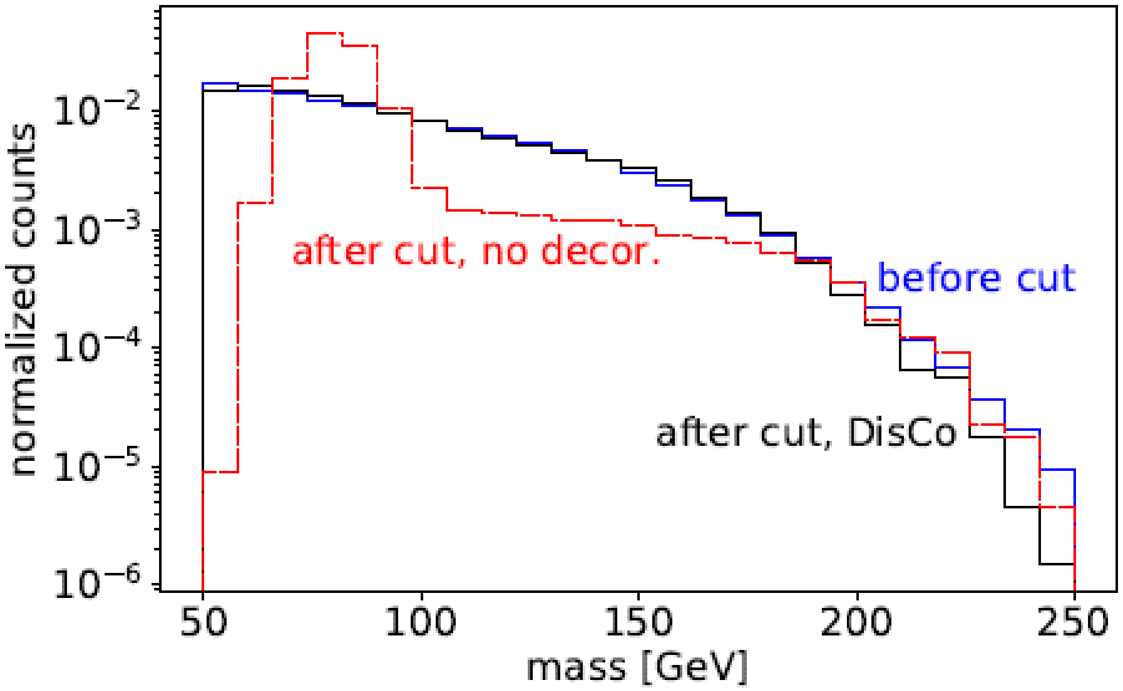
is then a [0,1] measure, null only if x,y are fully independent. Crucially, it is differentiable and computable with data samples, so it can be included in the loss function (for label y and mass m) with a penalty regularization factor λ

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

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).


Signal and background before selections

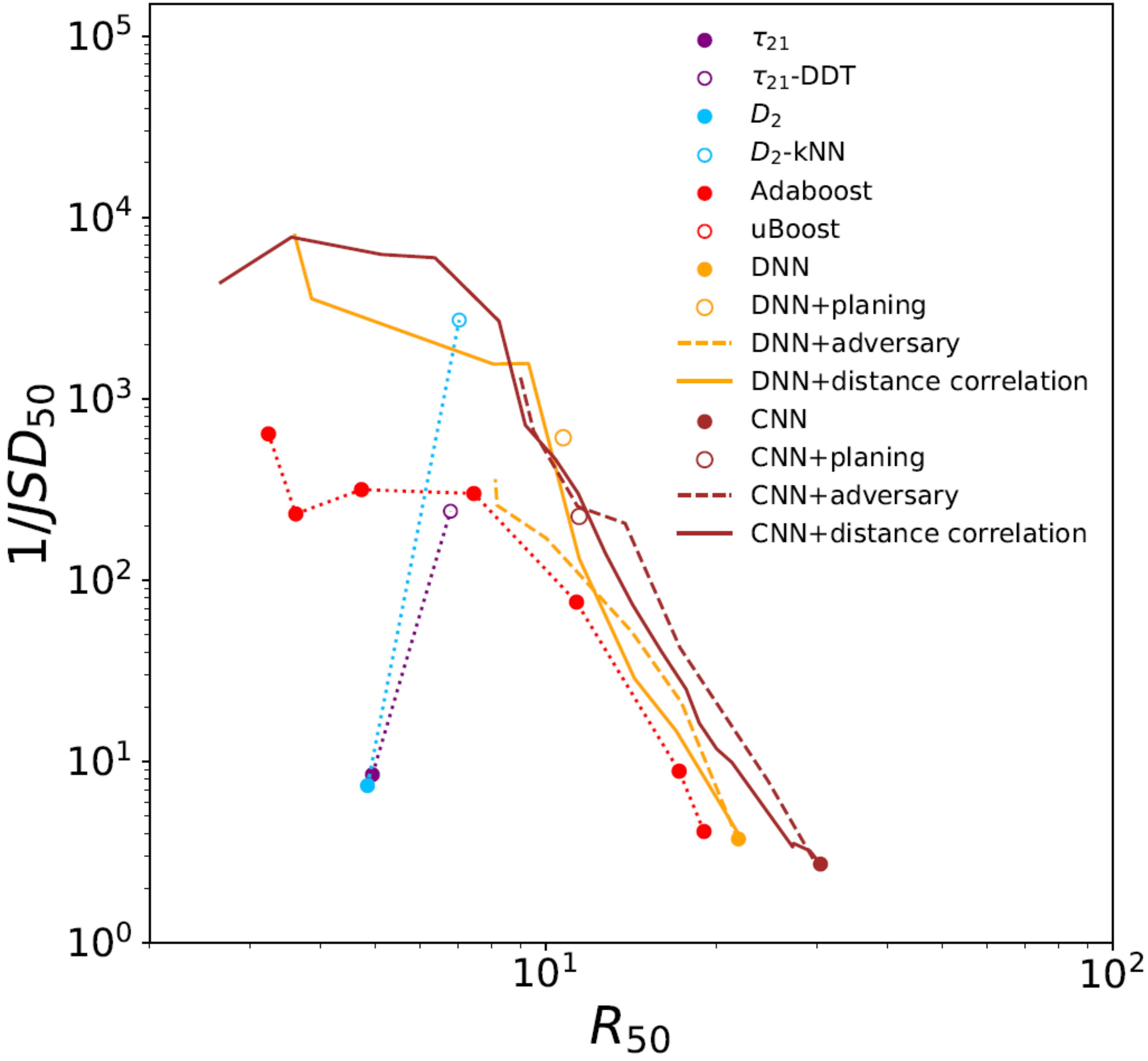


Comparisons

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.

Adversarial solutions

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

This 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].

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.

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;θ_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;θ_f), and the loss is defined globally as

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

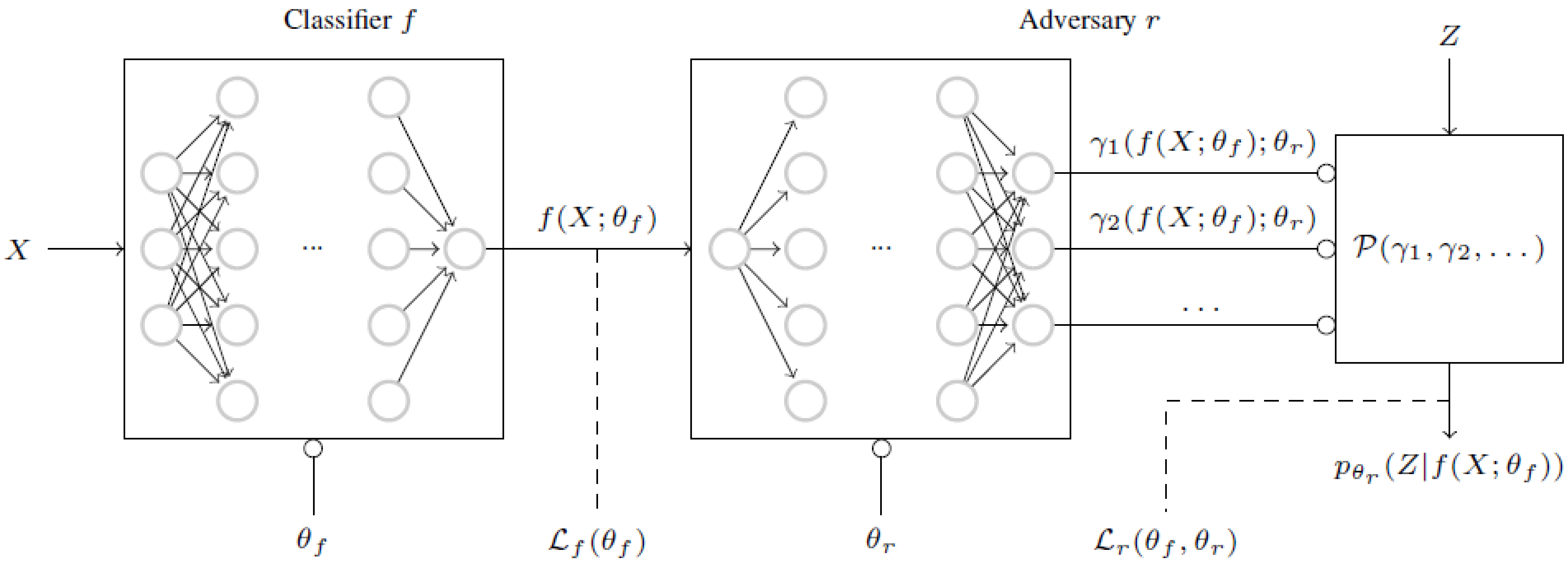
The minimax solution of this problem is reached for

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

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

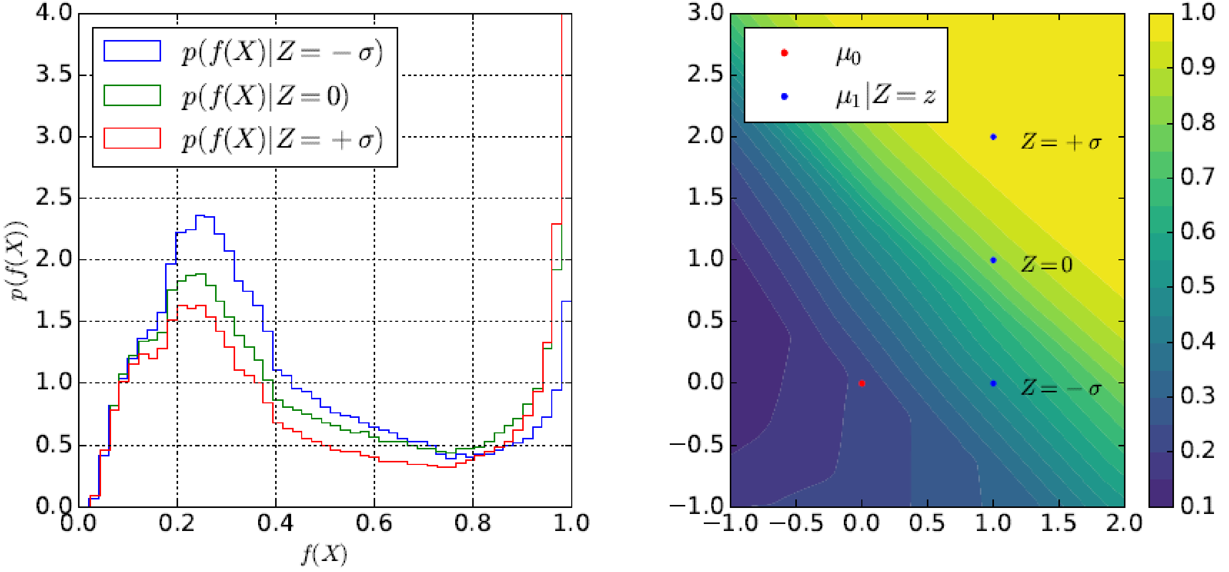
The architecture is a series of two discriminative classifiers: the adversary tries to model p(z|f(x)), and the global loss forces this toward the unconditional prior p(z). When this happens, f is independent on z.

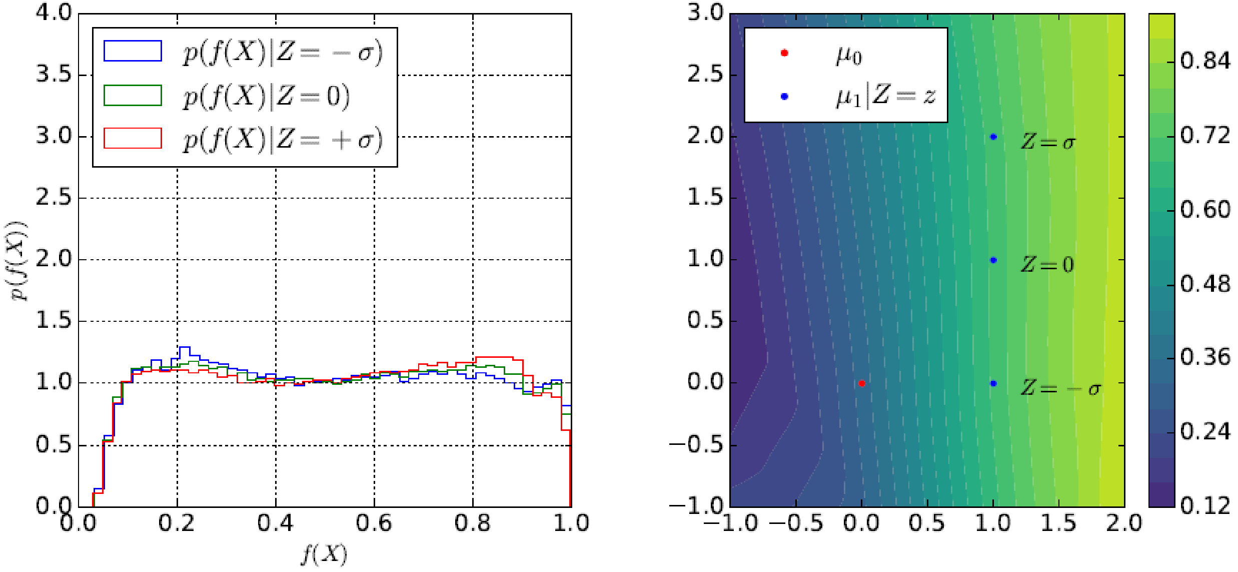

Classifier f
Adversary r
Z
X
$f(X;\theta_f)$
$\gamma_1(f(X;\theta_f);\theta_r)$
$\gamma_2(f(X;\theta_f);\theta_r)$
$\dots$
$\mathcal{P}(\gamma_1,\gamma_2,\dots)$
$p_{\theta_r}(Z|f(X;\theta_f))$
θ_f
$\mathcal{L}_f(\theta_f)$
θ_r
$\mathcal{L}_r(\theta_f,\theta_r)$

Experiments with the pivot

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.



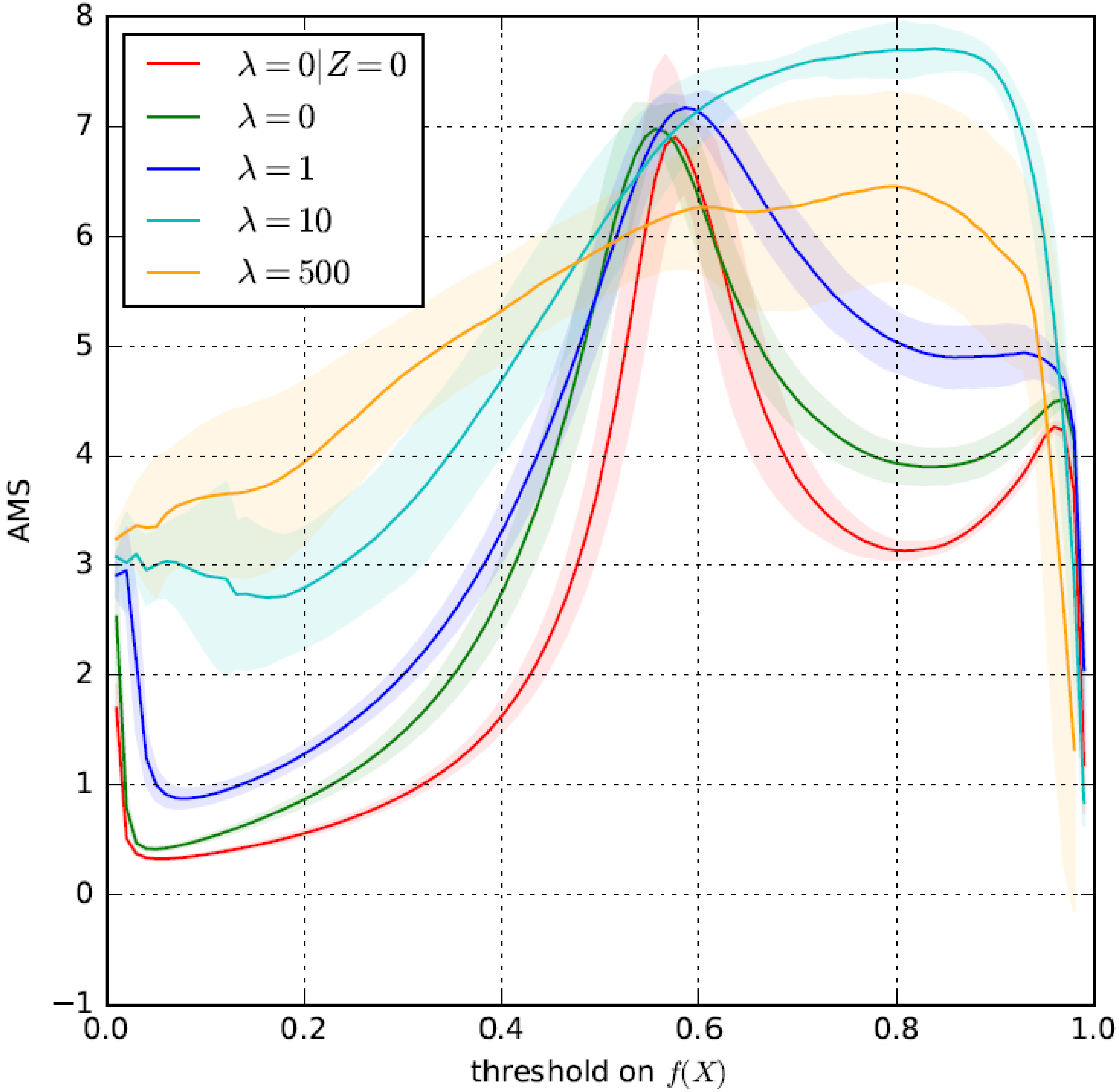


HEP example

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 Lf 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



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 (e.g. control samples, different admixtures)

These 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 or BCE, and t is the target label), in LLP one only exploits knowledge of the fraction of each label in training data:

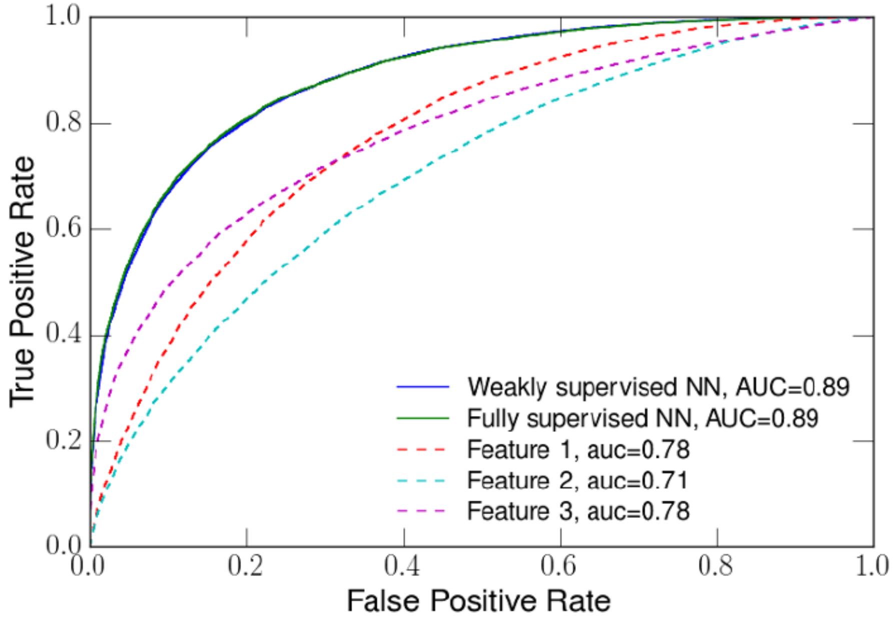
$$f_{\text{weak}} = \text{argmin}_{f':\mathbb{R}^n \to [0,1]} \ell\left(\sum_{i=1}^{N} \frac{f'(x_i)}{N} - y\right)$$

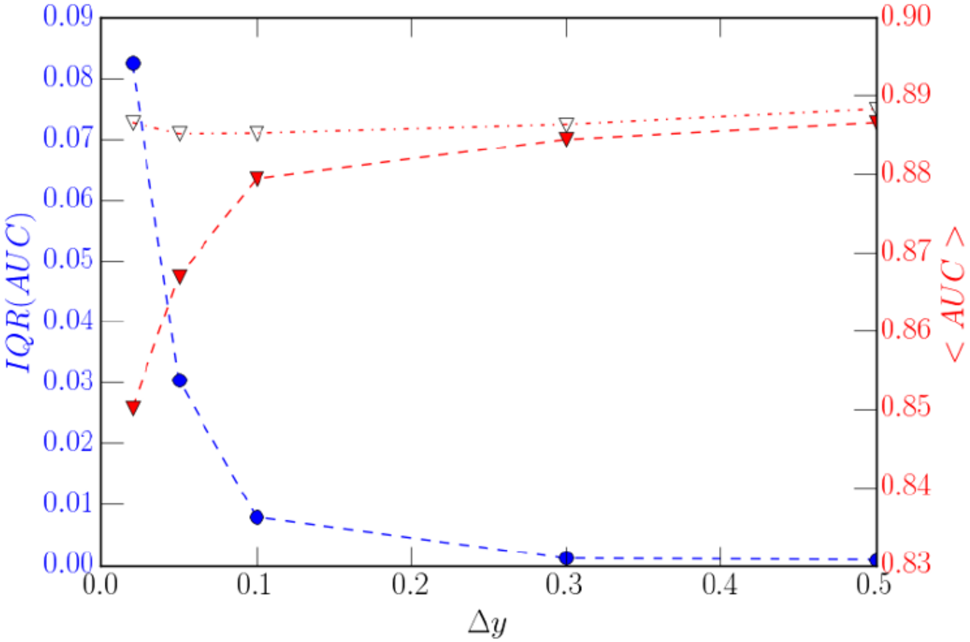
The problem is thus very ill-constrained, but a minimization of the loss can still be performed with batches of data of different class proportions, *as long as the PDFs of the features do not change in the batches*.

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)



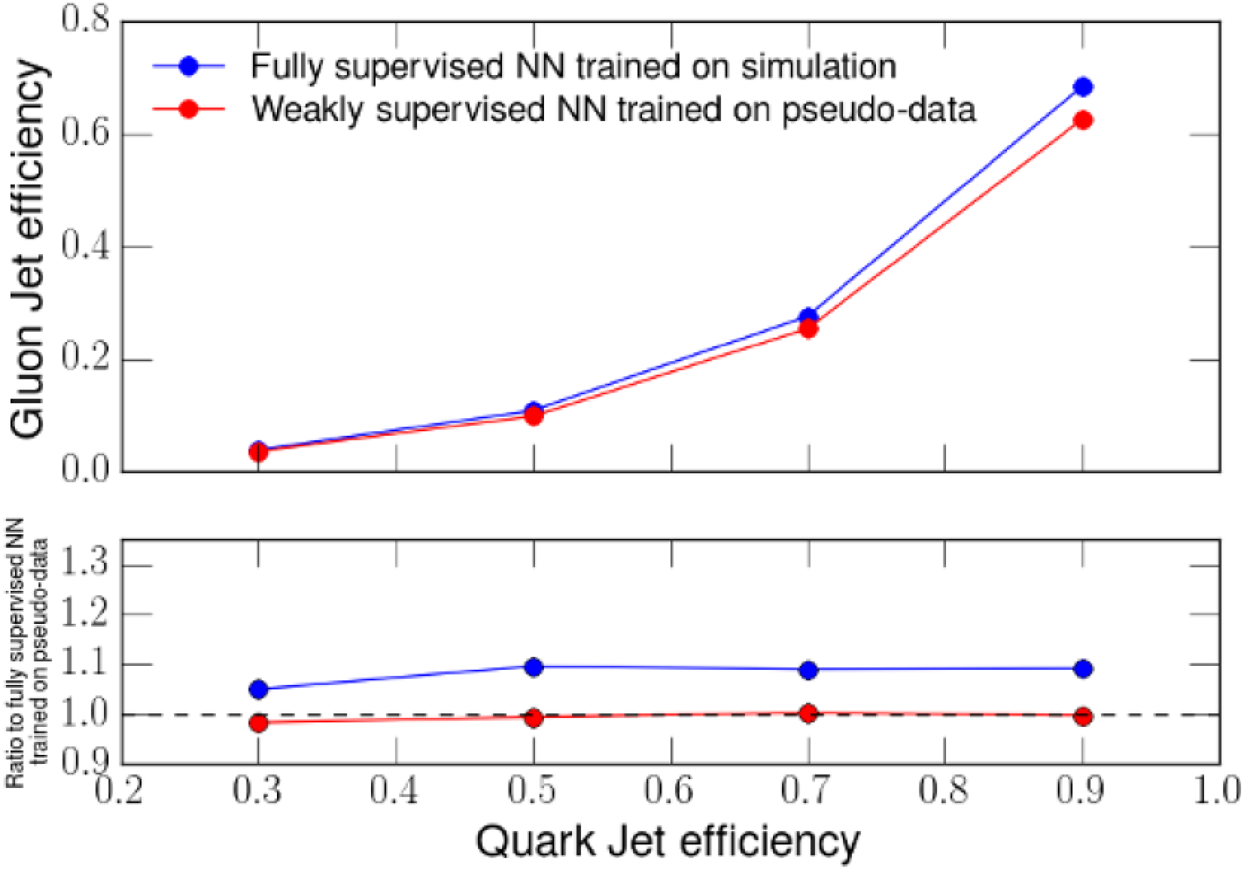


Test of LLP on Q/G discrimination

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.

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

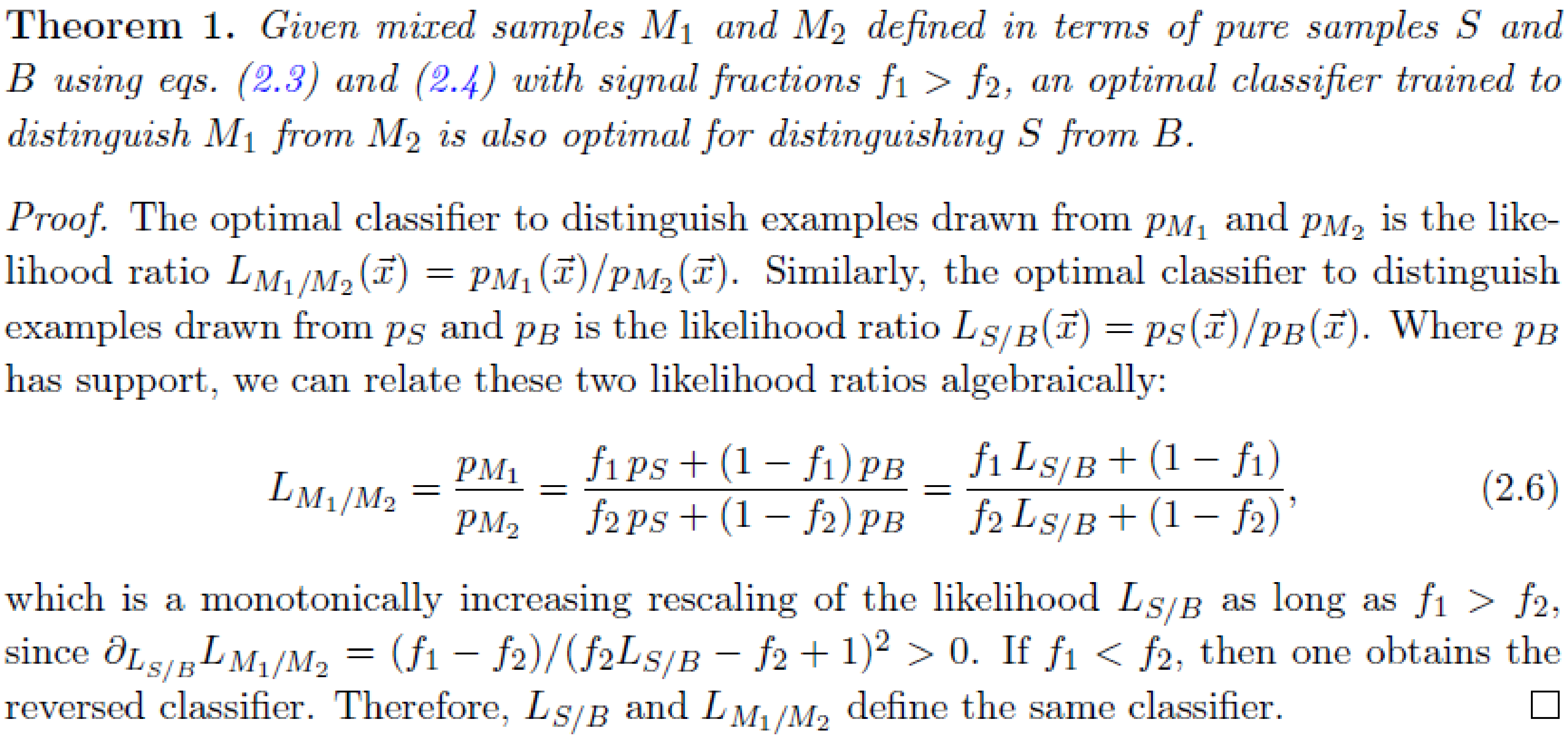
In this work, a distortion of real data is mimicked by modeling previous studies; then a comparison with a full supervised classifier shows 10% advantages (lower G efficiency for given Q efficiency)



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.*

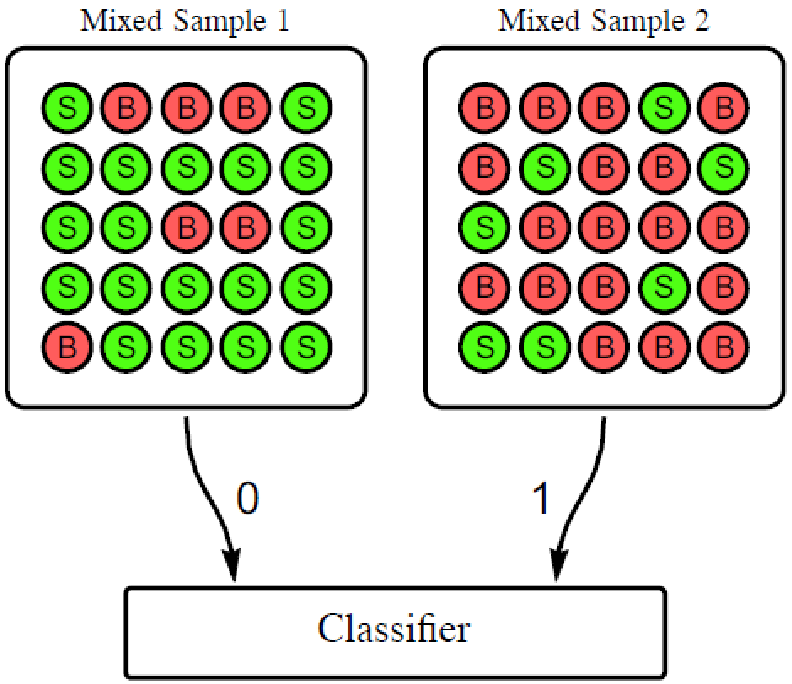
Proof. The optimal classifier to distinguish examples drawn from p_{M_1} and p_{M_2} is the likelihood ratio $L_{M_1/M_2}(\vec{x}) = p_{M_1}(\vec{x})/p_{M_2}(\vec{x})$. Similarly, the optimal classifier to distinguish examples drawn from p_S and p_B is the likelihood ratio $L_{S/B}(\vec{x}) = p_S(\vec{x})/p_B(\vec{x})$.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

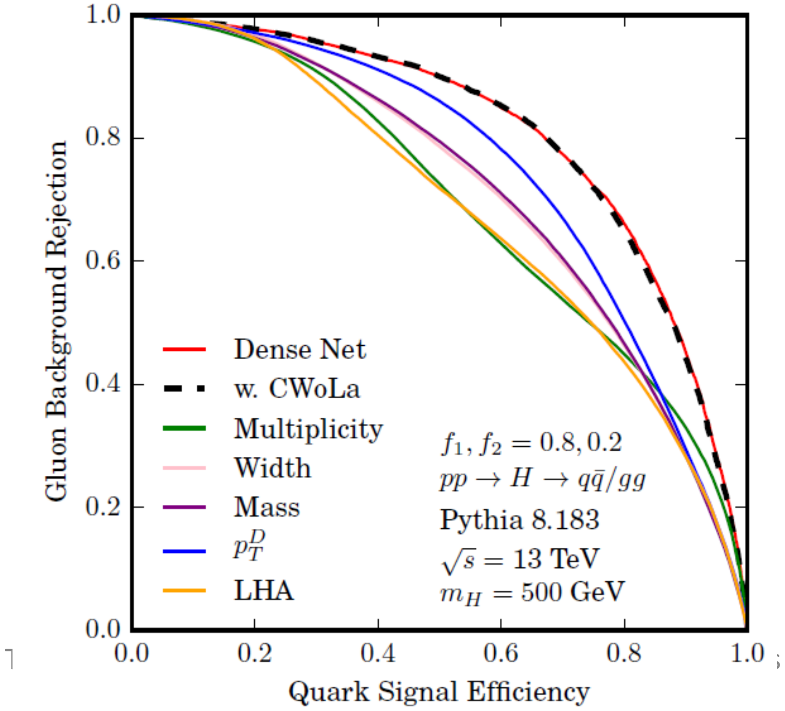
$$L_{M_1/M_2} = \frac{p_{M_1}}{p_{M_2}} = \frac{f_1\, p_S + (1 - f_1)\, p_B}{f_2\, p_S + (1 - f_2)\, p_B} = \frac{f_1\, L_{S/B} + (1 - f_1)}{f_2\, L_{S/B} + (1 - f_2)}, \tag{2.6}$$

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 performs as well as a NN working on pure classes if trained on classes with 80%-20% class proportion split.



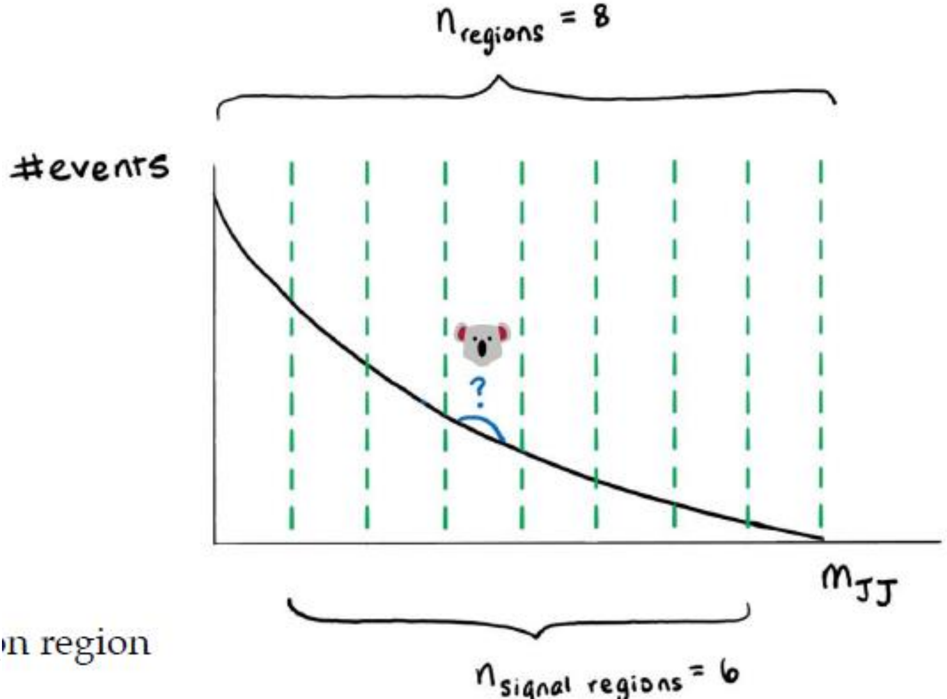


Of course the algorithm still requires labelled data for tests of performance and choice of operating point, but the proof of principle is encouraging.

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])



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 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-dim 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)

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

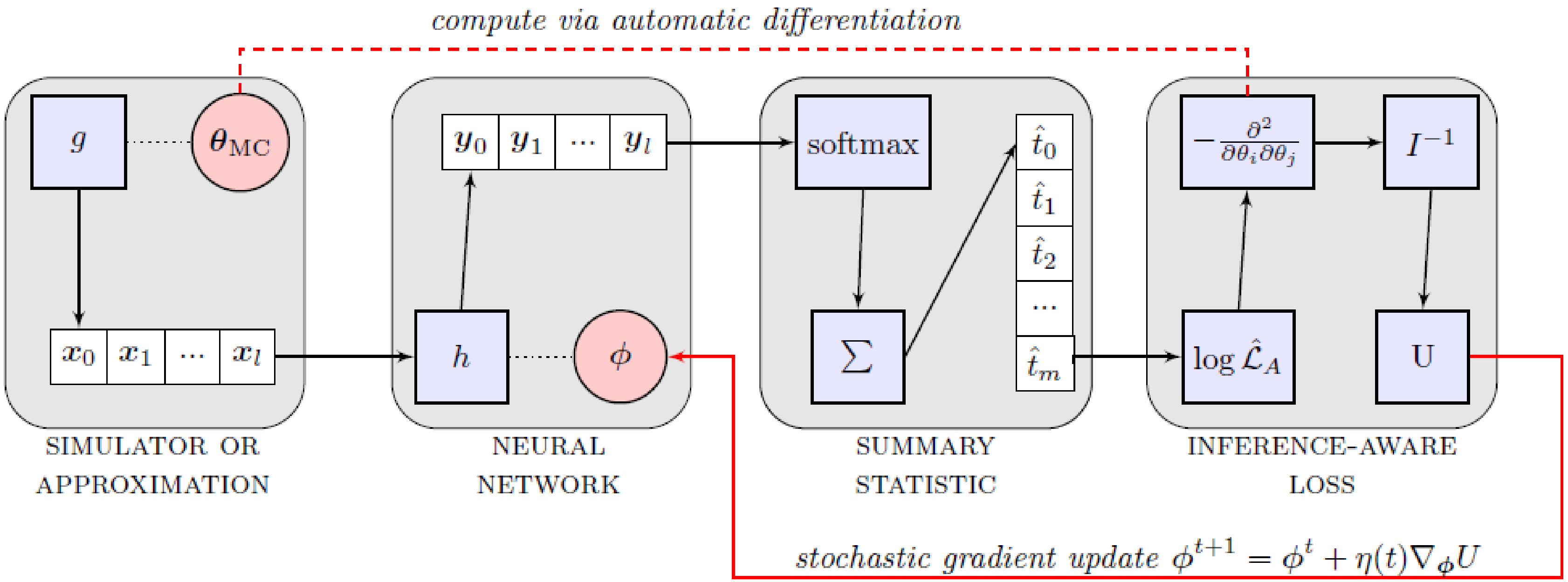
Block 1: A simulator or an approximation of it is used to sample observations given parameters θ

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


compute via automatic differentiation
g
θ_{MC}
x_0 x_1 $\cdots$ x_l
y_0 y_1 $\cdots$ y_l
h
ϕ
softmax
Σ
$\hat{t}_0$ $\hat{t}_1$ $\hat{t}_2$ $\cdots$ $\hat{t}_m$
$-\frac{\partial^2}{\partial\theta_i\partial\theta_j}$
I^{-1}
$\log \hat{\mathcal{L}}_A$
U
SIMULATOR OR APPROXIMATION
NEURAL NETWORK
SUMMARY STATISTIC
INFERENCE-AWARE LOSS
stochastic gradient update $\phi^{t+1} = \phi^t + \eta(t)\nabla_\phi U$

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:

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

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.

Instead of real data, if we use simulated data we may 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,

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

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, e.g.

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

INFERNO synthetic example

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

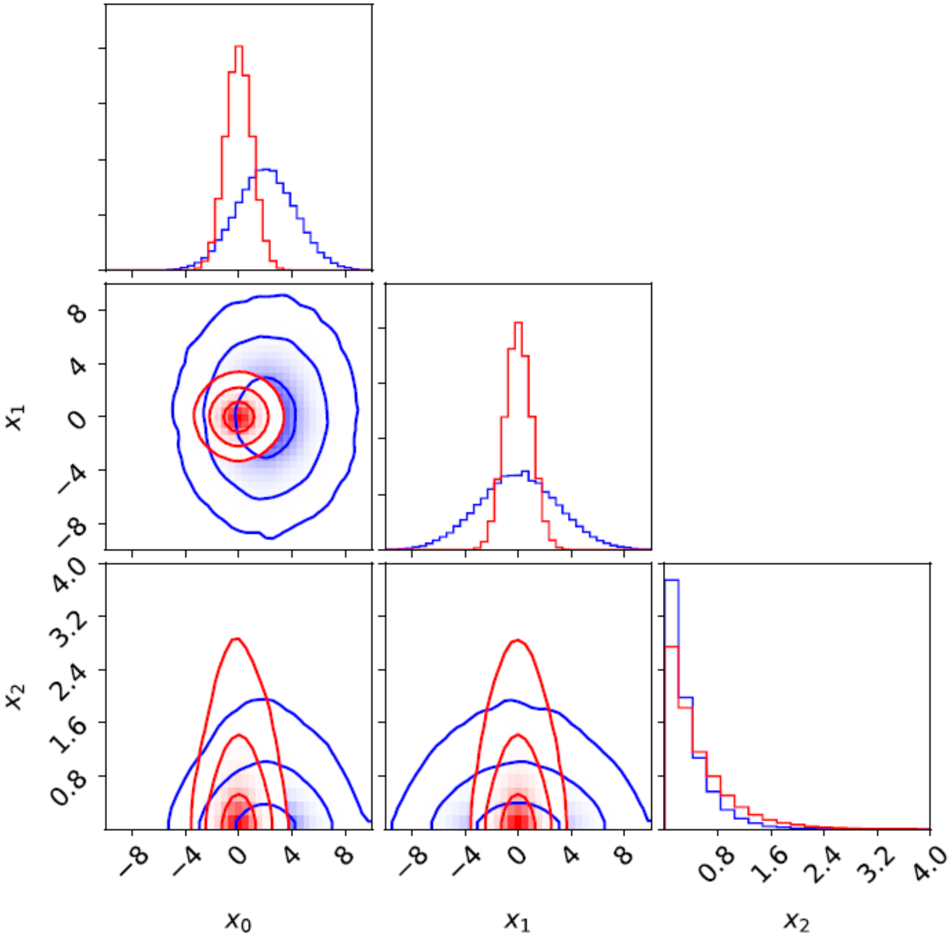
$$f_b(\boldsymbol{x}|r,\lambda) = \mathcal{N}\left((x_0,x_1)\,\middle|\,(2+r,0),\begin{bmatrix}5 & 0\\ 0 & 9\end{bmatrix}\right) Exp(x_2|\lambda)$$

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

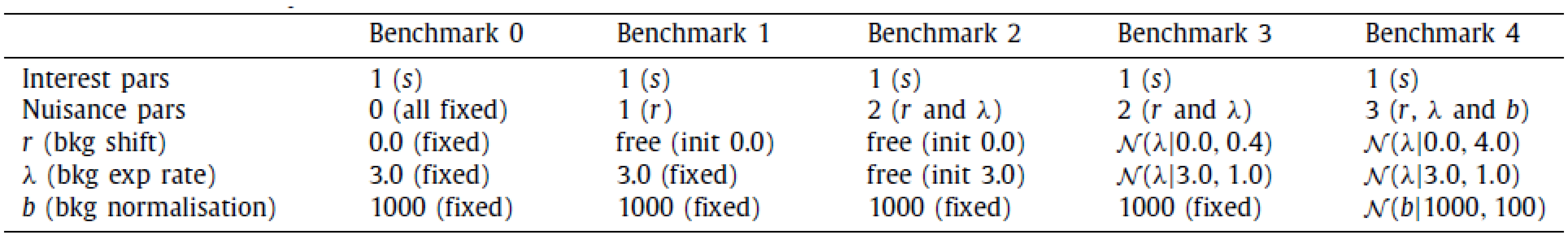
r shifts the background mean, λ changes the slope, and b is background normalization. The model is then

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

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

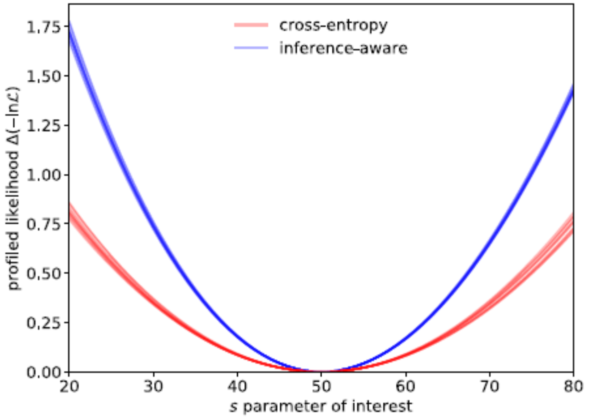


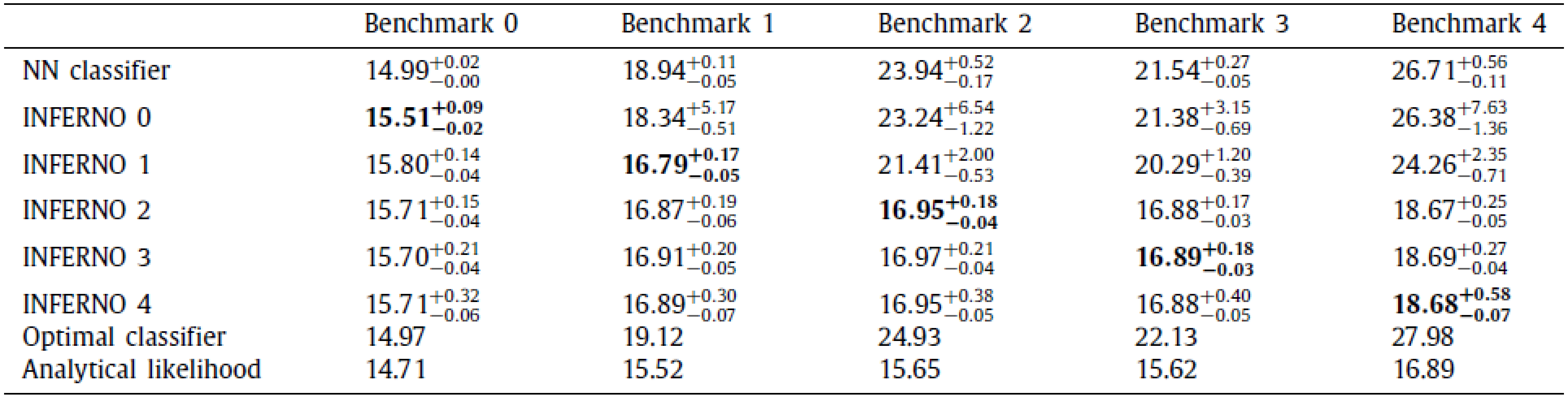
INFERNO results



	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

←Sqrt(variance) values

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 has been developed for HEP problems in recent years, to try and remove the impact of systematic uncertainties in supervised classification.
- The focus in many cases is achieving a decorrelation of salient features (jet mass), to maximize discovery significance
  - 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 a cutting-edge area of research in ML

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, ttbb cross section measurement, 2019.

[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.