Optimized Methods to Apply Neural Networks in HEP

- Main steps of general analysis approaches
- Optimization methods at different steps
- Possible new approaches for the optimizations

Lev Dudko

Lomonosov Moscow State University

General tasks of LHC analysis

- **Measure total cross section or some model parameter like coupling**
  - ➔ Maximize S/B ratio and perform statistical analysis with number of events or/and shape of some distributions
- **Measure differential cross section**
  - ➔ Maximize S/B ratio, estimate background contribution and distinguish signal contribution to the shape of a distribution
- **Search for a resonance particle or a special shape of one distribution**
  - ➔ Maximize S/B ratio and perform template fit analysis of a distribution

Every analysis needs to maximize S/B ratio or distinguish shapes of signal and background distributions. The optimal way is to prepare a classifier to separate signal from background.

Common analysis techniques

- **Weak points of the methods**
  - Cut-based and Decision Trees methods use triangle cuts in multi-dimension space therefore it is not very efficient. Boosting algorithm helps to improve the efficiency of DT, but also can be applied with NN and other classifiers.
  - Likelihood function is usually far from some optimal function to classify the events and requires special study in each case.
  - Matrix element approach tries to use analytic form of Matrix element of signal process for the probability function. The main problem – it is mostly impossible to get analytic form for the processes of interest and backgrounds. Therefore, use events simulated by MC and other classification methods is usually more optimal.
  - NN requires different steps of optimizations and tuning to avoid known problems and prepare efficient classifier
- **Mostly all of the methods require some set of observables to analyze**

Method of “optimal observables”

- **Provides general recipe how to choose most sensitive variables to separate signal and background**
  - It is based on the analysis of Feynman diagrams contributing to signal and background processes
  - Distinguish **three classes** of sensitive variables for the signal and each of kinematically different backgrounds
  - Set of variables can be extended with other type of information, like detector relative variables (jet width, b-tagging discriminant)

- **Described in different examples for the top and Higgs searches**
  - Eur.Phys.J. C11 (1999) 473-484
  - Nucl.Instrum.Meth. A502 (2003) 486-488
  - Phys.Atom.Nucl. 71 (2008) 388-393

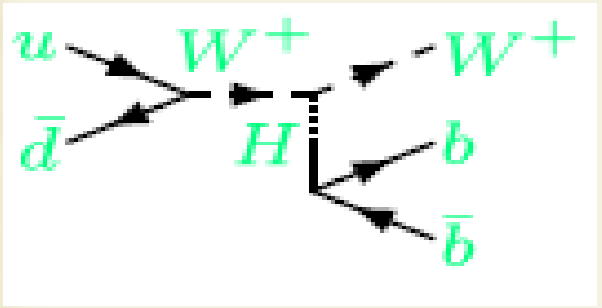
- **Applied in different experimental analysis in D0 and CMS**
  - Phys.Lett. B517 (2001) 282-294 and other D0 publications
  - CMS-PAS-TOP-14-007, new paper is coming soon

Three classes: Singular variables

- **Most of the rates of signal and background processes come from the integration over the phase space region close to the singularities. If some of the singular variables are different or the positions of the singularities are different the corresponding distributions will differ most strongly**
  - ➔ Corresponds to the differences in denominators of Feynman diagrams
  - ➔ There are only two types of singularities t- and s-channel singularities

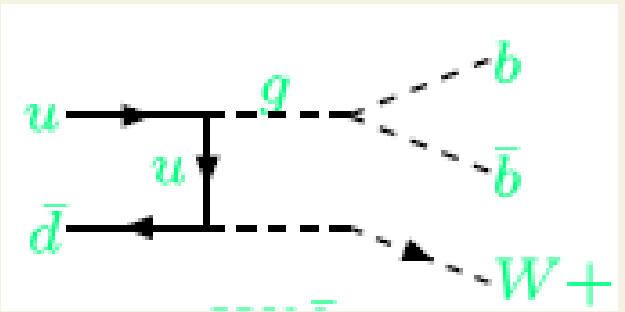
s-channel singularities

$$M^2_{f1,f2} = (p_{f1} + p_{f2})^2$$



t-channel singularities

$$\hat{t}_{i,f} = (p_f - p_i)^2 = -\sqrt{\hat{s}}e^{Y}\, p_T^f e^{-|y_f|}$$



Three classes: Angular and Threshold variables

- **Angular variables, reflect spin correlations**
  - Correspond to the differences in numerators of Feynman diagrams
  - Need special study for each particular process, most interesting processes are already considered in phenomenology papers, e.g. for single top:

$$\frac{1}{\Gamma_T}\frac{d\Gamma}{d(\cos\chi_\ell^W)} = \frac{3}{4}\frac{m_t^2\sin^2\chi_\ell^W + 2m_W^2\frac{1}{2}(1-\cos\chi_\ell^W)^2}{m_t^2+2m_W^2}$$

G. Mahlon, S. Parke Phys.Rev. D55 (1997) 7249-7254

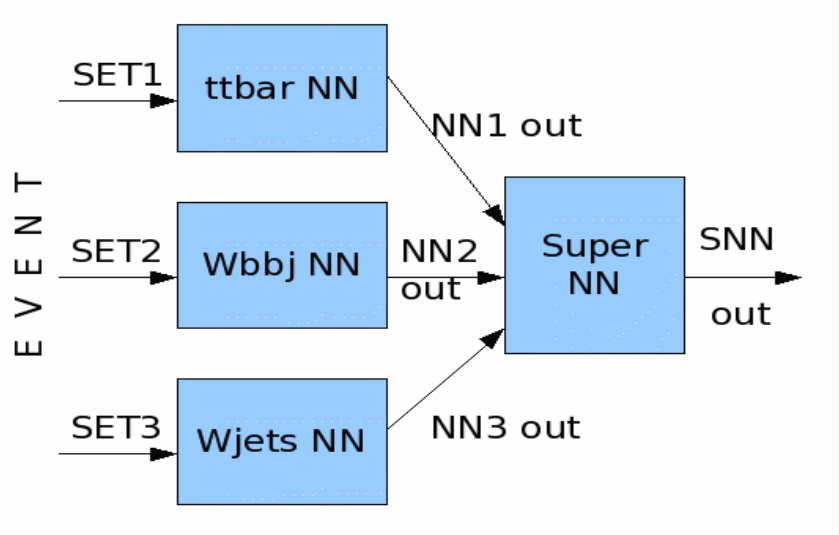
- **Threshold variables**
  - Various signal and background processes may have very different energy thresholds
  - Variables like S_hat, Ht are sensitive to the different energy thresholds

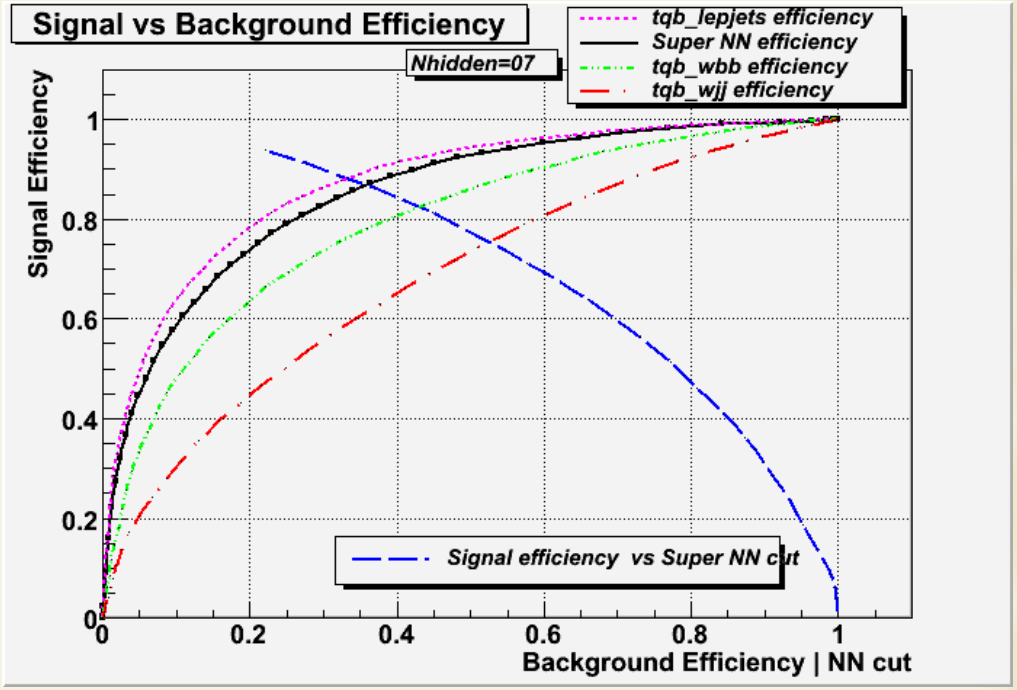
Application of feed-forward NNs

Phys.Atom.Nucl. 73 (2010) 971-984

- **Set of input variables requires a preprocessing for the most efficient separation of the classes by NN**
  - Main idea of preprocessing is to simplify the task of classification for the NN
  - At least need to apply normalisation of variables to the same scale and apply logarithmic transformation for the variables with long tail. Some times it is useful to apply more sophisticated methods like PCA, ICA, etc.
- **In case the backgrounds are significantly different in kinematics it is more efficient to prepare separate/parallel NNs to distinguish signal and one kinematical class of the background**
  - Provides better separation with particular backgrounds, but increases dimensionality of the output discriminants
  - Super NN can be applied to combine outputs of parallel NNs and provide one dimensional discriminant
- **Use NN output**
  - Apply a cut on NN output based on some criteria of optimization
  - Use shape of NN discriminant for the statistical analysis

Application of feed-forward NNs



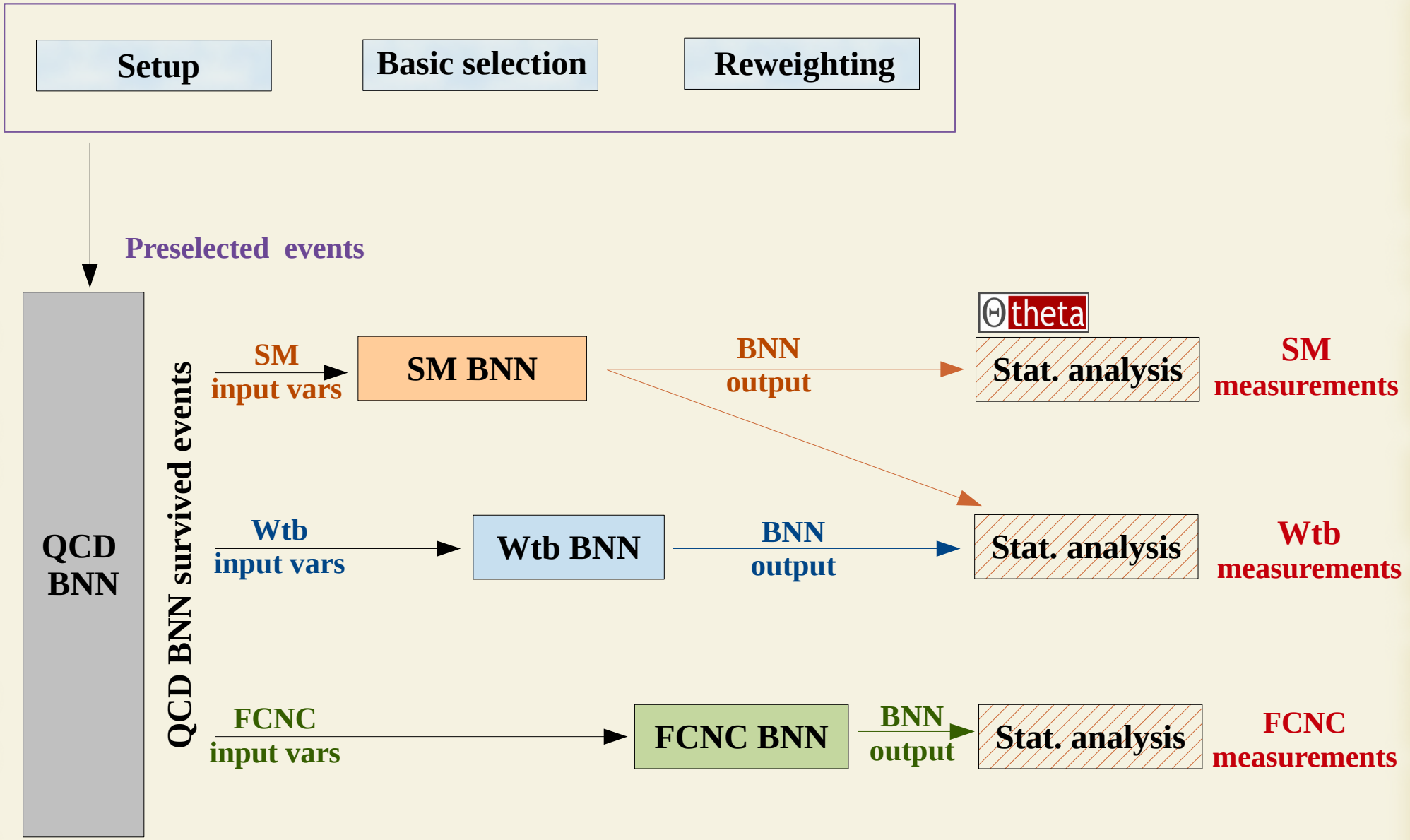


- **Real example with one signal and three backgrounds (MLPfit sftw.)**
  - Super NN is one dimensional and most efficient discriminant for the weighted sum of the backgrounds
  - Plot with Eff. demonstrates Eff. of sub-networks for particular S/B and weighted sum of the backgrounds
- **SuperNN usually increases shape systematic uncertainty. Can be applied if the analysis is limiting in statistical uncertainty, not systematic one**

Feed-forward NNs and Bayesian NNs

- **Optimal FF NN requires manual tuning**
  - Need to avoid over-fitting problem
  - Tune architecture of NN, training parameters, etc.
  - It is not very stable relative to the tuning
- **R. M. Neal proposed Bayesian NN approach where each NN internal weight is not a number but a distribution**
  - "Bayesian Learning for Neural Networks" http://www.cs.utoronto.ca/~radford/bnn.book.html
  - FBM package is the software realization for this idea http://www.cs.toronto.edu/~radford/fbm.software.html
  - First use in HEP: P. Bhat and H. Prosper, “Bayesian neural networks”, Conf. Proc. C050912 (2005) 151.
- **BNN provides the same level of Eff. as FF NN but it is very stable to the modifications of tuning parameters, architecture and practically does not affected by over-fitting problem**
  - Does not require manual tuning
  - Require significantly more CPU resources

Scheme of real application of BNNs in CMS

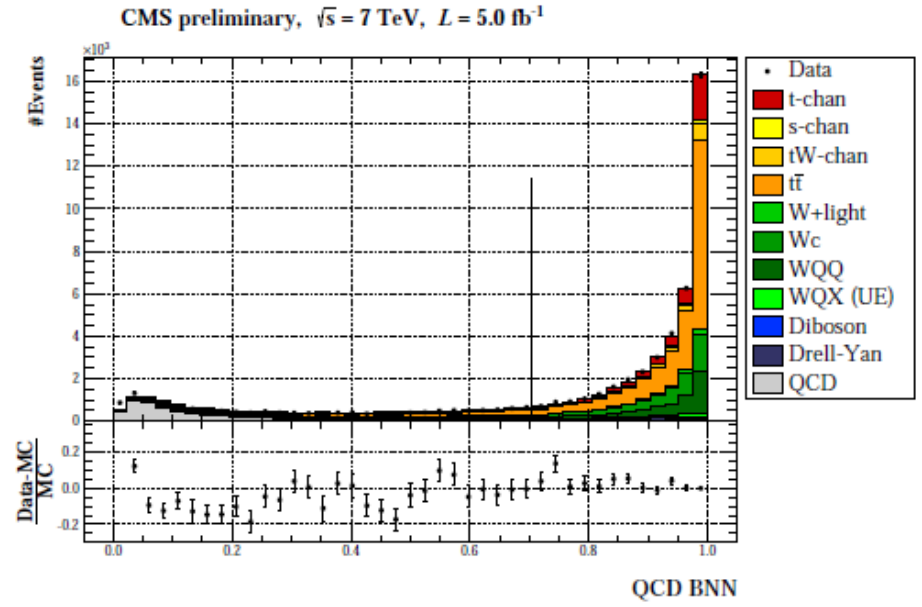


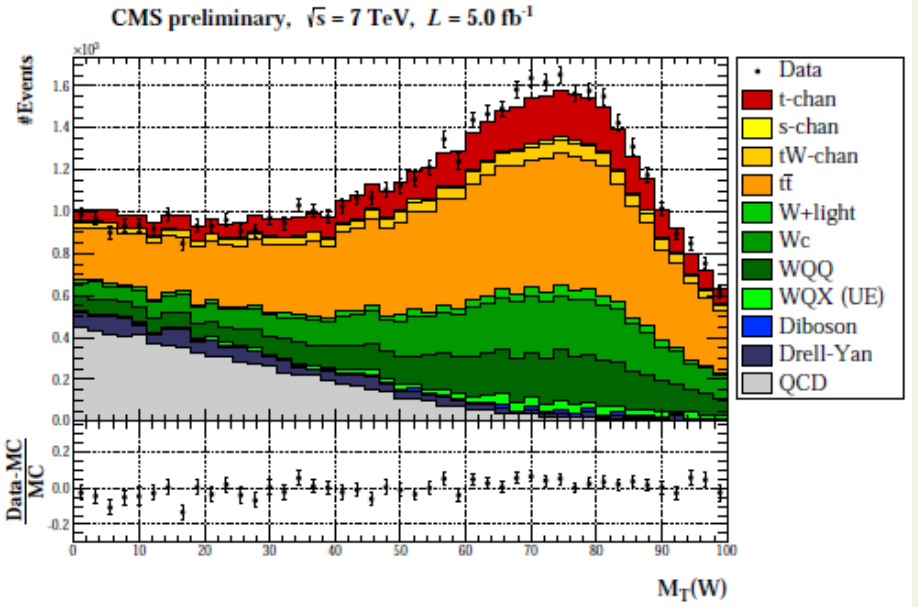
CMS-PAS-TOP-14-007

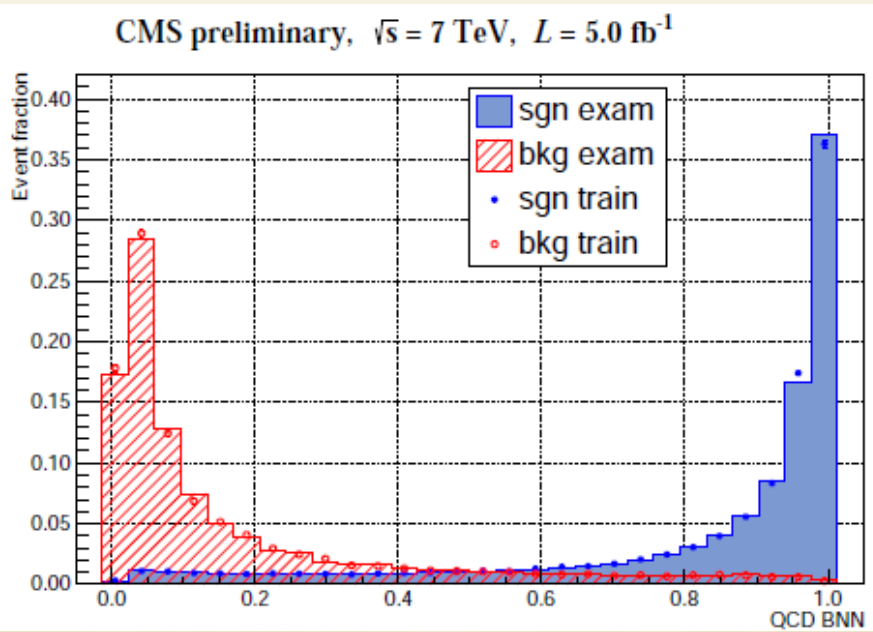
CMS AN-14-013, Paper TOP-14-007

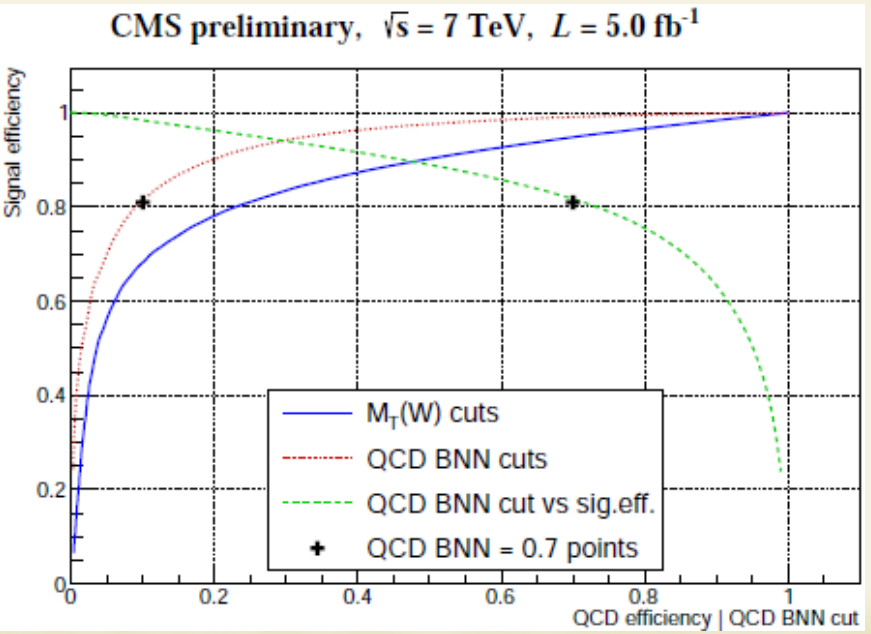
QCD BNN

Use 4 simple variables to minimize sys. uncert. Separation Eff. is significantly better than with one most efficient variable. Cut QCD BNN at 0.7 and use it as a filter. Survived events are passed to next analysis steps.



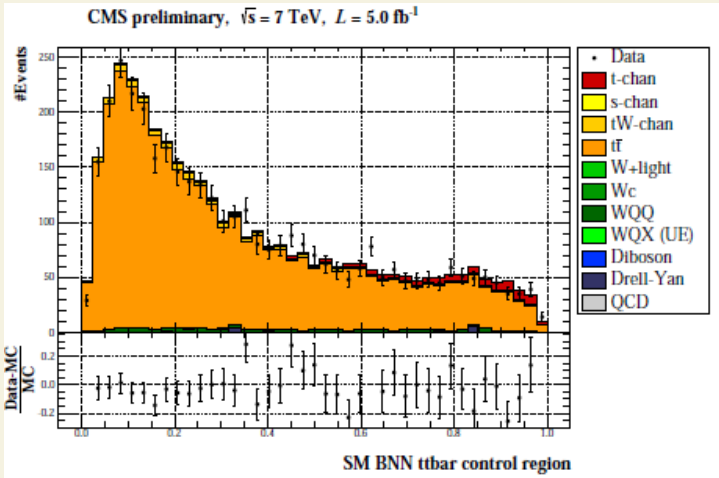


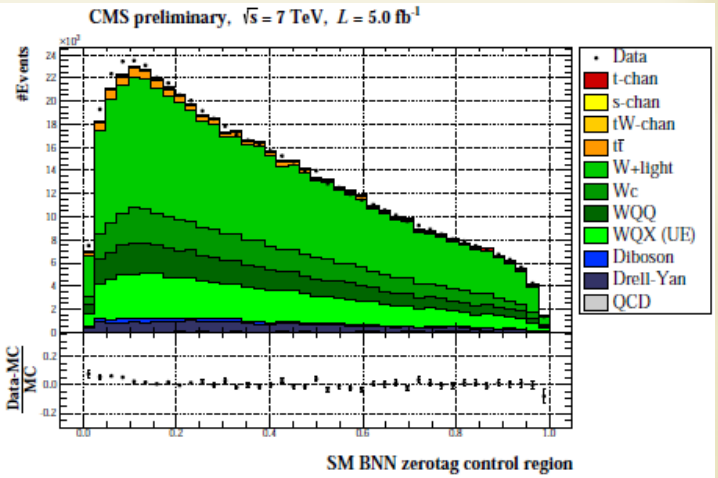


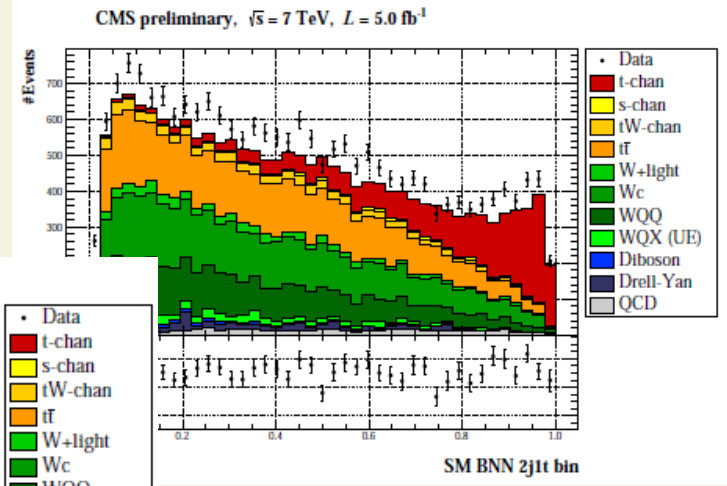


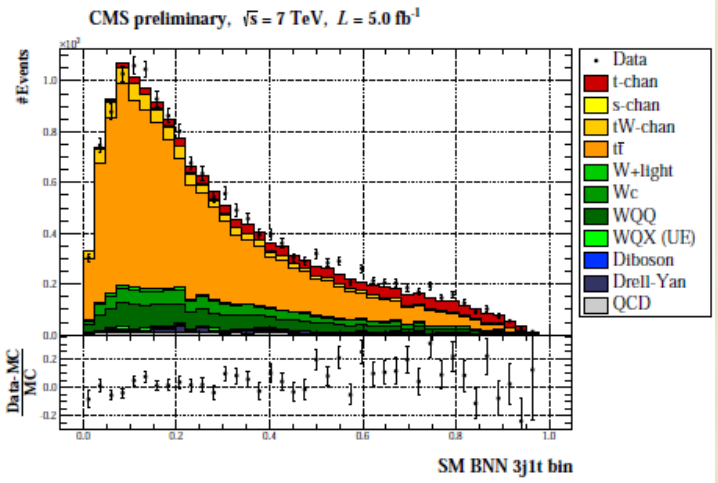
Cross checks of BNN

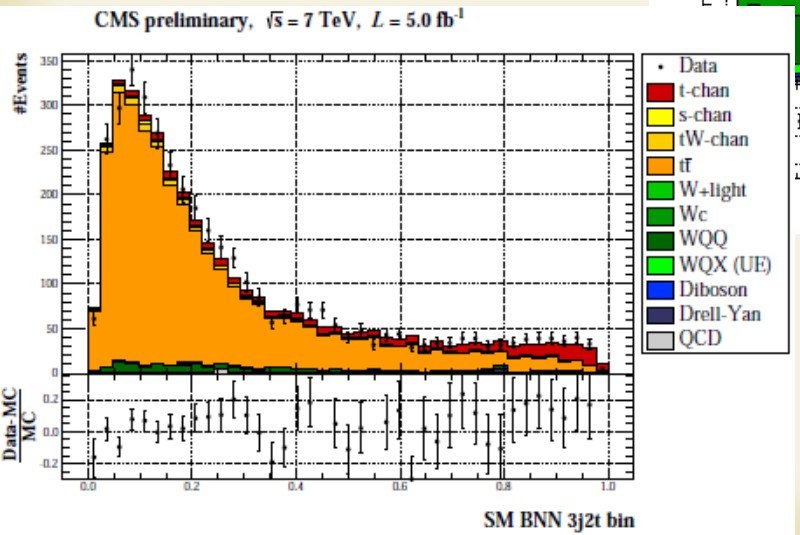
One of the main step of the analysis is to check the BNN with orthogonal event samples, in different regions of phase space and compare model response with real data if it is available





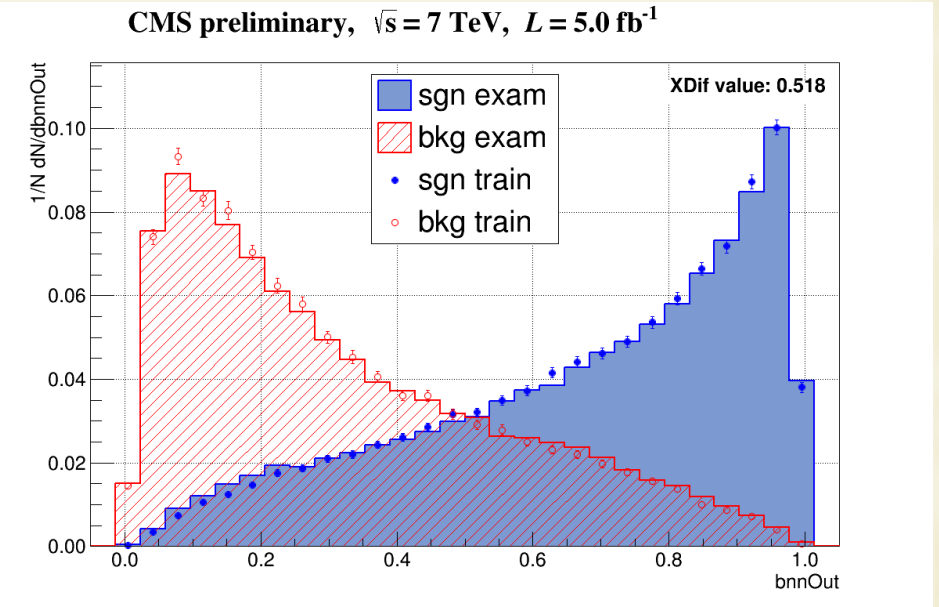




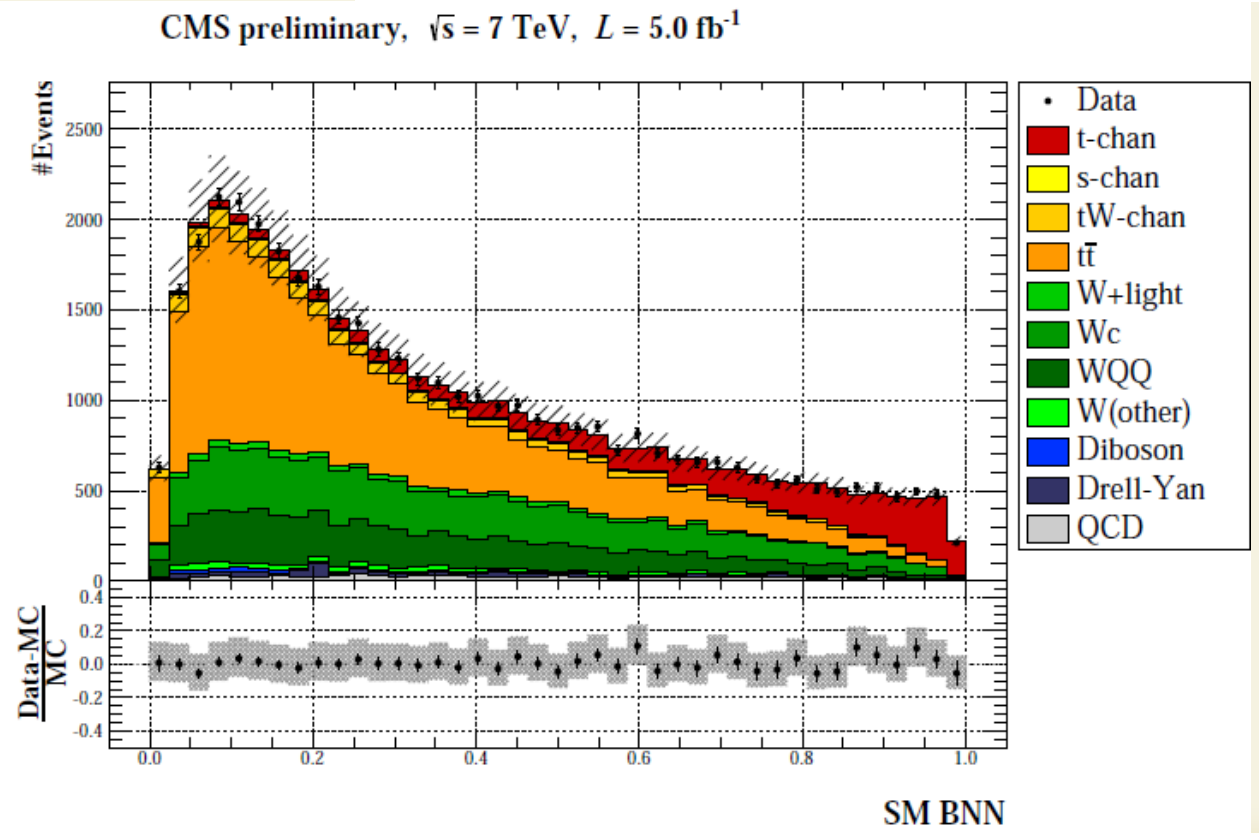


Final BNN classifier

After the set of cross checks and evaluation of the uncertainties the final classifier is ready for the statistical analysis to measure necessary physics parameter based on the shape of the BNN output.



This real example includes optimization of the S/B separtion Eff. and minimization of all types of uncertainties (syst-normalisation, syst-shape, syst-shape-umarginalized, stat).



What else we can optimize?

- **The Eff. of our classifiers are now close to optimal since we have to optimize not only the separation power but also the uncertainty**

- **Why classification? Clusterization task could be also efficient.**
  - Usually the clusterization with unsupervised training is less efficient from the separation point of view, but it could be efficient if we consider separation power and uncertainties simultaneously.
  - The possible methods are Self Organizing Maps and Radial Basis Functions

- **For what tasks the clusterization could be efficient?**
  - Blind search for something. Use it with known simulation to decrease systematic uncertainties.
  - New approach to reconstruction programs. Distinguish objects in subdetectors. Jet separation. Overall corrections.

Conclusion

- **Now, there are mostly standard methods to get optimized results with multivariate analysis for many LHC tasks**
  - This talk presents our group experience of the optimization approaches. We have found BNN as most stable and efficient classifier
  - Optimization for each of the analysis steps should take into account systematic and statistic uncertainties of the result, but not only separation power of a classifier
  - There is significant progress in ML software during the last years, but not in the classification methods used in HEP. Probably, we need fundamental changes in multivariate methods in HEP for farther improvements
- **Promising directions of farther improvements are clustering algorithms (SOM, RBF) and Deep learning algorithms**