Physics Analysis with Advanced Data Mining Techniques

Hai-Jun Yang

University of Michigan, Ann Arbor

CCAST Workshop

Beijing, November 6-10, 2006

Outline

- Why Advanced Techniques ?
- Artificial Neural Networks (ANN)
- Boosted Decision Trees (BDT)
- Application of ANN/BDT for MiniBooNE neutrino oscillation analysis at Fermilab
- Application of ANN/BDT for ATLAS Di-Boson Analysis
- Conclusions and Outlook

Why Advanced Techniques?

- Limited signal statistics, low Signal/Background ratio
  - To suppress more background & keep high Signal Efficiency

➔Traditional Simple-Cut technique
  - Straightforward, easy to explain
  - Usually poor performance

➔Artificial Neural Networks (ANN)
  - Non-linear combination of input variables
  - Good performance for input vars ~20 variables
  - Widely used in HEP data analysis

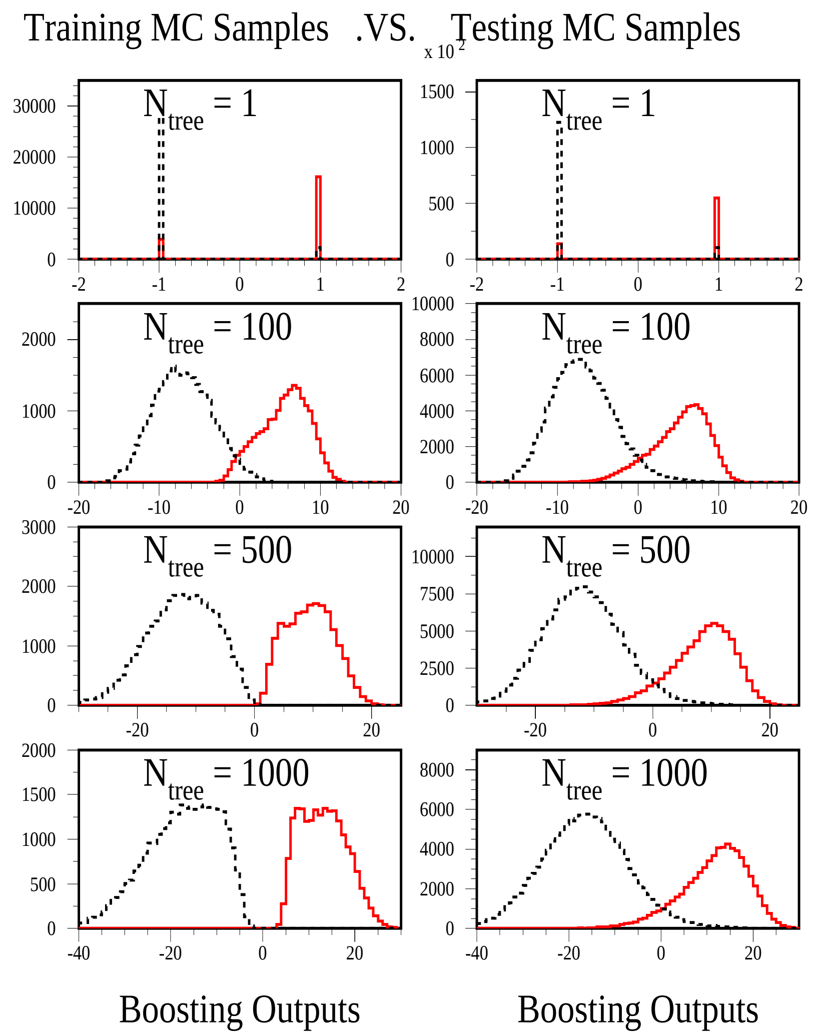
➔Boosted Decision Trees (BDT)
  - Non-linear combination of input variables
  - Great performance for large number of input variables (up to several hundred variables)
  - Powerful and stable by combining many decision trees to make a “majority vote”

Training and Testing Events

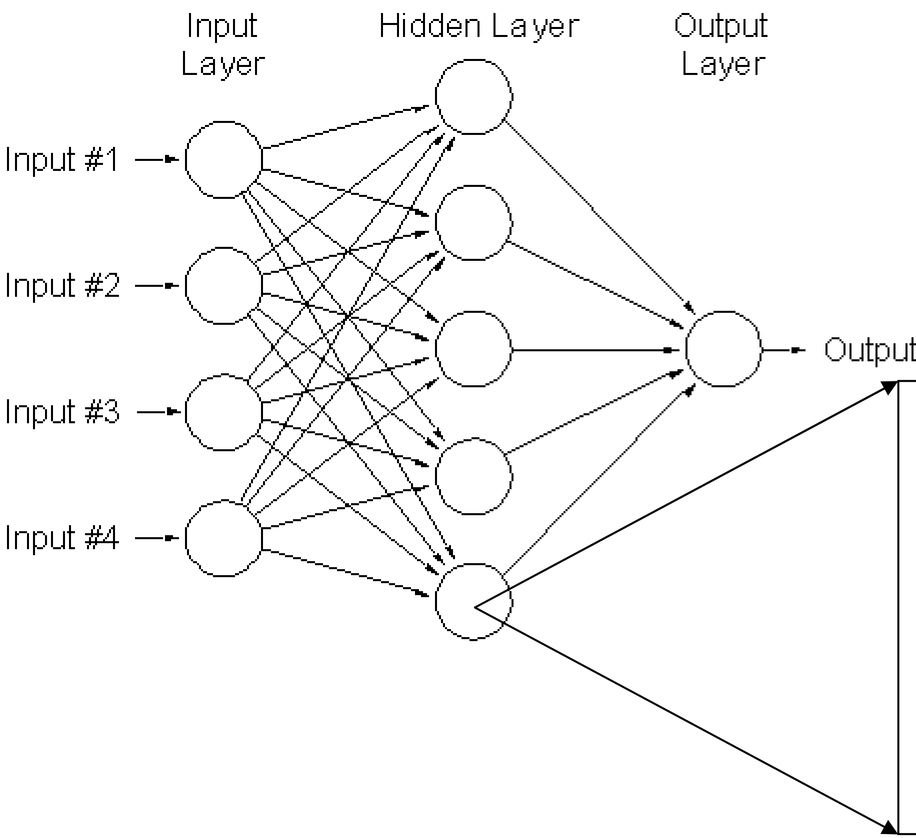
- Both ANN and BDT use a set of known MC events to train the algorithm.
- A new sample, an independent testing set of events, is used to test the algorithm.
- It would be biased to use the same event sample to estimate the accuracy of the selection performance because the algorithm has been trained for this specific sample.
- All results quoted in this talk are from the testing sample.

Results of Training/Testing Samples

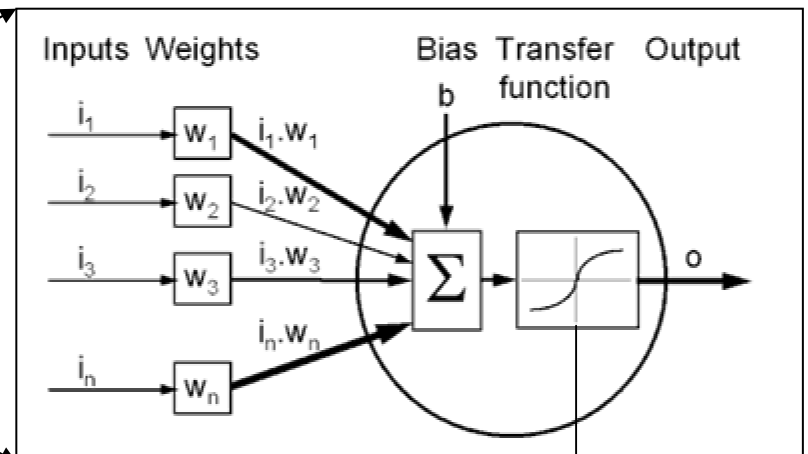
- ➔The AdaBoost outputs for MiniBooNE training/testing MC samples with number of tree iterations of 1, 100, 500 and 1000, respectively.
- ➔The signal and background (S/B) events are completely distinguished after about 500 tree iterations for the training MC samples. However, the S/B separation for testing samples are quite stable after a few hundred tree iterations.
- ➔The performance of BDT using training MC sample is overestimated.



Artificial Neural Networks (ANN)



➔Use a training sample to find an optimal set of weights/thresholds between all connected nodes to distinguish signal and background.



$$g(x) = \frac{1}{2}[1 + \tanh(x)] = (1 + e^{-2x})^{-1}$$

Artificial Neural Networks

- Suppose signal events have output 1 and background events have output 0.
- Mean square error E for given N_p training events with desired output o = 0 (for background) or 1 (for signal) and ANN output result t .

$$E = \frac{1}{2N_p} \sum_{p=1}^{N_p} \sum_i (o_i^{(p)} - t_i^{(p)})^2$$

Artificial Neural Networks

- Back Propagation Error to Optimize Weights

$$w_{t+1} = w_t + \Delta w_t,$$

$$where$$

$$\Delta w_t = -\eta \frac{\partial E}{\partial w}$$

$$+\alpha \Delta w_{t-1}, "momentum\_term\_to\_stabalize"$$

$$+\sigma, "noise\_term\_to\_avoid\_local\_\min ima"$$

ANN Parameters

$\eta = 0.05$
$\alpha = 0.07$
$T = 0.50$

- Three layers for the application
  - # input nodes(= # input variables) – input layer
  - # hidden nodes(= 1~2 X # input variables) – hidden layer
  - 1 output node – output layer

Boosted Decision Trees

- What is a decision tree?
- How to boost decision trees?
- Two commonly used boosting algorithms.

Decision Trees & Boosting Algorithms

- ➔ Decision Trees have been available about two decades, they are known to be powerful but unstable, i.e., a small change in the training sample can give a large change in the tree and the results.

  Ref: L. Breiman, J.H. Friedman, R.A. Olshen, C.J.Stone, "Classification and Regression Trees", Wadsworth, 1983.

- ➔ The boosting algorithm (AdaBoost) is a procedure that combines many "weak" classifiers to achieve a final powerful classifier.

  Ref: Y. Freund, R.E. Schapire, "Experiments with a new boosting algorithm", Proceedings of COLT, ACM Press, New York, 1996, pp. 209-217.

- ➔ Boosting algorithms can be applied to any classification method. Here, it is applied to decision trees, so called "Boosted Decision Trees". The boosted decision trees has been successfully applied for MiniBooNE PID, it is 20%-80% better than that with ANN PID technique.

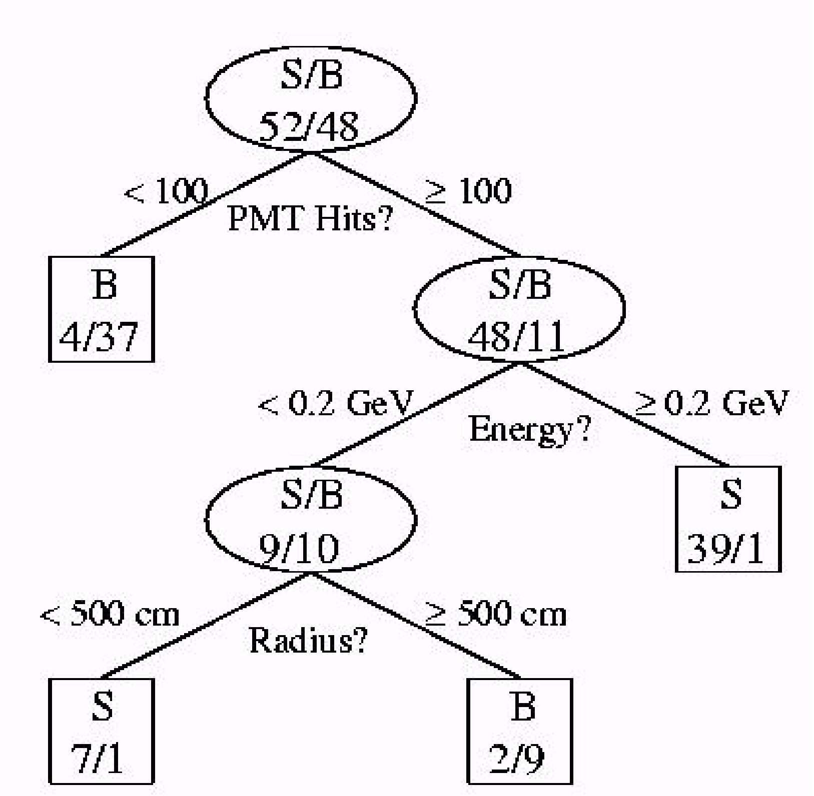
  * Hai-Jun Yang, Byron P. Roe, Ji Zhu, " Studies of boosted decision trees for MiniBooNE particle identification", physics/0508045, NIM A 555:370,2005

  * Byron P. Roe, Hai-Jun Yang, Ji Zhu, Yong Liu, Ion Stancu, Gordon McGregor," Boosted decision trees as an alternative to artificial neural networks for particle identification", NIM A 543:577,2005

  * Hai-Jun Yang, Byron P. Roe, Ji Zhu, "Studies of Stability and Robustness of Artificial Neural Networks and Boosted Decision Trees", physics/0610276.

How to Build A Decision Tree ?

1. Put all training events in root node, then try to select the splitting variable and splitting value which gives the best signal/background separation.
2. Training events are split into two parts, left and right, depending on the value of the splitting variable.
3. For each sub node, try to find the best variable and splitting point which gives the best separation.
4. If there are more than 1 sub node, pick one node with the best signal/background separation for next tree splitter.
5. Keep splitting until a given number of terminal nodes (leaves) are obtained, or until each leaf is pure signal/background, or has too few events to continue.



* If signal events are dominant in one leaf, then this leaf is signal leaf (+1); otherwise, backgroud leaf (score= -1).

Criterion for "Best" Tree Split

- Purity, *P*, is the fraction of the weight of a node (leaf) due to signal events.
- Gini Index: Note that Gini index is 0 for all signal or all background.

$$Gini = (\sum_{i=1}^{n} W_i) P(1-P)$$

- The criterion is to minimize

  Gini_left_node+ Gini_right_node.

Criterion for Next Node to Split

- Pick the node to maximize the change in Gini index. Criterion =

  $$\mathrm{Gini}_{\mathrm{parent\_node}} - \mathrm{Gini}_{\mathrm{right\_child\_node}} - \mathrm{Gini}_{\mathrm{left\_child\_node}}$$

- We can use Gini index contribution of tree split variables to sort the importance of input variables. (show example later)
- We can also sort the importance of input variables based on how often they are used as tree splitters. (show example later)

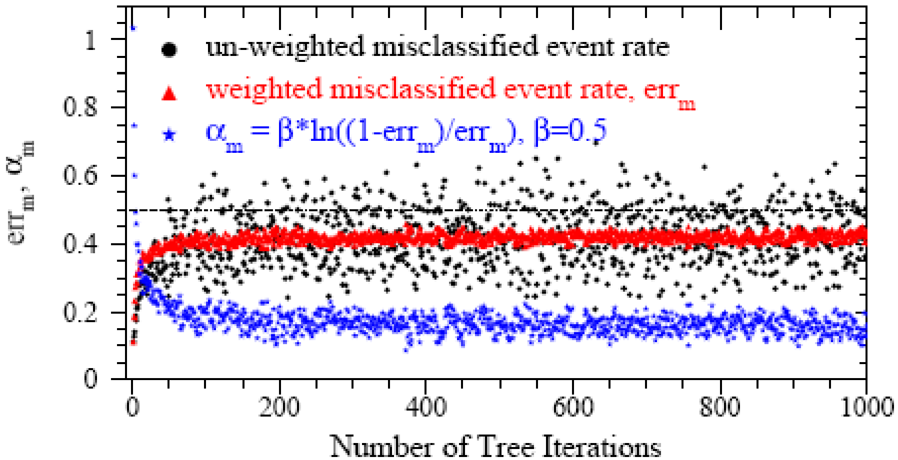
Signal and Background Leaves

- Assume an equal weight of signal and background training events.
- If event weight of signal is larger than ½ of the total weight of a leaf, it is a signal leaf; otherwise it is a background leaf.
- Signal events on a background leaf or background events on a signal leaf are misclassified events.

How to Boost Decision Trees ?

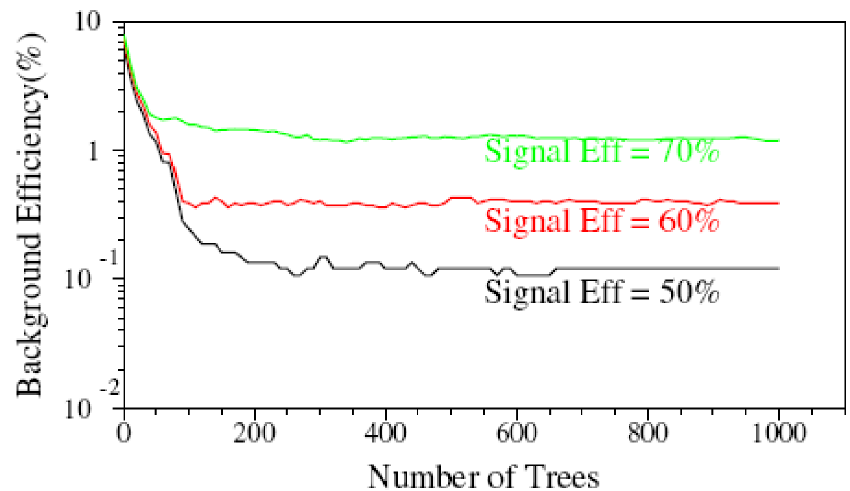
- ➔ For each tree iteration, same set of training events are used but the weights of misclassified events in previous iteration are increased (boosted). Events with higher weights have larger impact on Gini index values and Criterion values. The use of boosted weights for misclassified events makes them possible to be correctly classified in succeeding trees.
- ➔ Typically, one generates several hundred to thousand trees until the performance is optimal.
- ➔ The score of a testing event is assigned as follows: If it lands on a signal leaf, it is given a score of 1; otherwise -1. The sum of scores (weighted) from all trees is the final score of the event.

Weak → Powerful Classifier



➔ The advantage of using boosted decision trees is that it combines all decision trees, "weak" classifiers, to make a powerful classifier. The performance of BDT is stable after few hundred tree iterations.

➔ Boosted decision trees focus on the misclassified events which usually have high weights after hundreds of tree iterations. An individual tree has a very weak discriminating power; the weighted misclassified event rate err_m is about 0.4-0.45.



Two Boosting Algorithms

- AdaBoost Algorithm:

1. Initialize the observation weights $w_i = 1/n$, i = 1, 2,..., n
2. For m = 1 to M:
   2.a Fit a classifier $T_m(x)$ to the training data using weights w_i
   2.b Compute
   $$err_m = \frac{\sum_{i=1}^{n} w_i I(y_i \neq T_m(x_i))}{\sum_{i=1}^{n} w_i}$$
   2.c Compute $\alpha_m = \beta \times log((1 - err_m)/err_m)$
   2.d Set $w_i \leftarrow w_i \times exp(\alpha_m I(y_i \neq T_m(x_i)))$, i=1, 2,...,n
   2.e Re-normalize $w_i = w_i / \sum_{i=1}^{n} w_i$
3. Output T(x) $= \sum_{m=1}^{M} \alpha_m T_m(x)$

I = 1, if a training event is misclassified; Otherwise, I = 0

- $\epsilon-$boosting Algorithm:

1. Initialize the observation weights $w_i = 1/n$, i = 1, 2,..., n
2. For m = 1 to M:
   2.a Fit a classifier $T_m(x)$ to the training data using weights w_i
   2.b Set $w_i \leftarrow w_i \times exp(2\epsilon I(y_i \neq T_m(x_i)))$, i=1, 2,...,n
   2.c Re-normalize $w_i = w_i / \sum_{i=1}^{n} w_i$
3. Output T(x) $= \sum_{m=1}^{M} \epsilon T_m(x)$

Example

- **AdaBoost: the weight of misclassified events is increased by**
  - error rate=0.1 and β = 0.5, α_m = 1.1, exp(1.1) = 3
  - error rate=0.4 and β = 0.5, α_m = 0.203, exp(0.203) = 1.225
  - Weight of a misclassified event is multiplied by a large factor which depends on the error rate.
- **ε–boost: the weight of misclassified events is increased by**
  - If $\varepsilon = 0.01$, exp(2*0.01) = 1.02
  - If $\varepsilon = 0.04$, exp(2*0.04) = 1.083
  - It changes event weight a little at a time.

➔ AdaBoost converges faster than ε-boost. However, the performance of AdaBoost and ε–boost are very comparable with sufficient tree iterations.

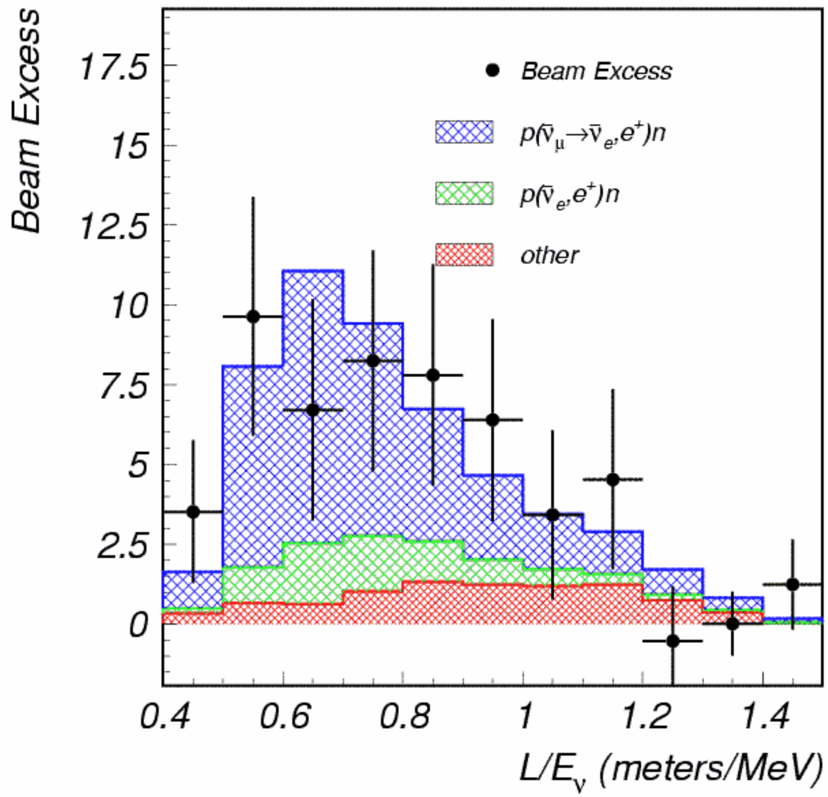
Application of ANN/BDT for MiniBooNE Experiment at Fermilab

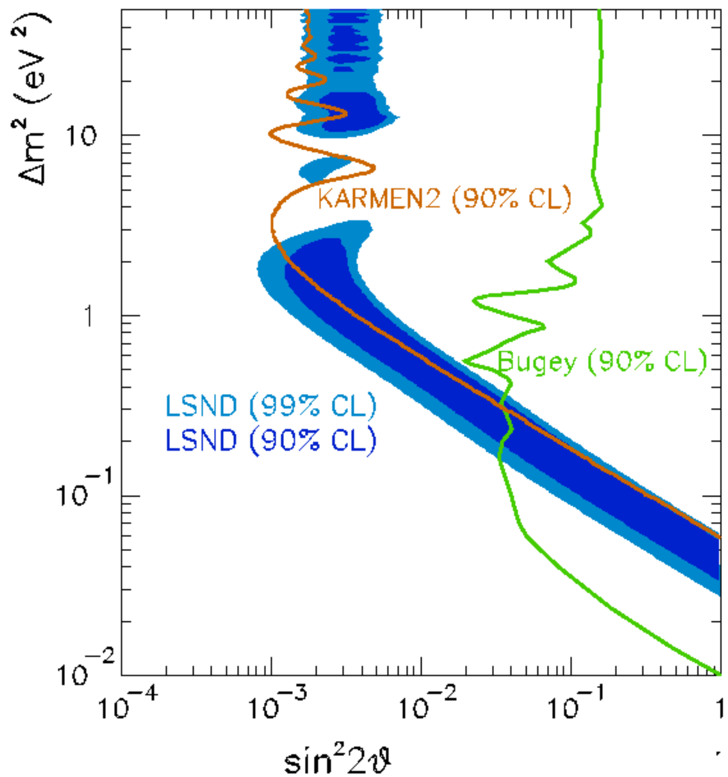
- Physics Motivation
- The MiniBooNE Experiment
- Particle Identification Using ANN/BDT

Physics Motivation

➔ LSND observed a positive signal(~4σ), but not confirmed.

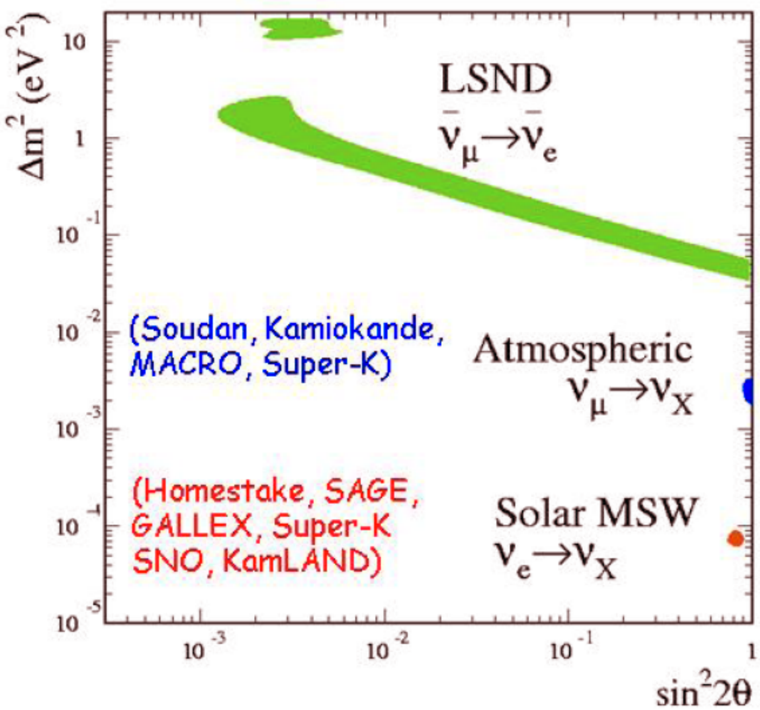
$$P(\bar{\nu}_\mu \to \bar{\nu}_e) = \sin^2(2\theta)\sin^2(\frac{1.27 L\Delta m^2}{E}) = (0.264 \pm 0.067 \pm 0.045)\%$$





Physics Motivation

State of Oscillation Results



- Simplest model has three neutrino mass eigenstates, but…
- Data indicates 3 mass differences
  - $\Delta m^2_{atm} \sim 2\text{-}3\times10^{-3}\ eV^2$
  - $\Delta m^2_{sol} \sim 7\times10^{-5}\ eV^2$
  - $\Delta m^2_{LSND} \sim .1\text{-}10\ eV^2$

$$\Delta m^2_{atm} + \Delta m^2_{sol} \neq \Delta m^2_{lsnd}$$

➔ If the LSND signal does exist, it will imply new physics beyond SM.

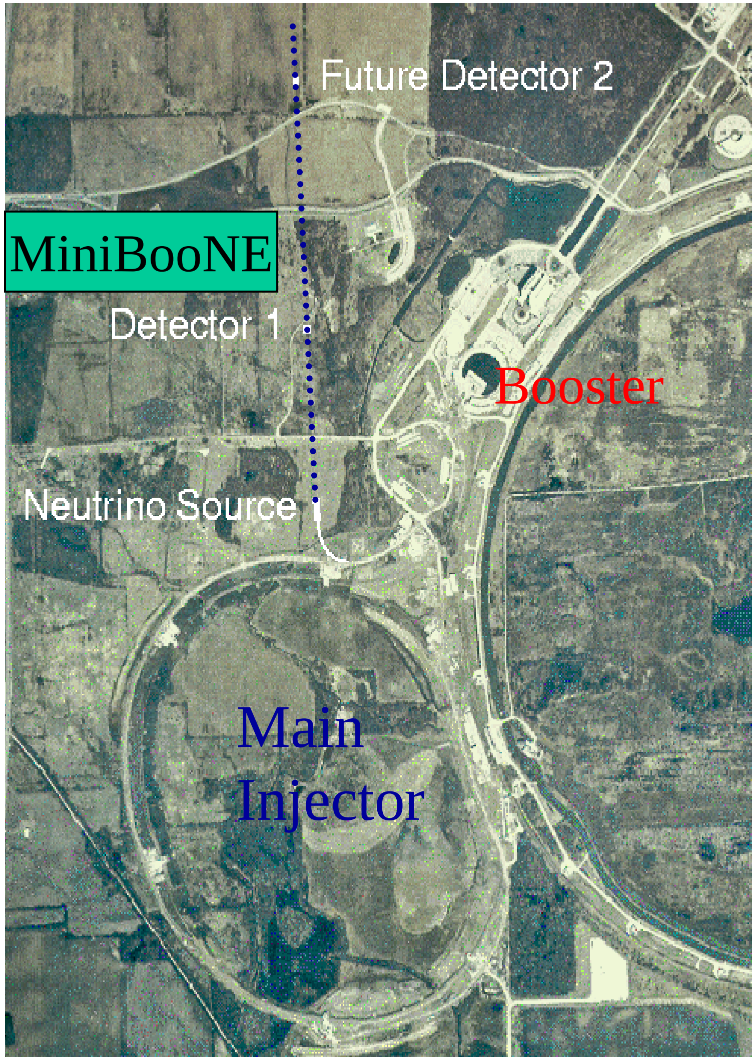
➔ The MiniBooNE is designed to confirm or refute LSND oscillation result at $\Delta m^2 \sim 1.0\ eV^2$.

The MiniBooNE Collaboration

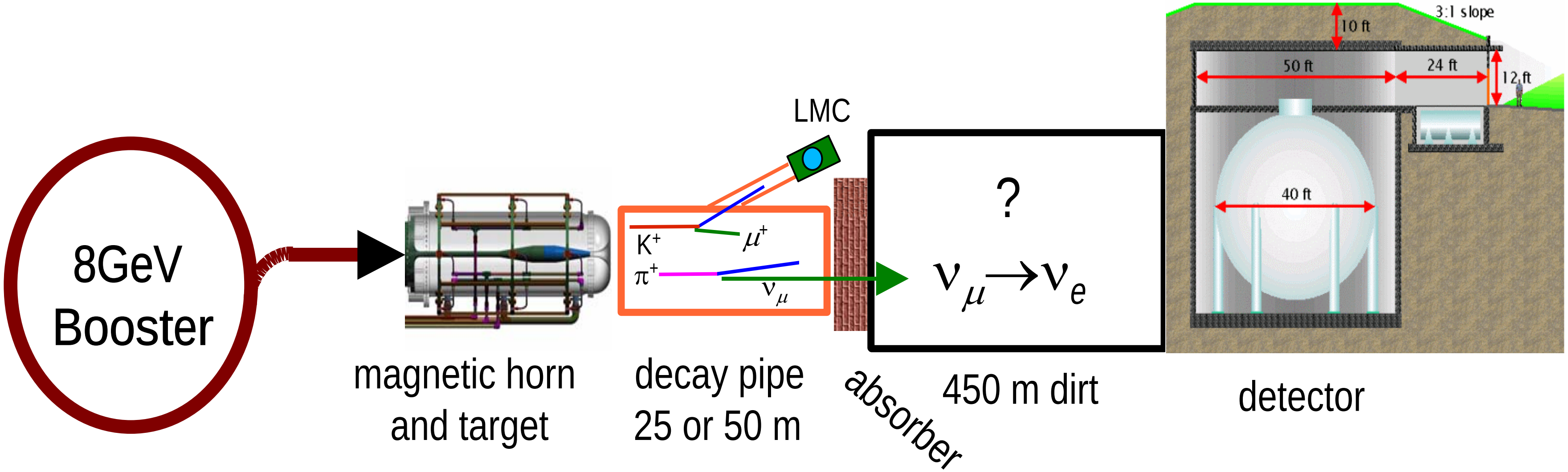
Y.Liu, D.Perevalov, I.Stancu
University of Alabama
S.Koutsoliotas
Bucknell University
R.A.Johnson, J.L.Raaf
University of Cincinnati
T.Hart, R.H.Nelson, M.Tzanov
M.Wilking, E.D.Zimmerman
University of Colorado
A.A.Aguilar-Arevalo, L.Bugel L.Coney,
J.M.Conrad, Z. Djurcic,
K.B.M.Mahn, J.Monroe, D.Schmitz
M.H.Shaevitz, M.Sorel, G.P.Zeller
Columbia University
D.Smith
Embry Riddle Aeronautical University
L.Bartoszek, C.Bhat, S.J.Brice
B.C.Brown, D. A. Finley, R.Ford,
F.G.Garcia, P.Kasper, T.Kobilarcik,
I.Kourbanis, A.Malensek, W.Marsh,
P.Martin, F.Mills, C.Moore, E.Prebys,
A.D.Russell , P.Spentzouris,
R.J.Stefanski, T.Williams
Fermi National Accelerator Laboratory
D.C.Cox, T.Katori, H.Meyer, C.C.Polly
R.Tayloe
Indiana University

G.T.Garvey, A.Green, C.Green, W.C.Louis, G.McGregor, S.McKenney
G.B.Mills, H.Ray, V.Sandberg, B.Sapp, R.Schirato, R.Van de Water
N.L.Walbridge, D.H.White
Los Alamos National Laboratory
R.Imlay, W.Metcalf, S.Ouedraogo, M.O.Wascko
Louisiana State University
J.Cao, Y.Liu, B.P.Roe, H.J.Yang
University of Michigan
A.O.Bazarko, P.D.Meyers, R.B.Patterson, F.C.Shoemaker, H.A.Tanaka
Princeton University
P.Nienaber Saint Mary's University of Minnesota
J. M. Link Virginia Polytechnic Institute and State University
E.Hawker Western Illinois University
A.Curioni, B.T.Fleming Yale University

Fermilab Booster



The MiniBooNE Experiment



- The FNAL Booster delivers 8 GeV protons to the MiniBooNE beamline.
- The protons hit a 71cm beryllium target producing pions and kaons.
- The magnetic horn focuses the secondary particles towards the detector.
- The mesons decay into neutrinos, and the neutrinos fly to the detector, all other secondary particles are absorbed by absorber and 450 m dirt.
- 5.7E20 POT for neutrino mode since 2002.
- Switch horn polarity to run anti-neutrino mode since January 2006.

MiniBooNE Flux

8 GeV protons on Be target gives:

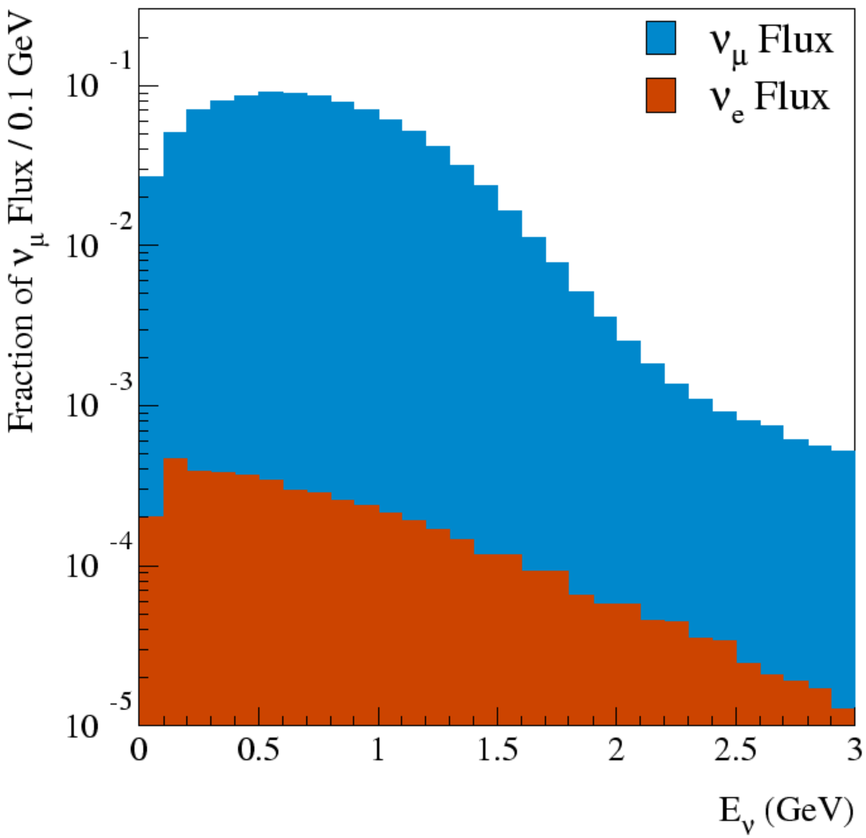
p + Be $\rightarrow \pi^+$, K^+ , K^0_L

ν_μ from:

$\pi^+ \rightarrow \mu^+ \nu_\mu$ $\quad K^+ \rightarrow \mu^+ \nu_\mu$ $\quad K^0_L \rightarrow \pi^- \mu^+ \nu_\mu$

Intrinsic ν_e from:

$\mu^+ \rightarrow e^+ \nu_e \bar{\nu}_\mu$ $\quad K^+ \rightarrow \pi^0 e^+ \nu_e$ $\quad K^0_L \rightarrow \pi^- e^+ \nu_e$

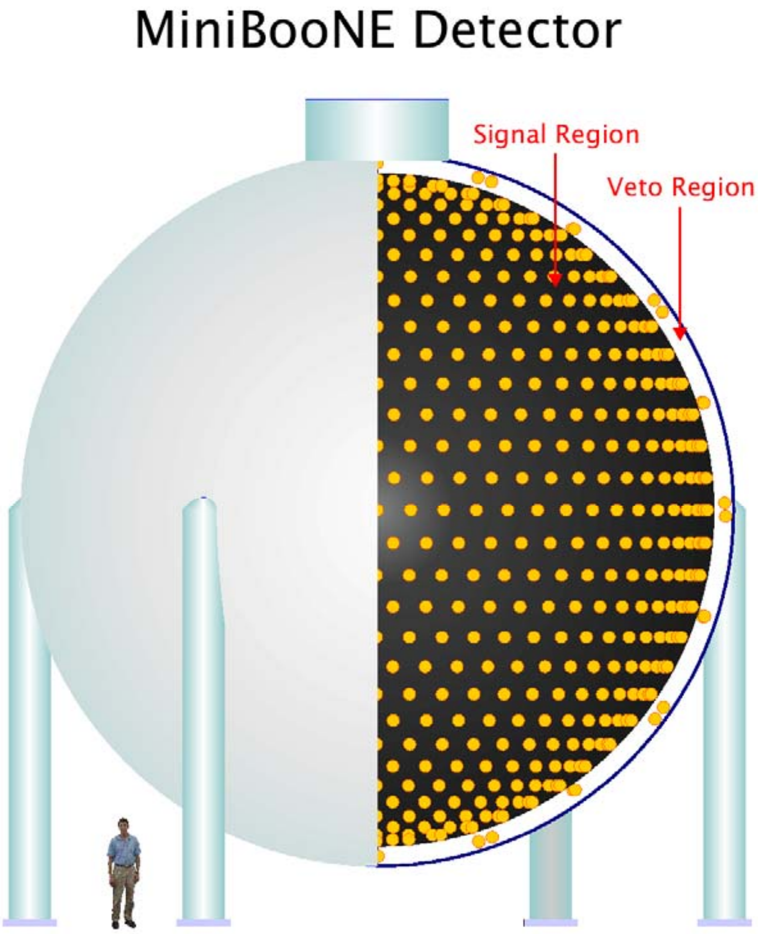


The intrinsic ν_e is ~0.5% of the neutrino Flux, it's one of major backgrounds for $\nu_\mu \rightarrow \nu_e$ search.

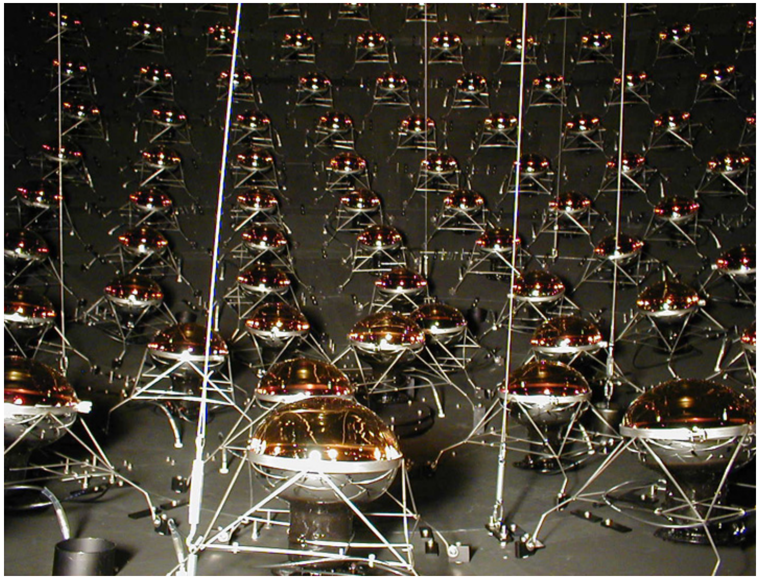
$$P(\nu_\mu \rightarrow \nu_e) = \sin^2(2\theta)\sin^2\left(\frac{1.27 L \Delta m^2}{E}\right)$$

L(m), E(MeV), $\Delta m^2(eV^2)$

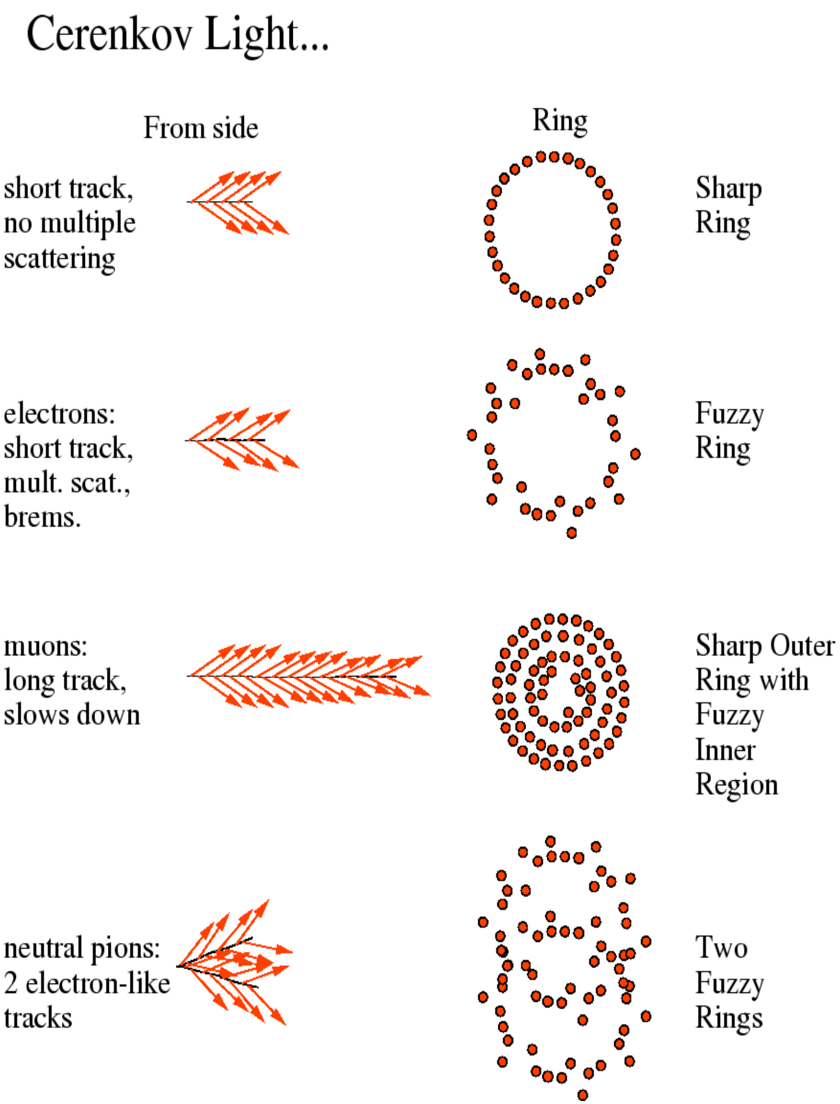
The MiniBooNE Detector

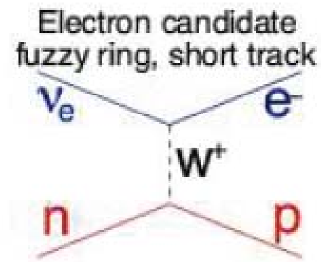


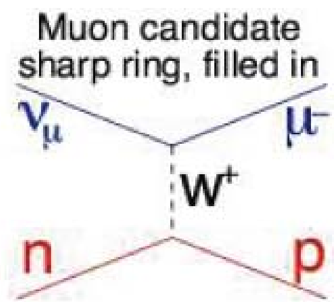
- 12m diameter tank
- Filled with 800 tons of ultra pure mineral oil
- Optically isolated inner region with 1280 PMTs
- Outer veto region with 240 PMTs.

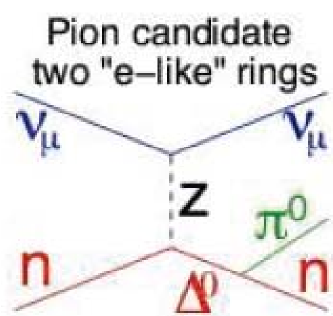


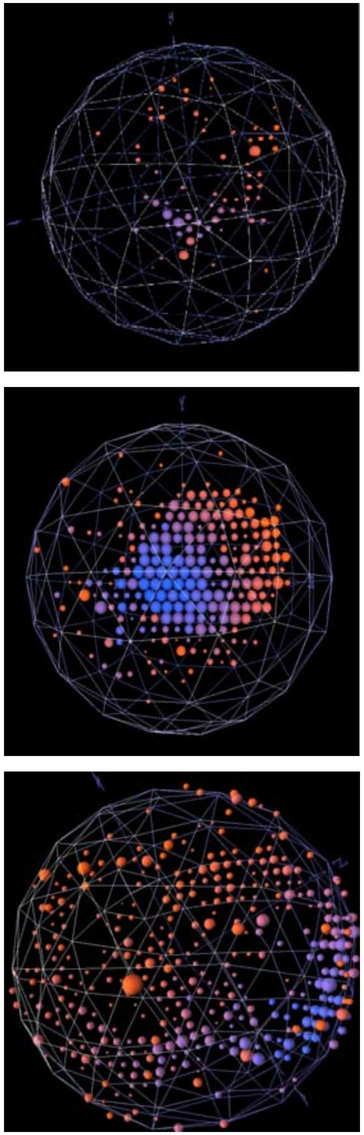
Event Topology



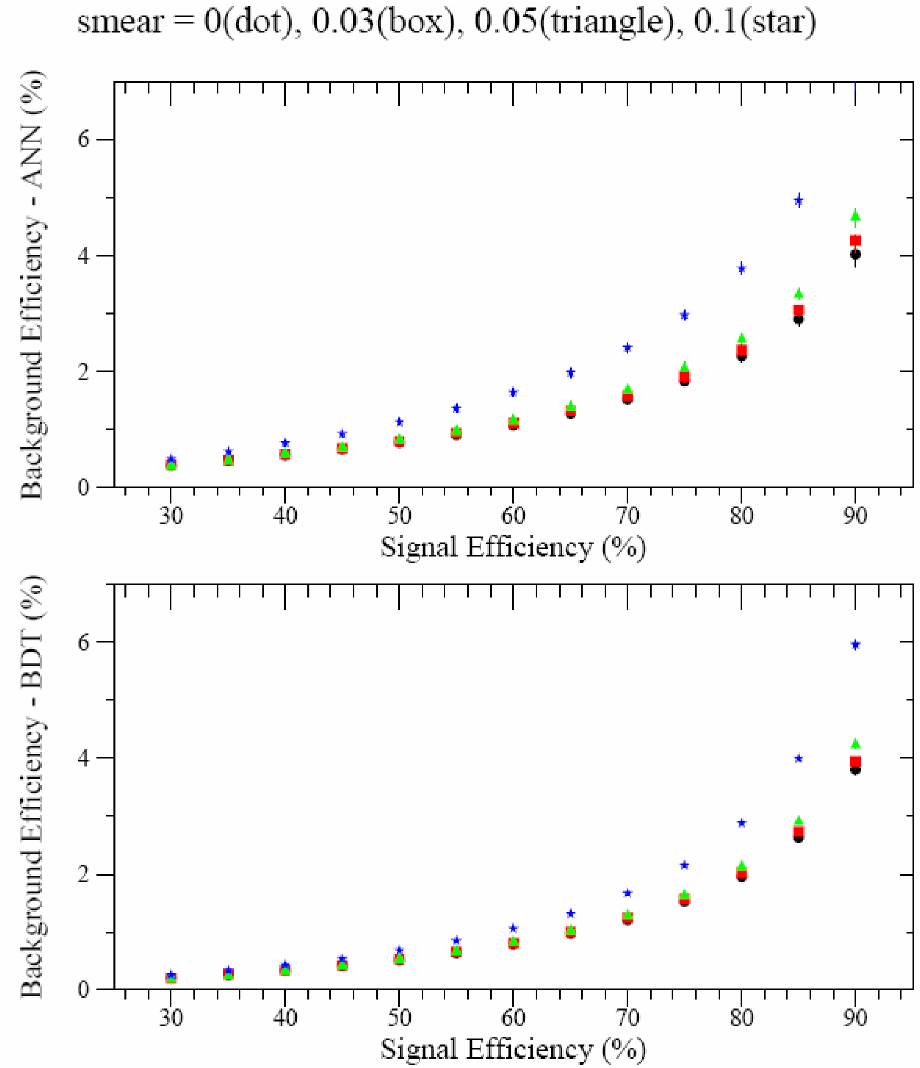








ANN vs BDT-Performance/Stability



➔30 variables for training

➔10 Training Samples(30k/30k): selected randomly from 50000 signal and 80000 background events.

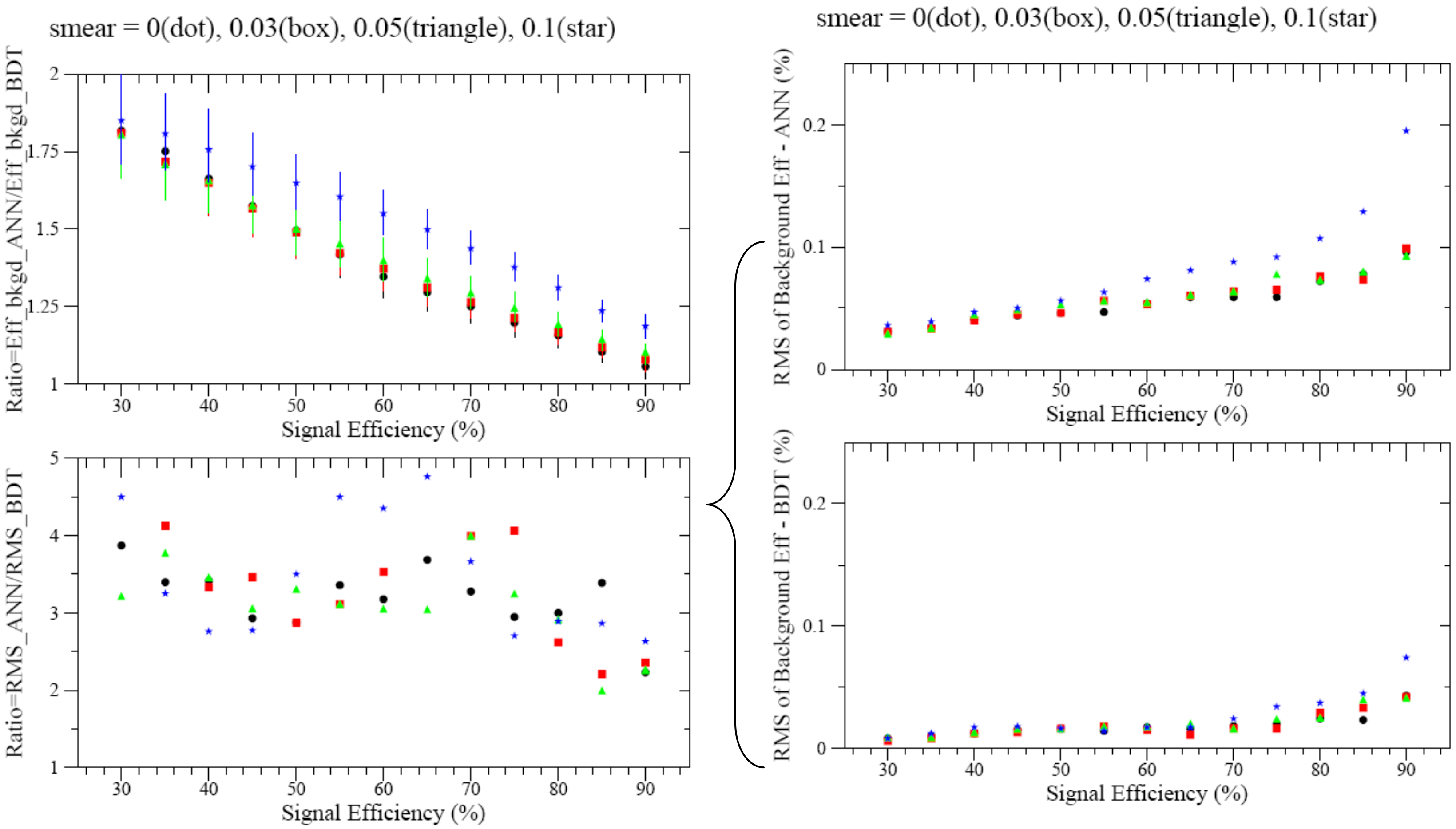
➔Testing Sample: 54291 signal and 166630 background

➔Smearing Testing Sample: Each Variable and testing event is smeared randomly using the formula,

V_ij = V_ij * (1 + smear*Rand_ij)

Where Rand_ij is random number with normal Gaussian distribution.

ANN vs BDT-Performance/Stability

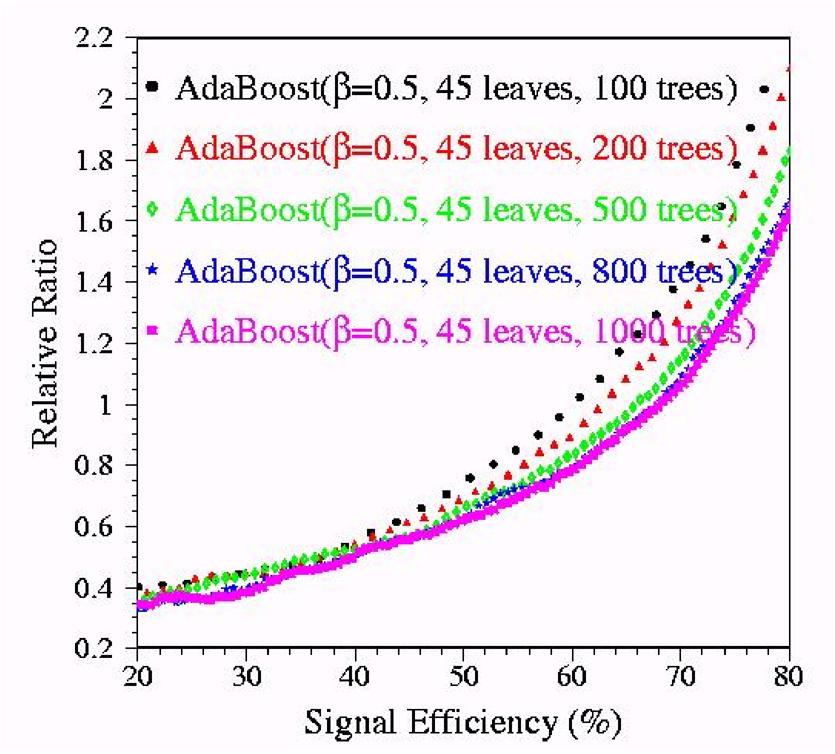


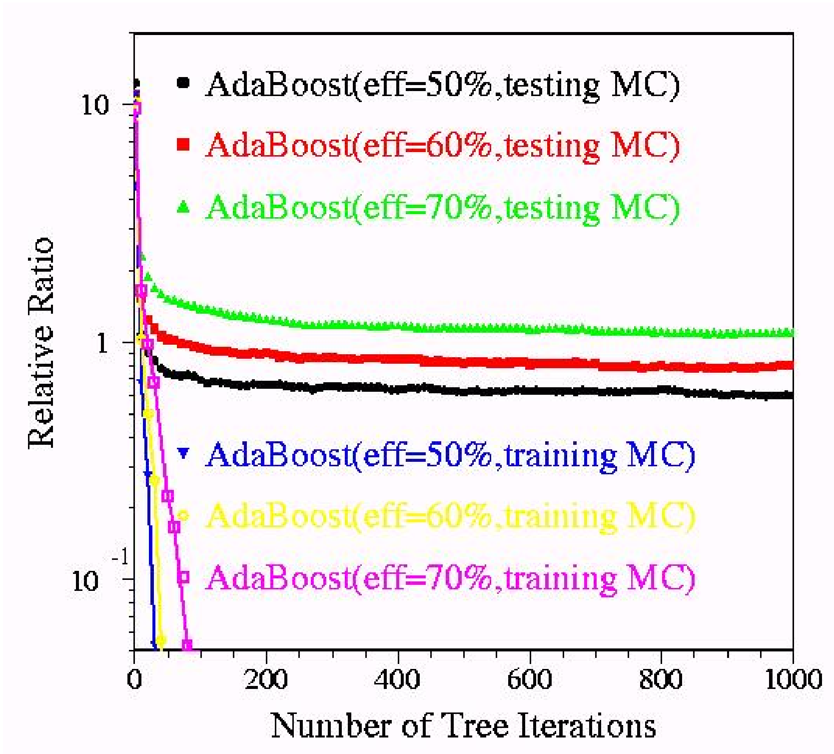
➔ BDT is more powerful and stable than ANN !

Effect of Tree Iterations

➔ It varies from analysis to analysis, depends on the training and testing samples. For MiniBooNE MC samples (52 input variables), we found ~1000 tree iterations works well.

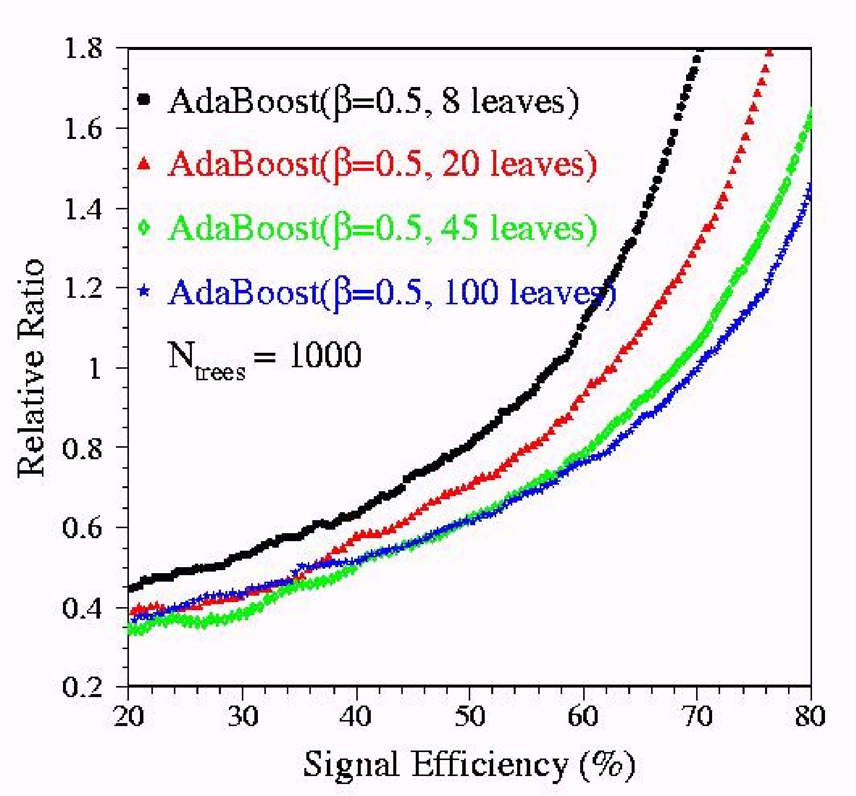
Relative Ratio = Background Eff / Signal Eff × Constant





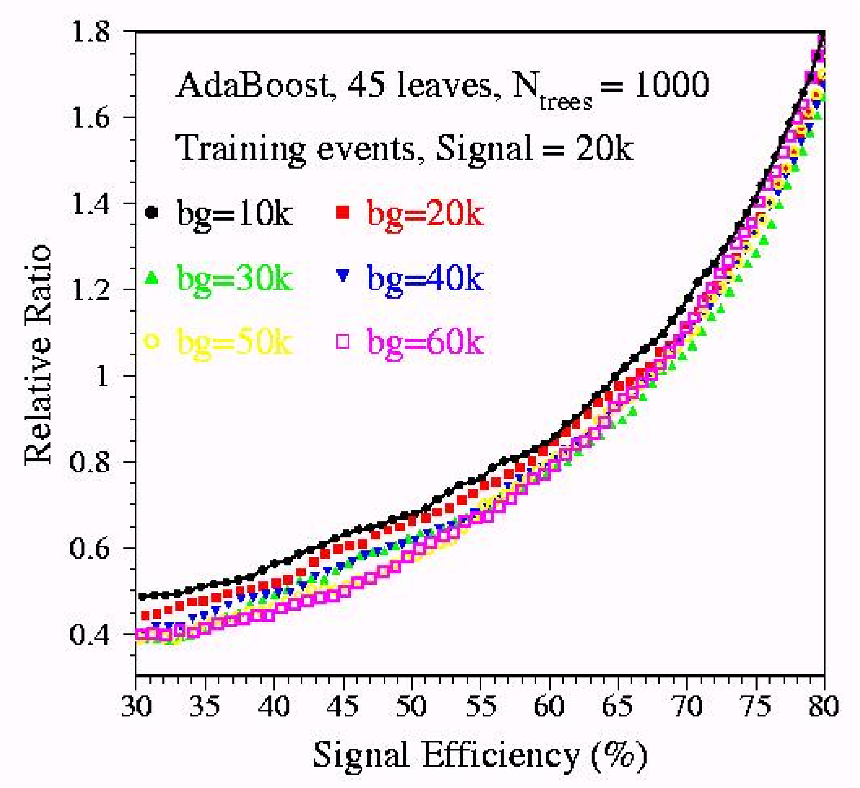
Effect of Decision Tree Size

- ➔ Statistical literature suggests 4 – 8 leaves per decision tree, we found larger tree size works significantly better than BDT with a small tree size using MiniBooNE MC.
- ➔ The MC events are described by 52 input variables. If the size of decision tree is small, only small fraction of variables can be used for each tree, so the decision tree cannot be fully developed to capture the overall signature of the MC events.



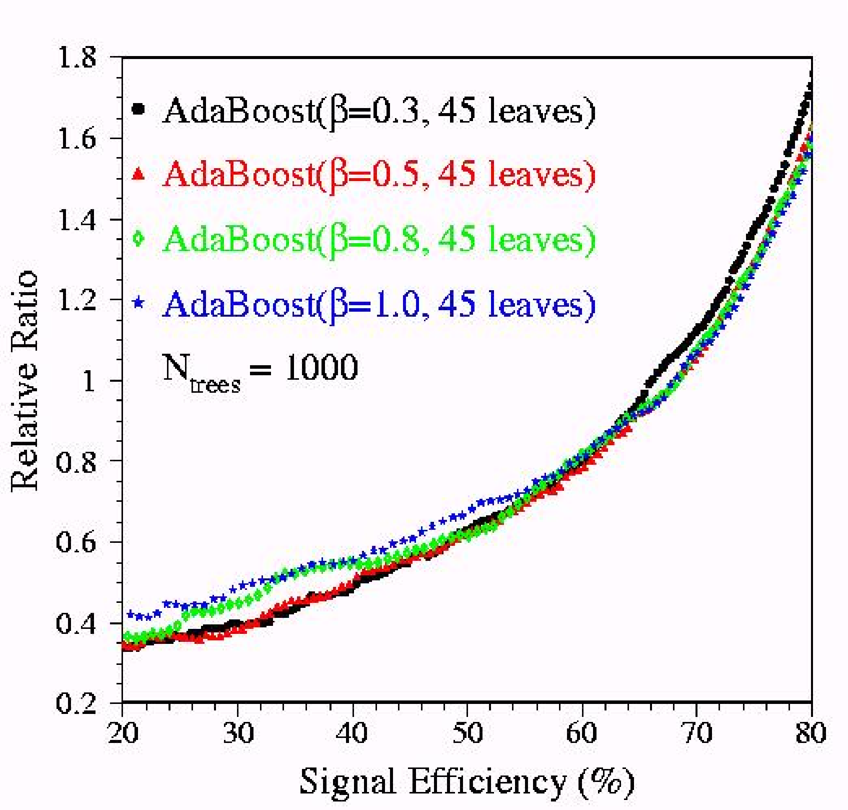
Effect of Training Events

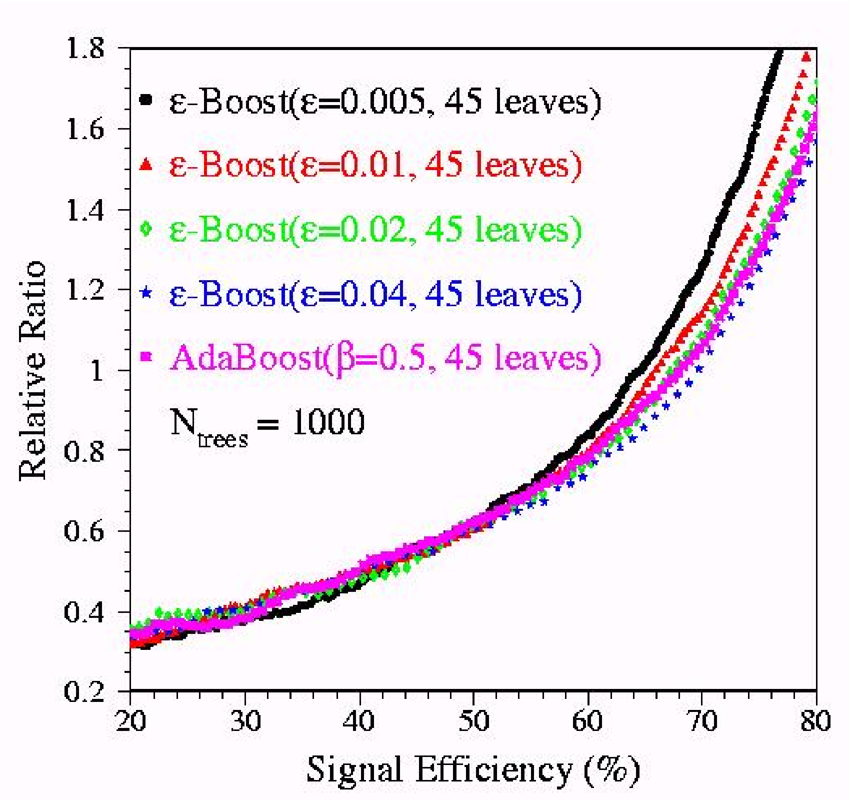
- ➔ Generally, more training events are preferred. For MiniBooNE MC samples, the use of 10-20K signal events, 30K or more background events works fairly well. Fewer background events for training degrades the boosting PID Performance.



Tuning Beta (β) and Epsilon (ε)

➔ β (AdaBoost) and Epsilon (ε-boost) are parameters to tune the weighting update rate, hence the speed of boosting convergence. β= 0.5, ε = 0.04 works well for MiniBooNE MC samples.

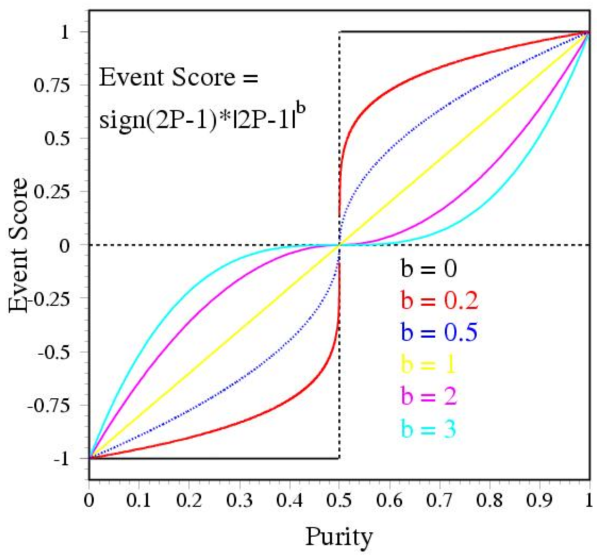


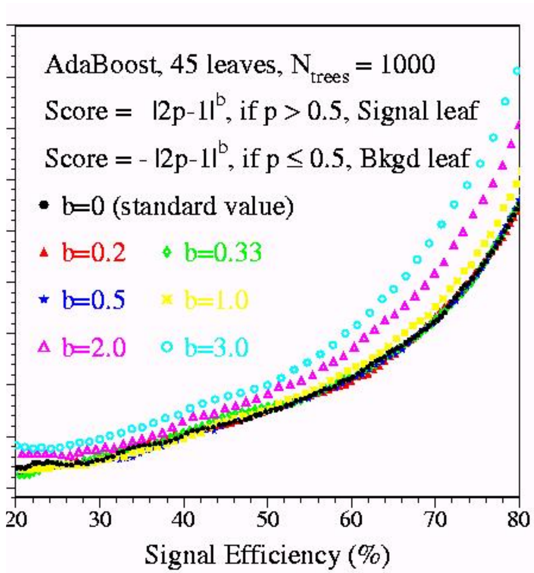


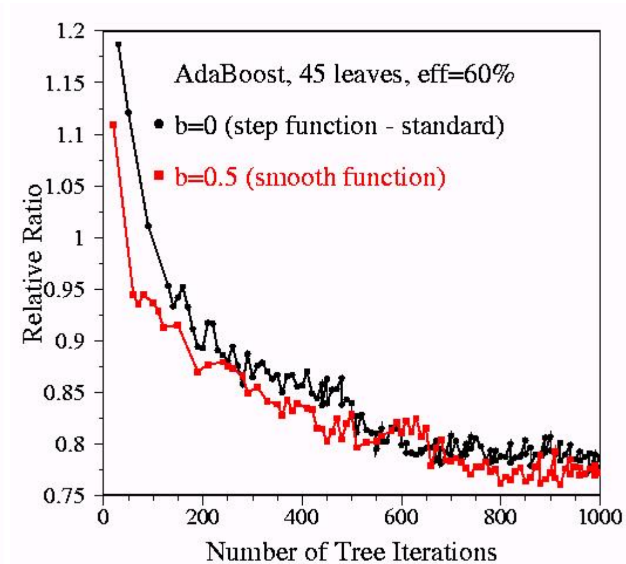
Soft Scoring Functions

- In standard boost, the score for an event from an individual tree is a simple step function depending on the purity of the leaf on which the event lands. If the purity is greater than 0.5, the score is 1 and otherwise it is -1. Is it optimal ? If the purity of a leaf is 0.51, should the score be the same as if the purity were 0.99?

➔ For a smooth function ($score=sign(2P\text{-}1)\times|2P\text{-}1|^b$) with b=0.5, AdaBoost performance converges faster than the original AdaBoost for the first few hundred trees. However the ultimate performances are comparable.

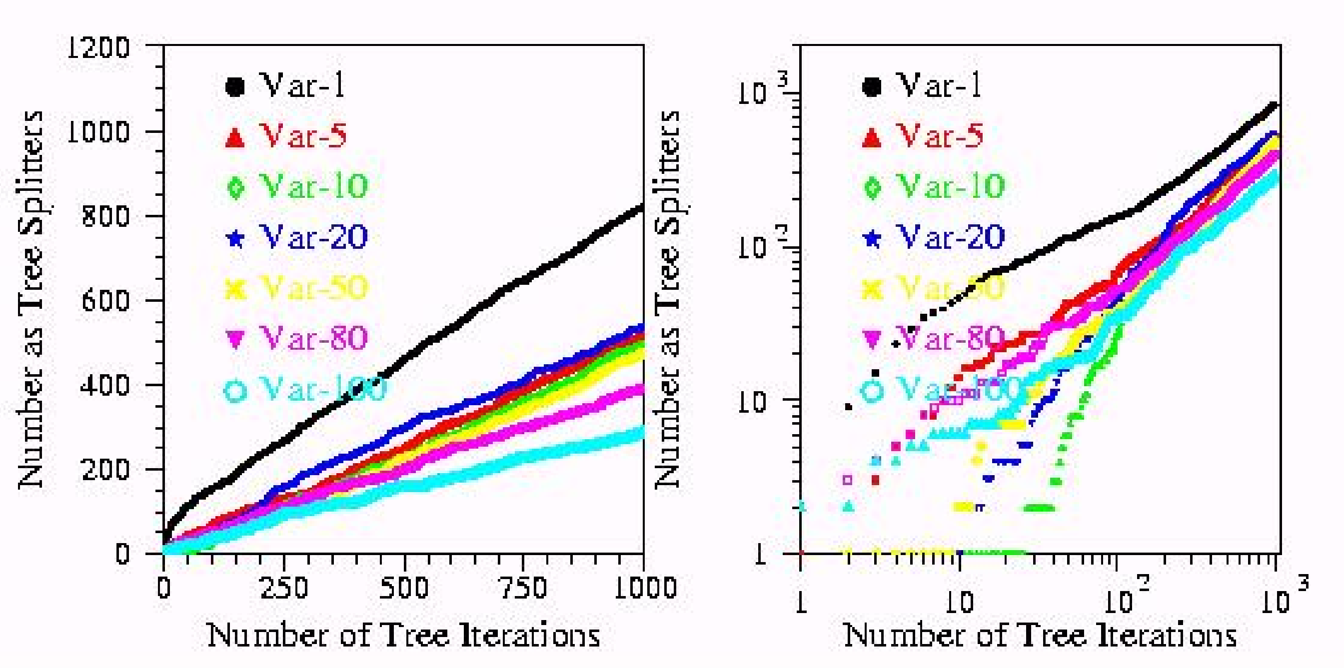






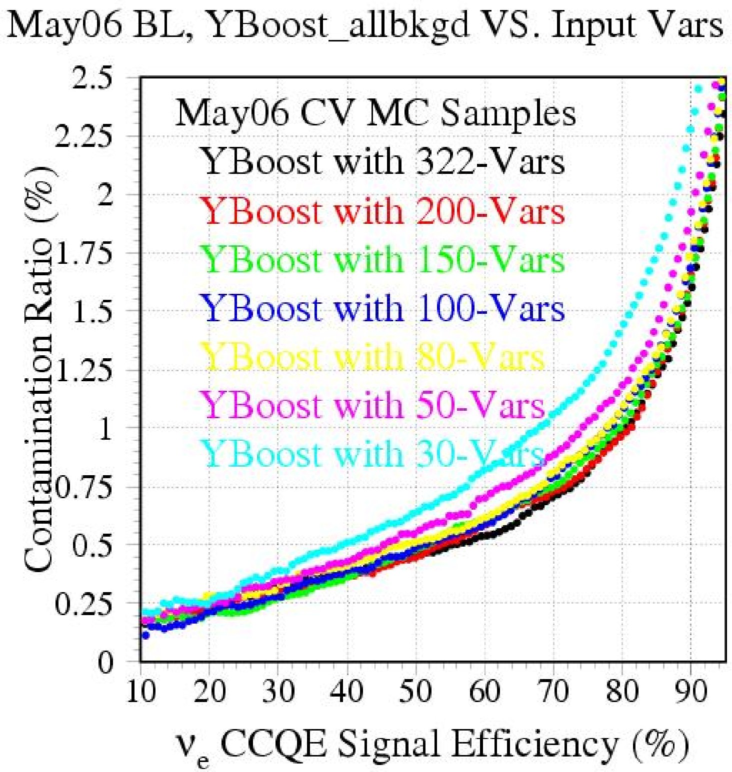
How to Select Input Variables ?

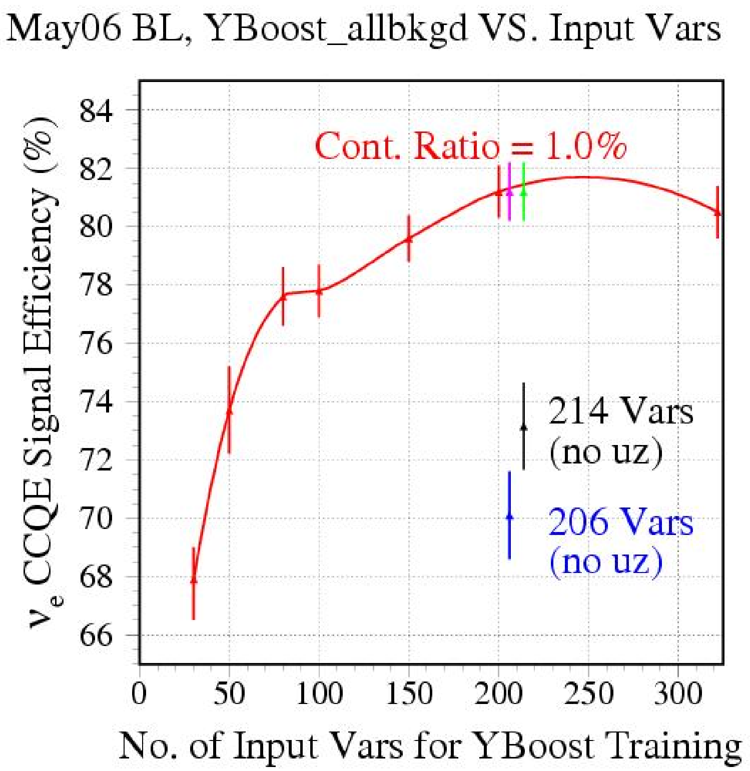
- ➔ The boosted decision trees can be used to select the most powerful variables to maximize the performance. The effectiveness of the input variables was rated based on how many times they were used as tree splitters, or which variables were used earlier than others, or their Gini index contributions. The performance are comparable for different rating techniques.
- ➔ Some input variables look useless by eyes may turn out to be quite useful for boosted decision trees.



How to Select Input Variables ?

- ➔ The boosting performance steadily improves with more input variables until ~ 200 for MiniBooNE MC samples. Adding further input variables (relative weak) doesn't improve and may slightly degrade the boosting performance.
- ➔ The main reason for the degradation is that there is no further useful information in the additional variables and these variables can be treated as "noise" variables for the boosting training.

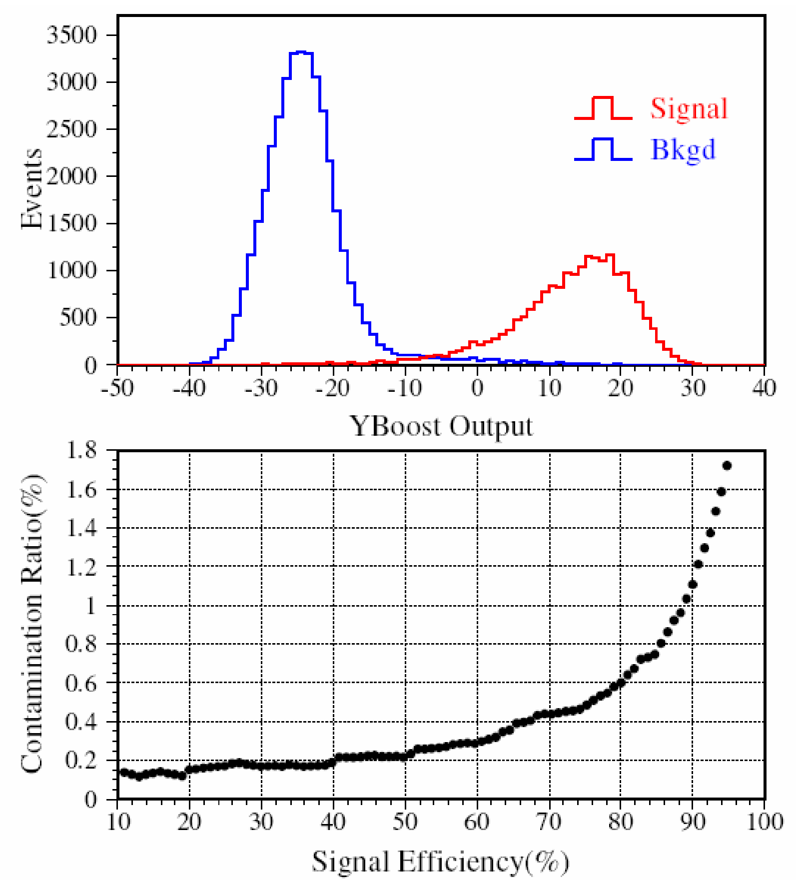




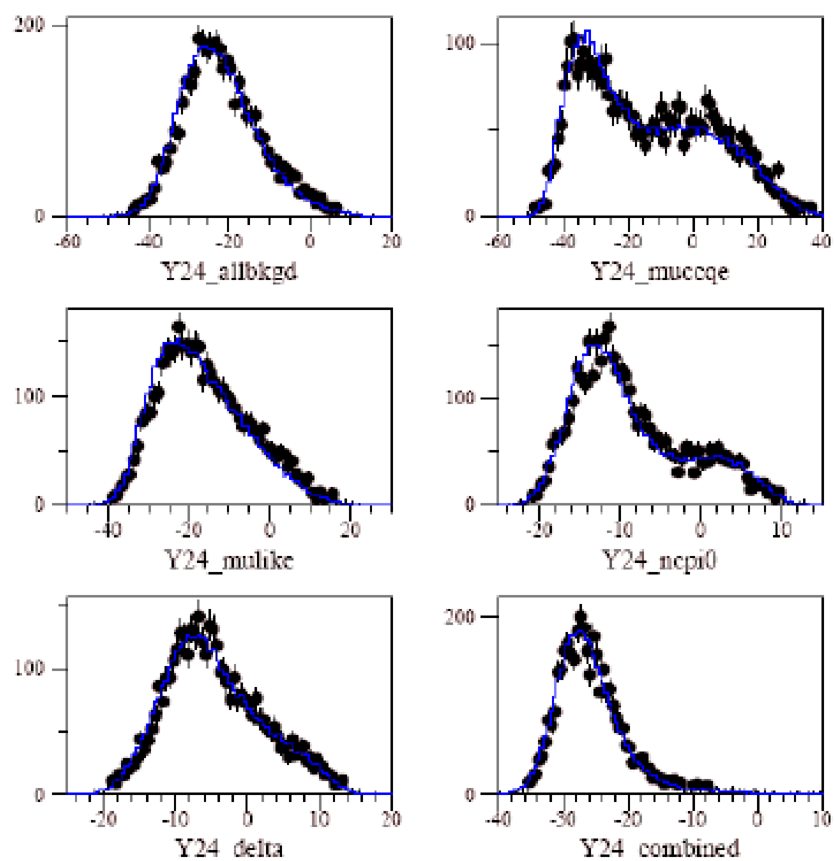
Output of Boosted Decision Trees

Osc ν_e CCQE vs All Background

MC vs ν_μ Data



1subevt,thits>200,vhits<6,R<500 cm,0.1<Efull<1.2 GeV,Y21<-5

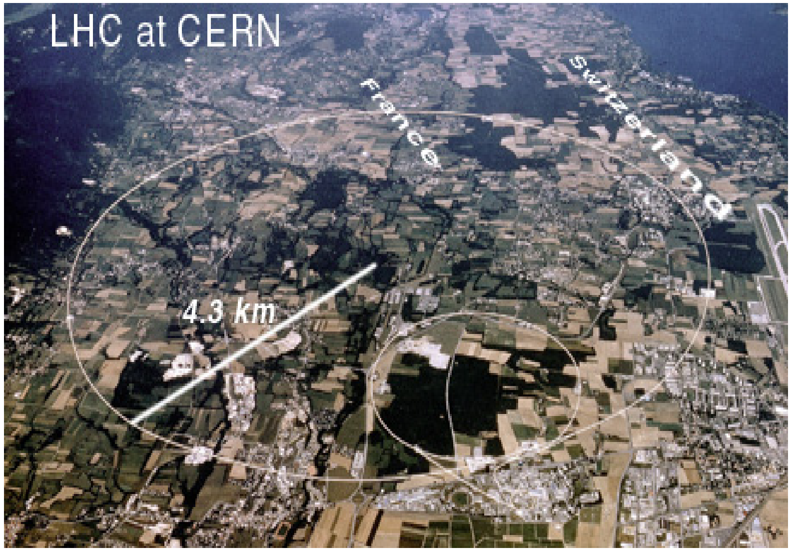


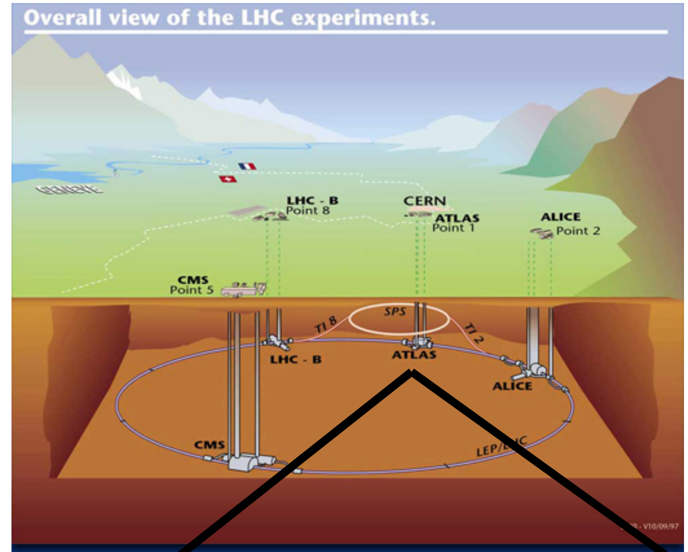
Application of ANN and BDT for ATLAS Di-Boson Analysis

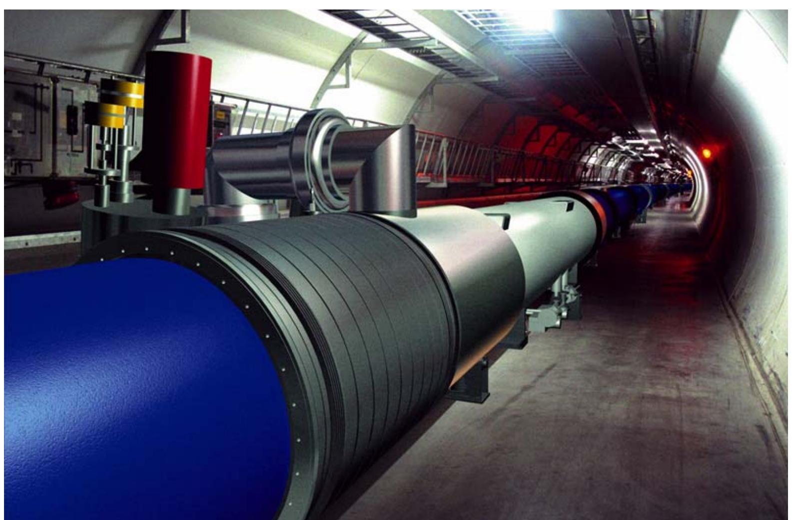
(H.J. Yang, Z.G. Zhao, B. Zhou)

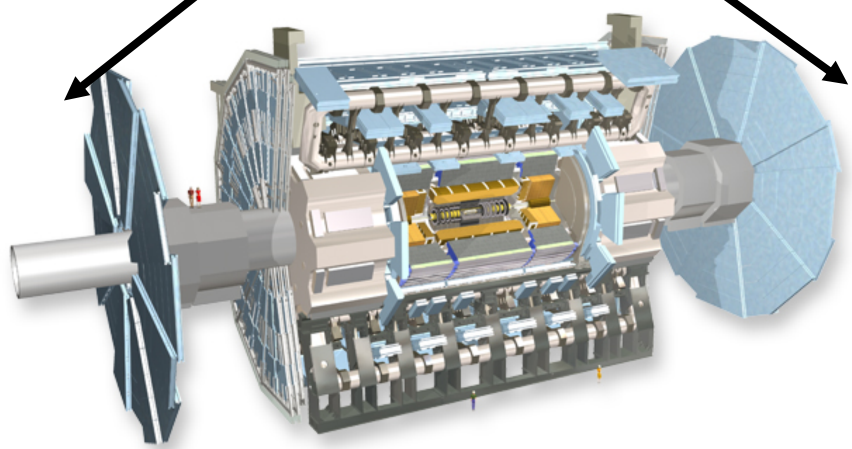
- ATLAS at CERN
- Physics Motivation
- ANN/BDT for Di-Boson Analysis

ATLAS at CERN





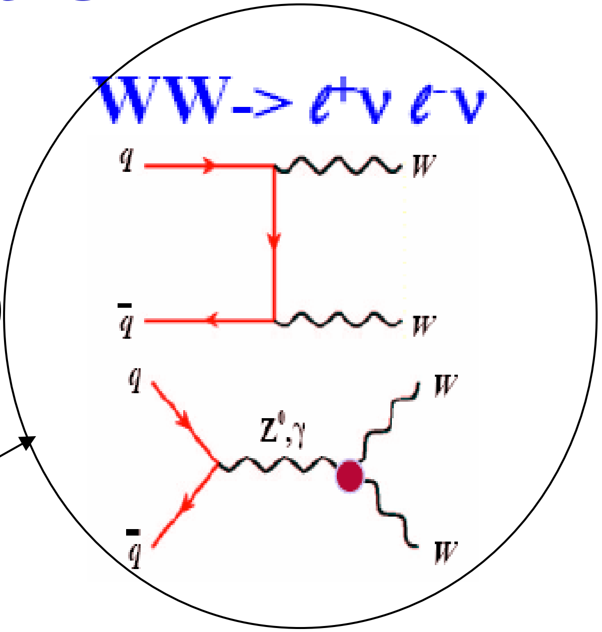




ATLAS Experiment

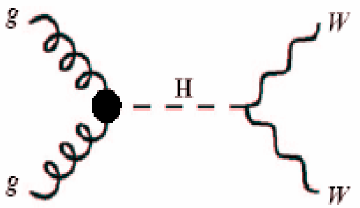
- **ATLAS** is a particle physics experiment that will explore the fundamental nature of matter and the basic forces that shape our universe.
- **ATLAS** detector will search for new discoveries in the head on collisions of protons of very high energy (14 TeV).
- **ATLAS** is one of the largest collaborations ever in the physical sciences. There are ~1800 physicists participating from more than 150 universities and laboratories in 35 countries.
- **ATLAS** is expected to begin taking data in 2007.

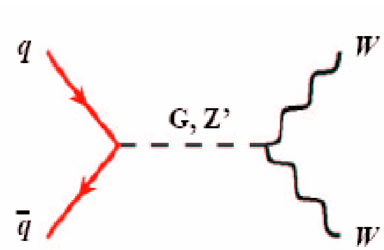
Physics Motivation



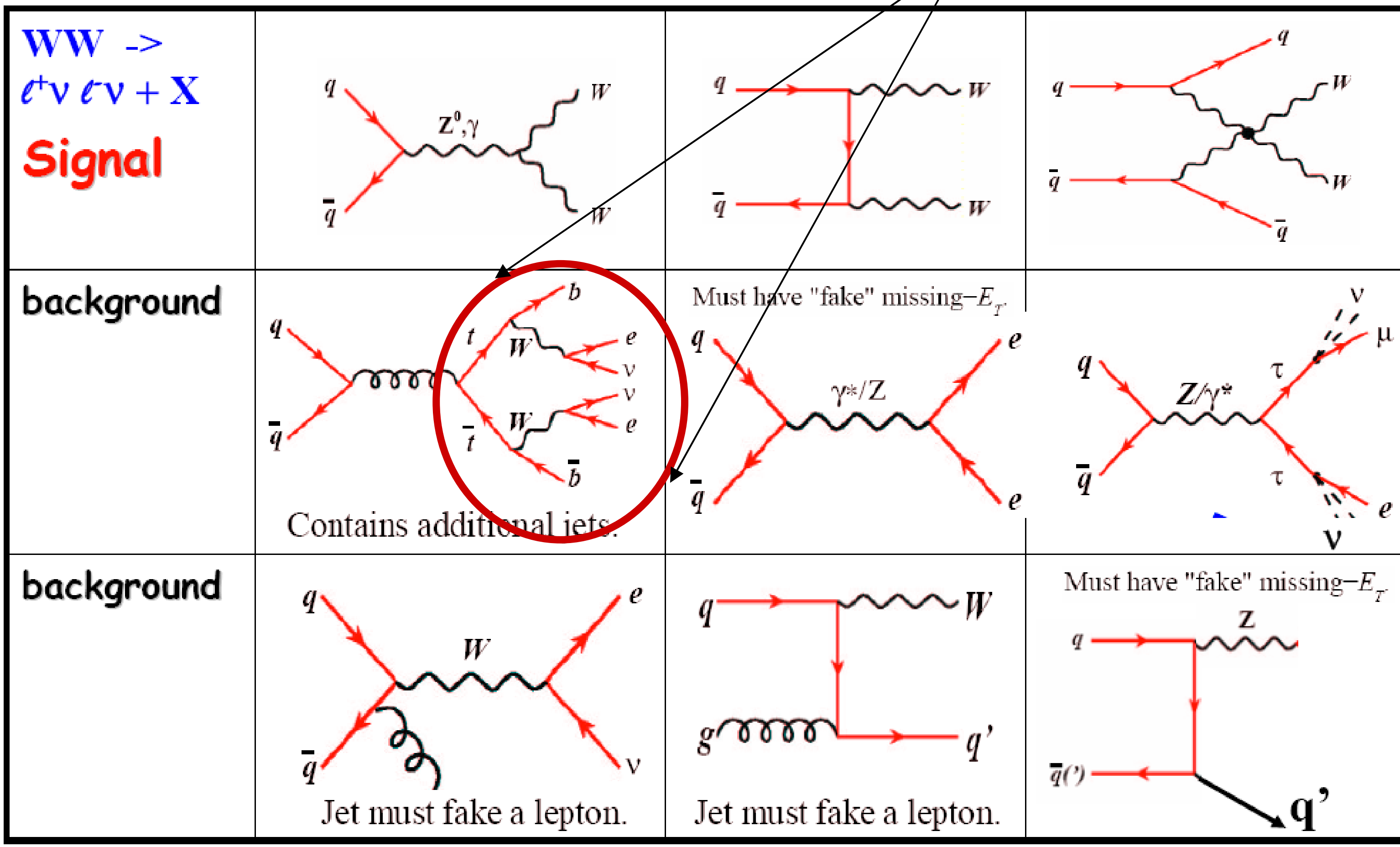
- Standard Model
  - Di-Boson (WW, ZW, ZZ, W γ, Z γ etc.)
  - to measure triple-gauge-boson couplings, ZWW and γWW etc.
  - Example: WW leptonic decay

- New Physics
  - to discover and measure Higgs → WW
  - to discover and measure G, Z’ → WW
  - More …





WW signal and background



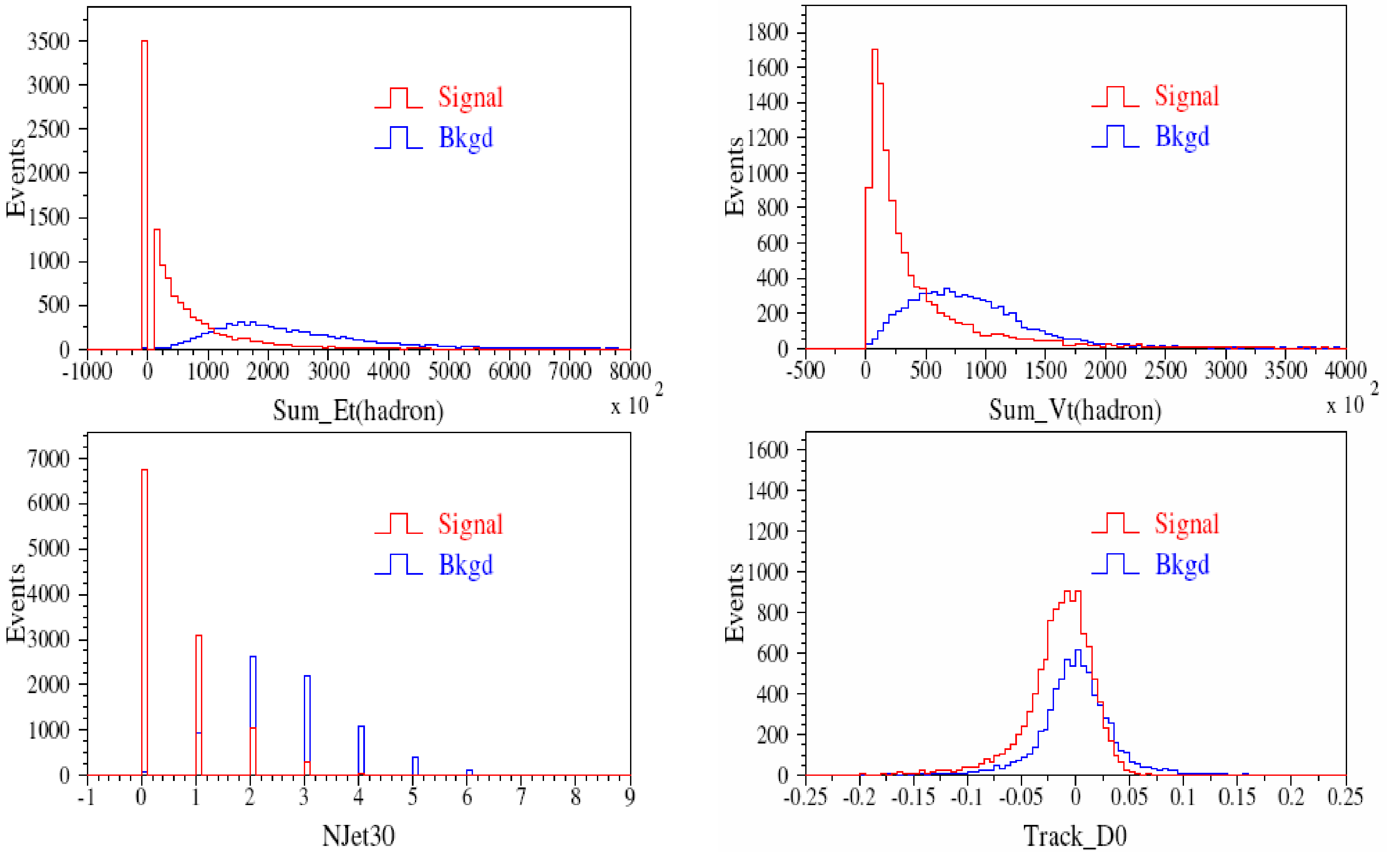
Background rates are of the order of 3-4 higher than the signal

WW (eμX) vs tt (background)

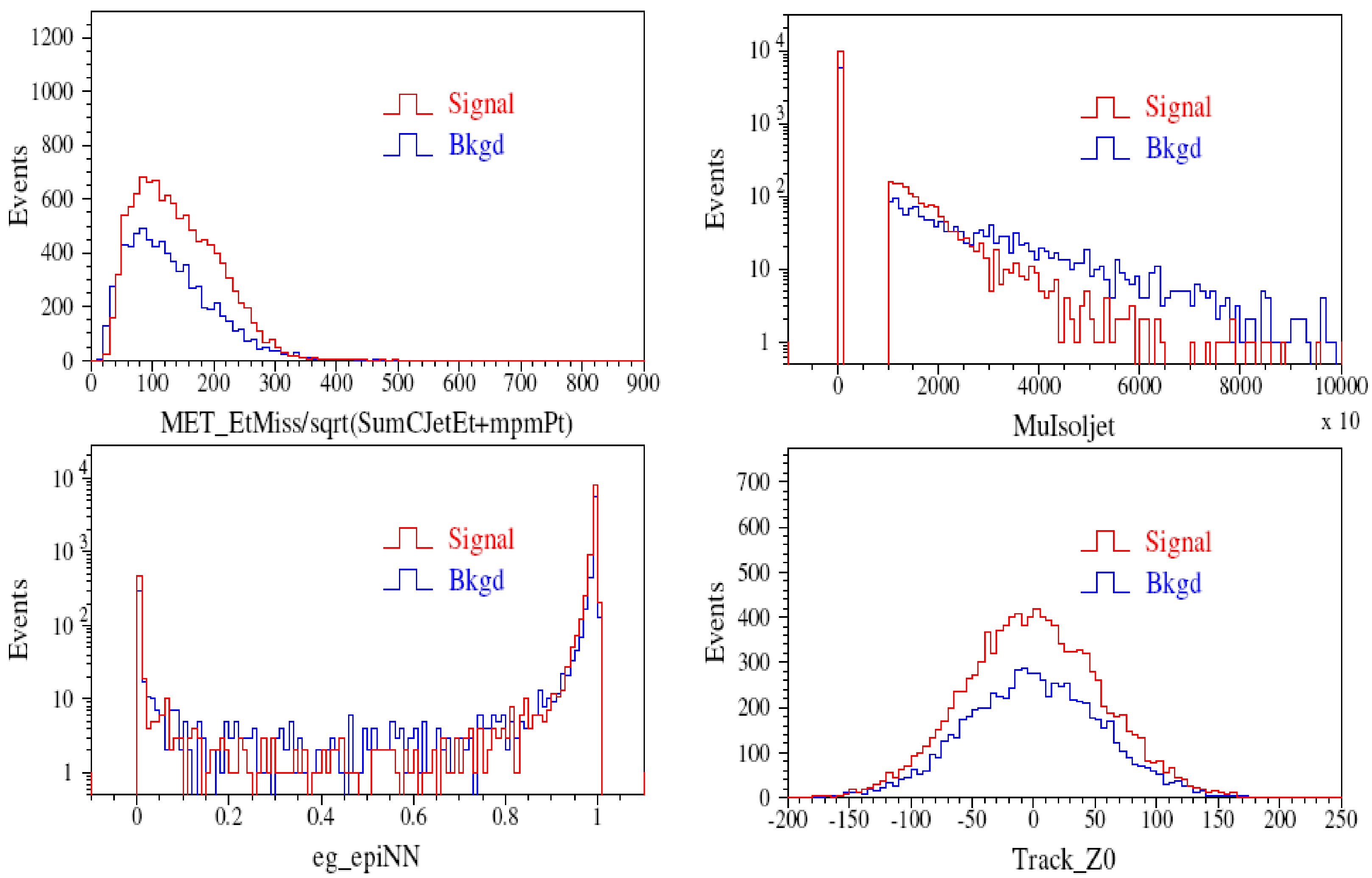
- Preselection Cuts:
  - e, μ with Pt > 10 GeV,
  - Missing Et > 15 GeV
  - Signal: WW → eμX, 47050 →18233 (Eff = 38.75%)
  - Background: tt, 433100 →14426 (Eff = 3.33%)
- All 48 input variables for ANN/BDT training
- Training Events – selected randomly
  - 7000 signal and 7000 background events for training
  - To produce ANN weights and BDT Tree index data file, which will be used for testing.
- Testing Events – the rest events for test
  - 11233 signal and 7426 background events
- More MC signal and background events will be used for ANN/BDT training and testing to obtain better results.

Some Powerful Variables

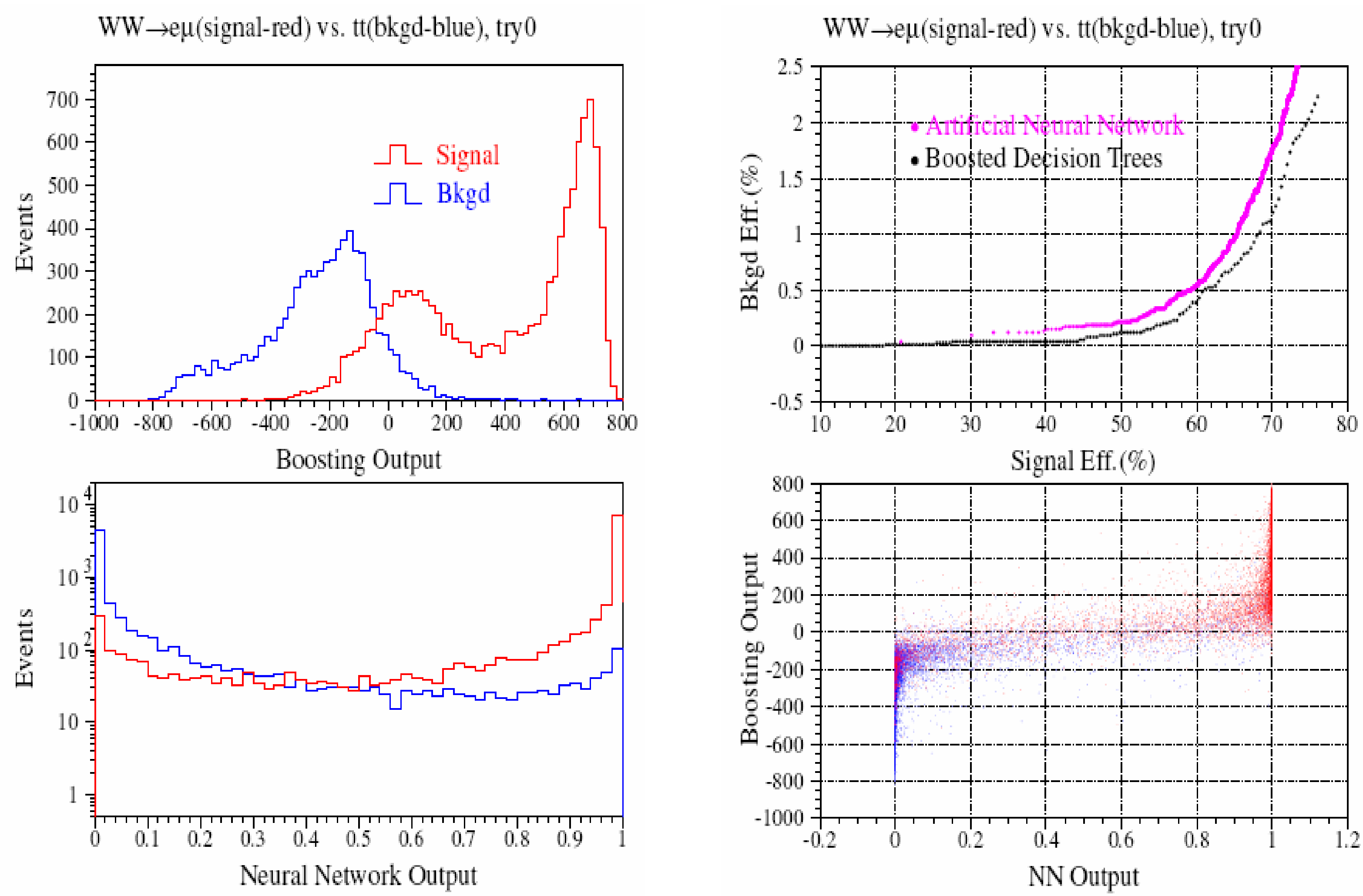
➔ Four most powerful variables are selected based on their Gini contributions.



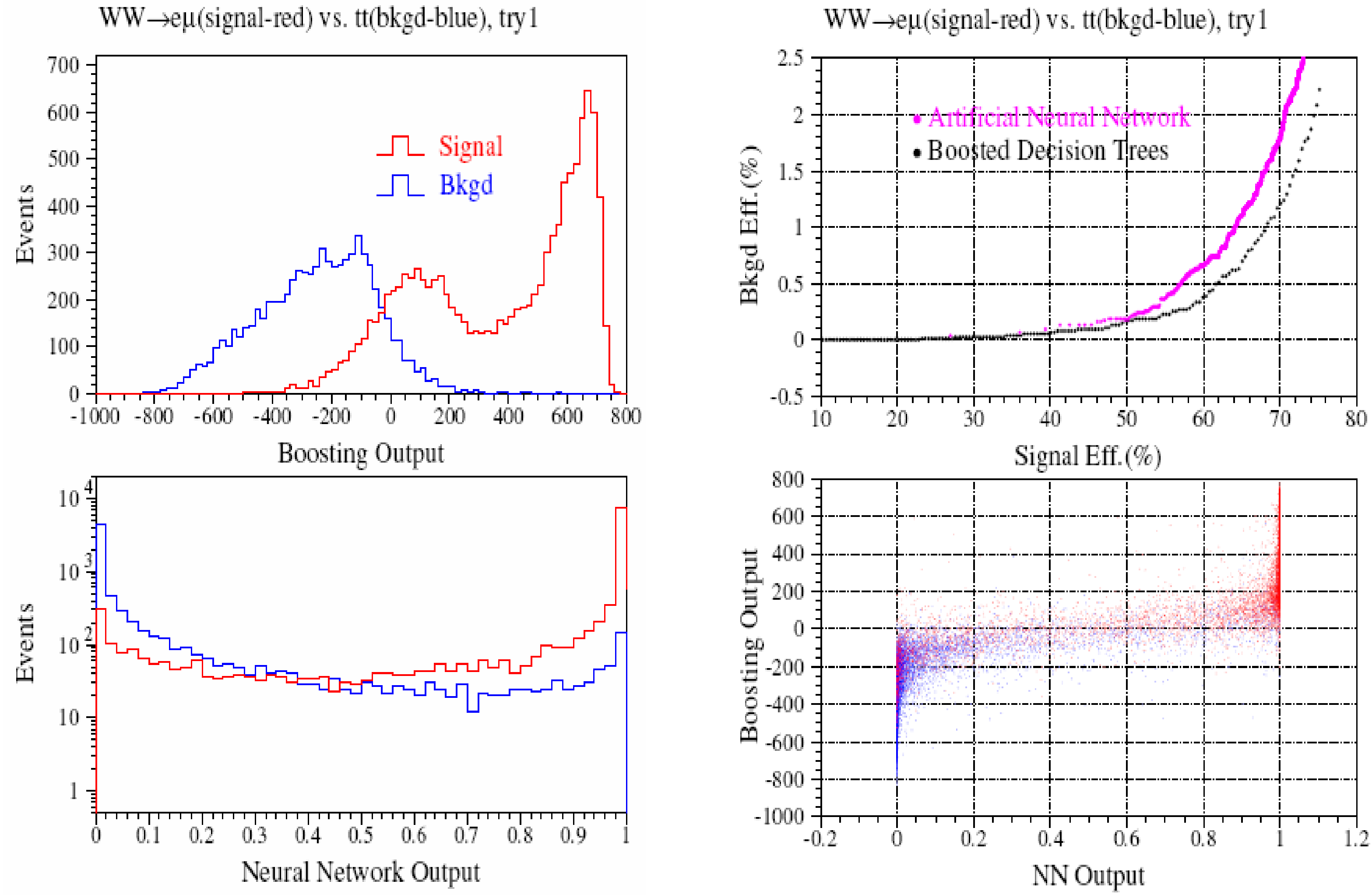
Some Weak Variables



Testing Results – 1(Sequentially)



Testing Results – 2 (Randomly)



Testing Results

- To train/test ANN/BDT with 11 different sets of MC events by selecting events randomly.
- To calculate average/RMS of 11 testing results

➔For given signal efficiency (50%-70%), ANN keeps more background events than BDT.

Signal Eff	Effbg_ANN Nbg_ANN	Effbg_BDT Nbg_BDT	Effbg_ANN/Effbg_BDT
50%	(0.267+-0.043)% 20	(0.138+-0.033)% 10	1.93
60%	(0.689+-0.094)% 51	(0.380+-0.041)% 28	1.81
70%	(1.782+-0.09)% 132	(1.22+-0.073)% 91	1.46

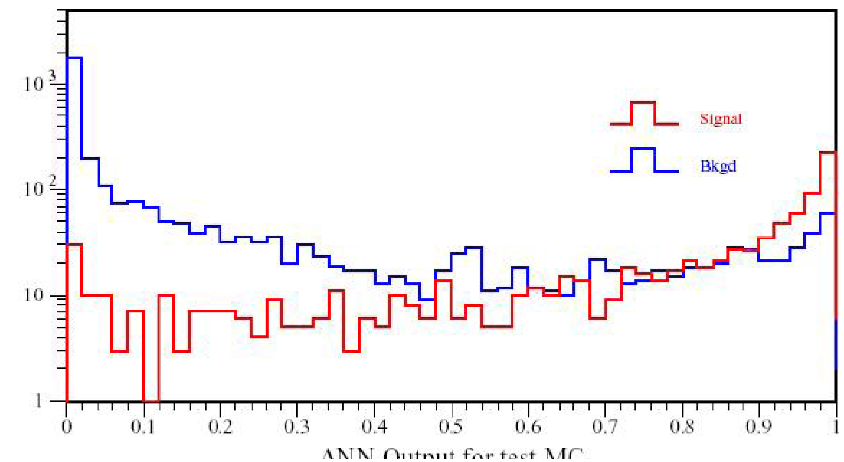
ZW/ZZ Leptonic Decays

- Signal Events - 3436:
  - ZW → eee, eeμ, eμμ, μμμ + X
- Background Events – 9279
  - ZZ → eee, eeμ, eμμ, μμμ + X
- Training Events – selected randomly
  - 2500 signal and 6000 background events
- Testing Events – the rest events for test
  - 936 signal and 3279 background events

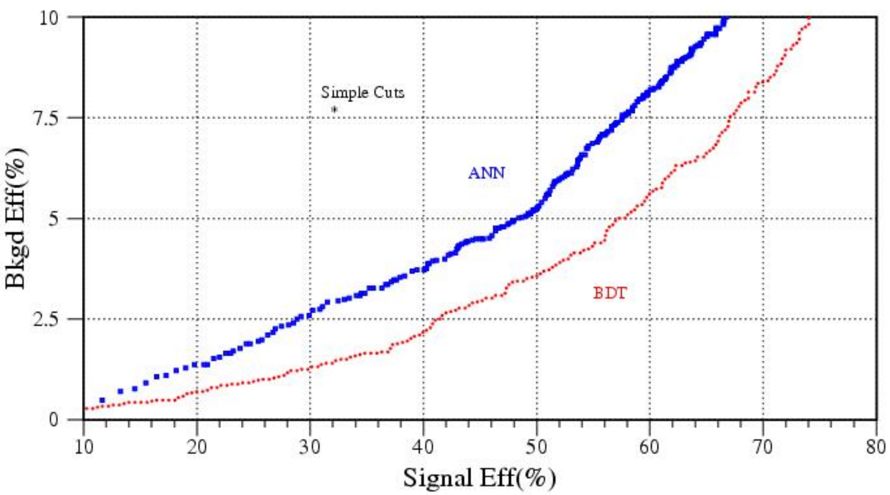
Testing Results

For fixed Eff_bkgd = 7.5 %
Signal Efficiencies are
- 32% -- Simple cuts
- 57% -- ANN
- 67% -- BDT

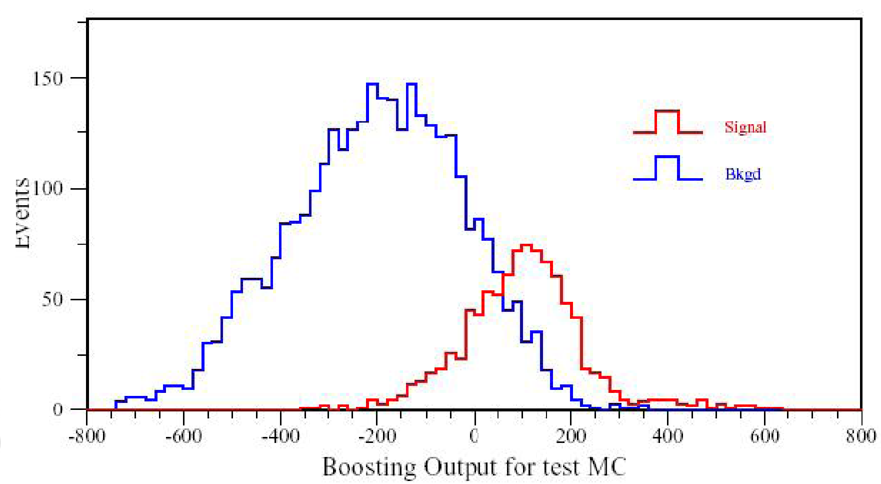
Artificial Neural Networks (ANN)



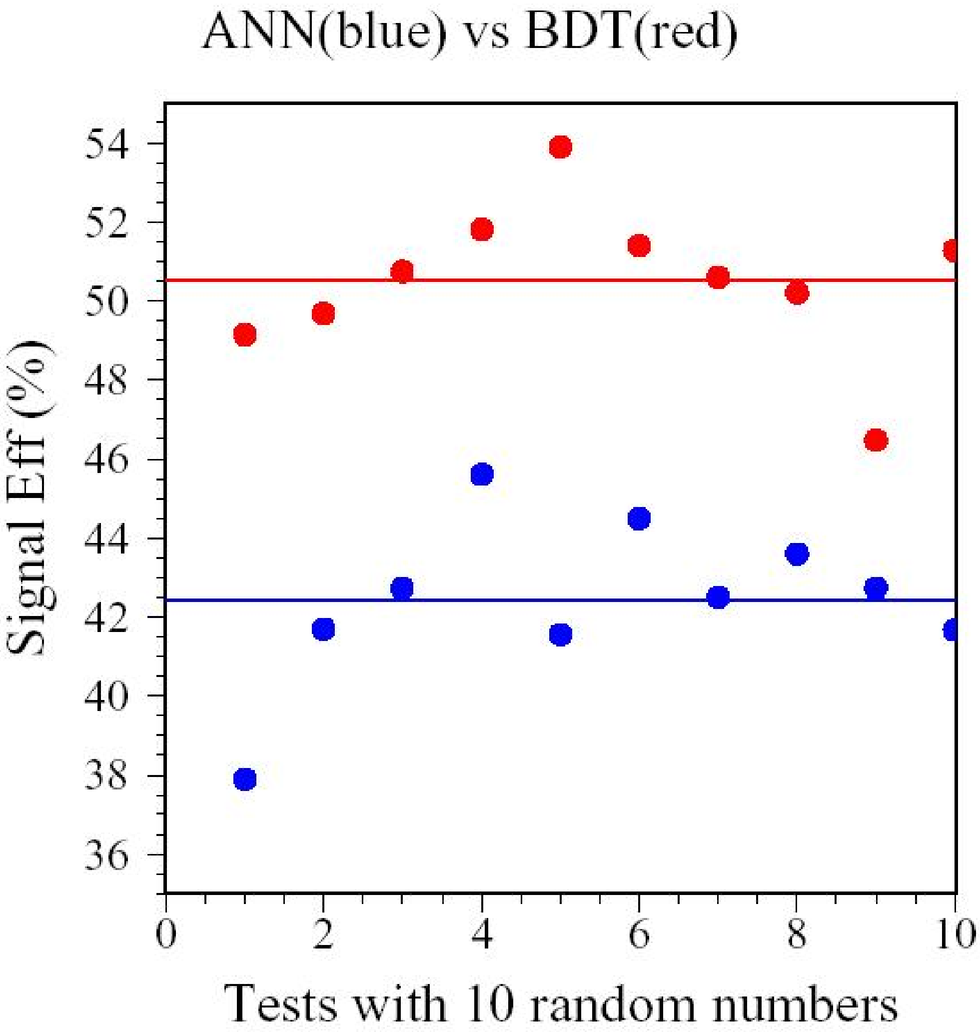
Boosted Decision Trees vs Artificial Neural Networks



Boosted Decision Trees (BDT)



ANN vs BDT - Performance



Training events are selected randomly. The rest events are used for test. The signal eff of ANN and BDT for 10 different random numbers are shown in the left plot.

For 3.5% background eff, the Signal eff are
42.45%+/-2.06%(RMS) for ANN
50.52%+/-1.93%(RMS) for BDT

ANN vs BDT - Stability

- Smear all input variables for all events in the testing samples.
- Var(i) = Var(i)*(1+0.05*normal Gaussian random number)
- For 3.5% bkgd eff, the signal eff are
  - Eff_ANN = 40.03% +/- 1.71%(RMS)
  - Eff_BDT = 50.27% +/- 2.20%(RMS)
- The degradation of signal eff using smeared test samples are
  - -2.43% +/- 2.68% for ANN
  - -0.25% +/- 2.93% for BDT

→ BDT is more stable than ANN for smeared test samples.

More Applications of BDT

- More and more major HEP experiments begin to use BDT (Boosting Algorithms) as an important analysis tool.
  - ATLAS Di-Boson analysis
  - ATLAS SUSY analysis – hep-ph/0605106 (JHEP060740)
  - BaBar data analysis – hep-ex/0607112, physics/0507143, 0507157
  - D0/CDF data analysis – hep-ph/0606257, Fermilab-thesis-2006-15
  - MiniBooNE data analysis – physics/0508045 (NIM A555, p370), physics/0408124 (NIM A543, p577), physics/0610276
  - ……
- Free softwares for BDT
  - http://gallatin.physics.lsa.umich.edu/~hyang/boosting.tar.gz
  - http://gallatin.physics.lsa.umich.edu/~roe/boostc.tar.gz, boostf.tar.gz
  - TMVA toolkit, CERN Root-integrated environment
    http://root.cern.ch/root/html/src/TMVA__MethodBDT.cxx.html
    http://tmva.sourceforge.net/

Conclusions and Outlook

➔BDT is more powerful and stable than ANN.

➔BDT is anticipated to have wide application in HEP data analysis to improve physics potential.

➔UM group plan to apply ANN/BDT to ATLAS SM physics analysis and searching for Higgs and SUSY particles.