

# Optimization of tau identification in ATLAS using multivariate tools

On behalf of the ATLAS Collaboration

**Marcin Wolter**

*Institute of Nuclear Physics PAN, Kraków*

***ACAT 2008, Erice, 3-7 November 2008***

# Physics processes with tau leptons

## Standard Model :

$Z \rightarrow \tau\tau$   
 $W \rightarrow \tau\nu_\tau$

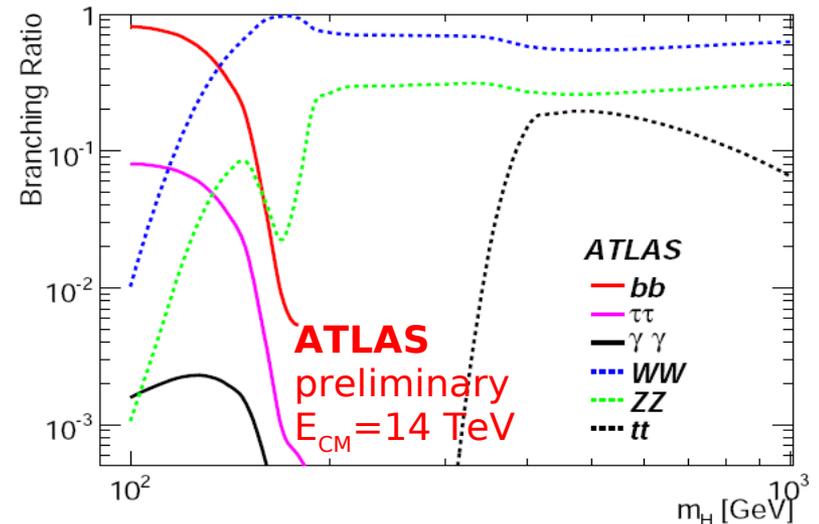
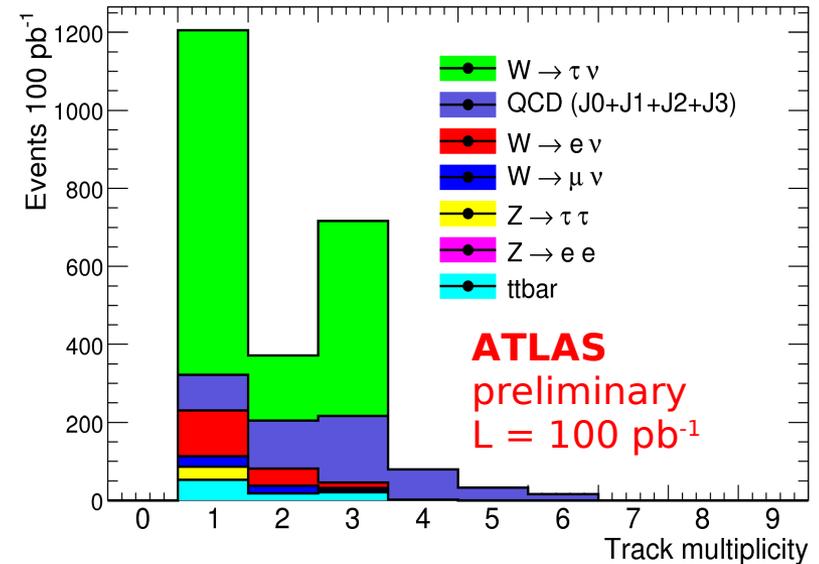
} important for first physics data analysis

## Higgs Processes :

- **Standard Model Higgs (VBF,ttH)**  
 $qqH \rightarrow qq\tau\tau$ ,  $ttH \rightarrow tt\tau\tau$
- **MSSM Higgs (h/A/H, H<sup>+</sup>)**  
 $h/A/H \rightarrow \tau\tau$ ,  $H^+ \rightarrow \tau\nu$

## Exotic Processes :

- **SUSY** signature with  $\tau$ 's in final state
- **Extra dimensions** ... new theories (?)



# Tau decays

Hadronic tau decays: mostly 1-prong or 3-prong decays (1 or 3 charged tracks):

Decay modes	Branching ratio
leptonic	35.2%
1-prong ( $\tau \rightarrow n\pi^0\pi^\pm\nu$ )	49.5%
3-prong ( $\tau \rightarrow n\pi^0\pi^\pm\pi^\mp\pi^\pm\nu$ )	14.6%

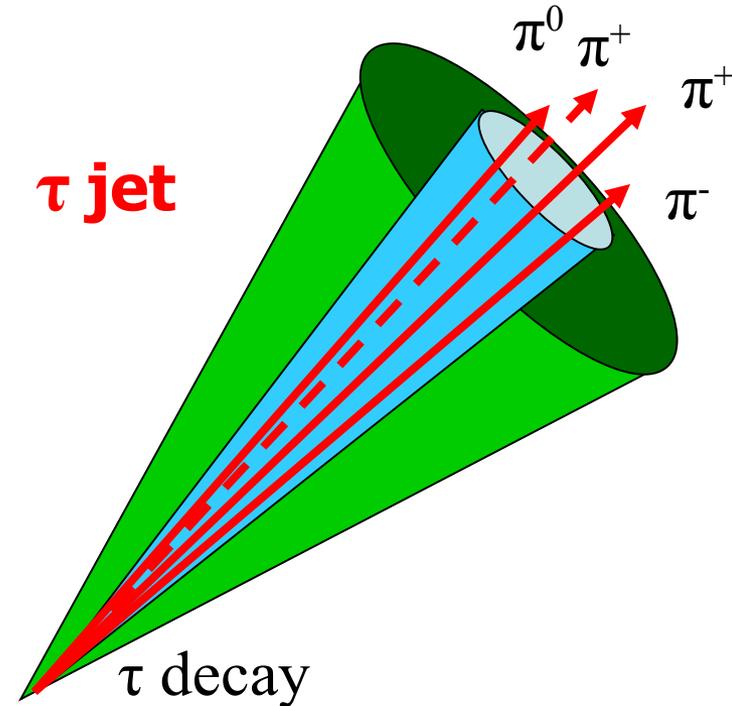
## Observed tau jets

### • Characteristics

- well-collimated calorimeter cluster
- Small number of associated charged tracks
- displaced secondary vertex

### • Main background

**QCD** jets – wider, higher track multiplicity, different characteristics of the track system and shapes of the calorimetric showers



# Tau reconstruction algorithm

## Track-seed and calo-seed:

- good quality tracks ( $p_T > 6$  GeV) as initial seed
- candidates with 1-8 quality tracks ( $p_T > 1$  GeV) in  $\Delta R < 0.2$  from the seed
- the  $\eta, \phi$  using  $p_T$  weighting of tracks, check charge consistency ( $|Q| \leq 2$ )
- matching cone 0.4 TopoJets ( $> 10$  GeV,  $\Delta R < 0.2$ ) as calo-seed
- $E_T$  (calorimetric) using H1-style calibration on cells from calo-seed
- $E_T^{\text{eflow}}$  (tracking) with energy-flow method (EM calo - separating neutral/charged sources of energy)
- Reconstruction of  $\pi^0$  subclusters

## Calo seed only:

- cone 0.4 TopoJets ( $> 10$  GeV) as calo-seed
- $\eta, \phi$  defined using calo-seed ( $\eta$  corrected for z vertex)
- looser track-quality selection, track  $p_T > 1$  GeV
- $E_T$  (calorimetric) using H1-style calibration on cells from calo-seed

## Track-seed only:

- small fraction of all candidates (few percent)

## **Both-seeds candidates (track and calo):**

signal reconstruction efficiency - 64% (1-prong) and 60% (3-prong)  
fakes from QCD background events - 4% (1-prong) and 8% (3-prong)

***Significant background suppression on the reconstruction level.***

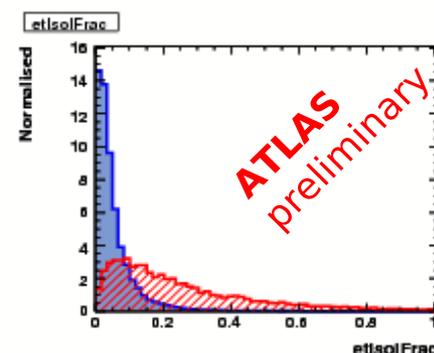
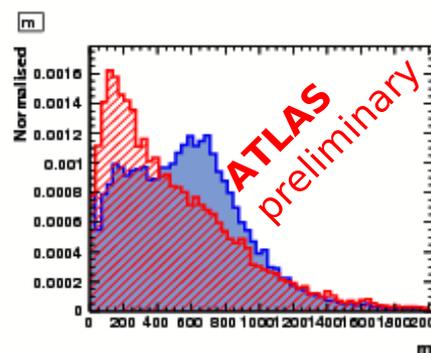
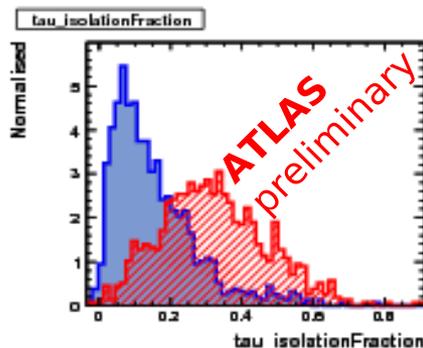
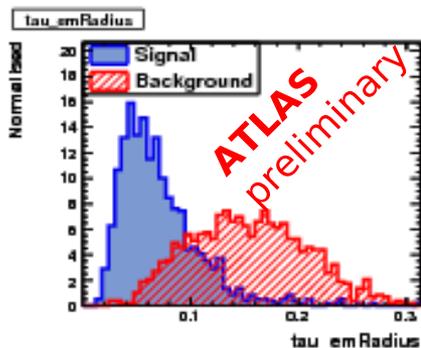
# Discriminating variables

## Calorimeter-seeded

- **8 identification variables.**

Example variables:

- 1) **emRadius**: radius of the cluster in the EM calorimeter,
- 2) **isolationFraction**: isolation fraction of transverse energy between  $0.1 < \Delta R < 0.2$  around the cluster barycenter



None of the variables alone gives sufficient signal/background separation:

**smart and efficient (multivariate) identification method needed.**

## Track-seeded

- 3 Prong – **11 id. variables**
- 1 Prong -- **9 id. variables**

Example variables:

- 1) **m**: invariant mass,
- 2) **etisolFrac**: ratio of transverse energy in  $0.2 < \Delta R < 0.4$  to total transverse energy

Analysis performed on ATLAS MC events:  
*Signal: W- $\rightarrow$  $\tau\nu$ (had)*  
*Background: QCD jet events*

# Tau identification algorithms

- **Cut analysis** - baseline method for track-seeded candidates (fast, robust, transparent)
- **Projected Likelihood** – implemented for both track and calo-seeded candidates
- **PDE-RS** – Probability Density Estimator with Range Searches (track-seeded candidates)
- **Neural Network** – implemented for track-seeded candidates,
- **BDT** - Boosted Decision Tree algorithm, relatively new in HEP (track-seeded and calo-seeded candidates)

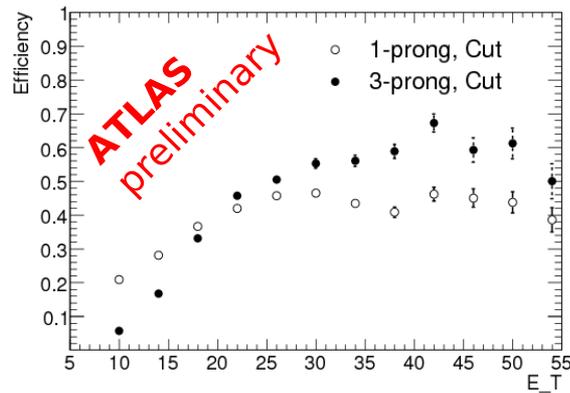
**All algorithms implemented in the TauDiscriminant package, which is a part of the ATLAS reconstruction software.**

# Cut analysis

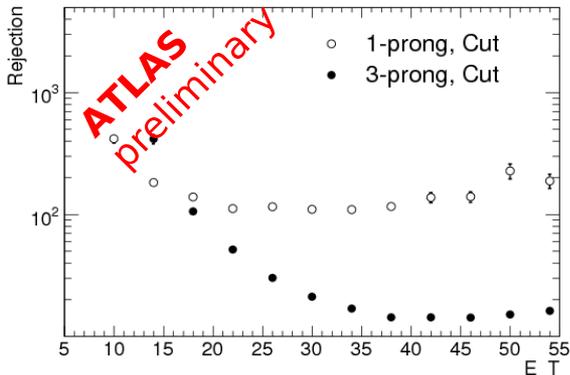
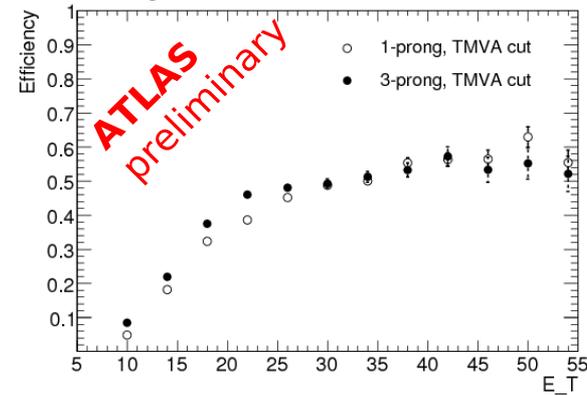
- Widely used because it is transparent and robust
- Cut analysis applied to track-seeded candidates

- **Human-optimized**
- baseline cut method

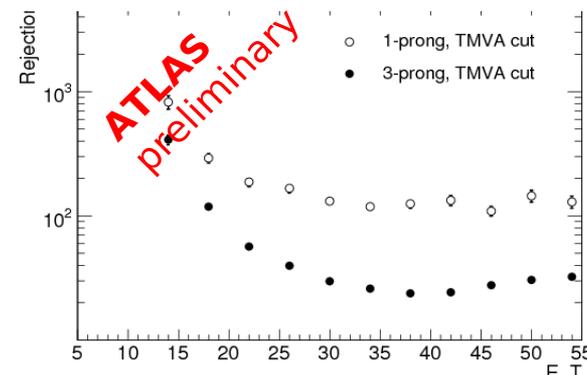
- **Automatically optimized** - *TMVA* package
- Decorrelation → scan in signal efficiency [0→1] and maximize background rejection
- random sampling: robust but slow



**Signal  
efficiency**



**Background  
rejection**



# Projected Likelihood Estimator

- Combine probability for an event to be signal / background from individual variables to

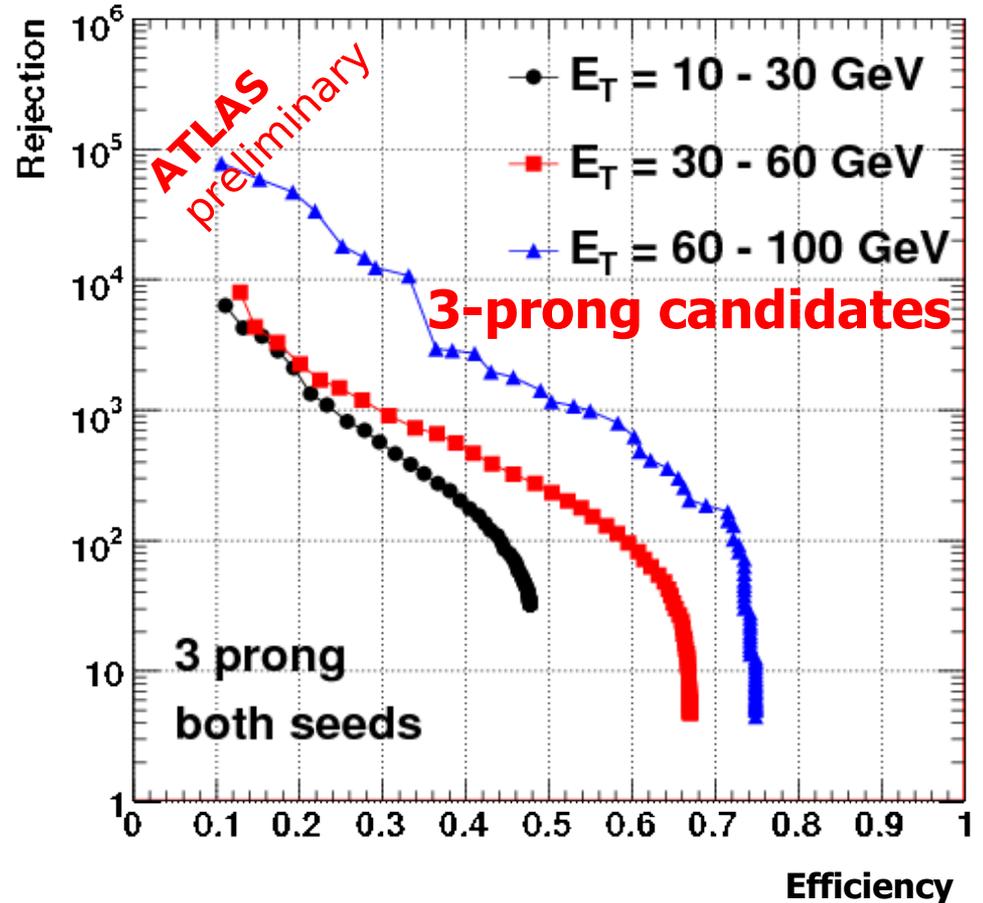
The diagram illustrates the relationship between probability and the likelihood estimator. A box labeled "Probability" has an arrow pointing to the left-hand side of the equation  $P_{signal}(event)$ . A box labeled "PDFs" has an arrow pointing to the product terms in the numerator and denominator of the fraction.

$$P_{signal}(event) = \frac{\prod P_{sig}(x_i)}{\prod P_{sig}(x_i) + \prod P_{bkg}(x_i)}$$

- Method assumes uncorrelated input variables:**
  - In that case it is the optimal MVA approach, since it contains *all* the information
  - usually it is *not true* → inferior to other more sophisticated MV methods (BDT, Neural Network, etc...)
  - **Commonly used in High Energy Physics**

# Projected Likelihood Estimator

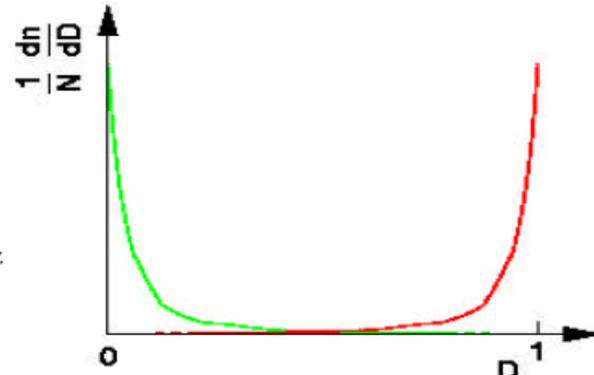
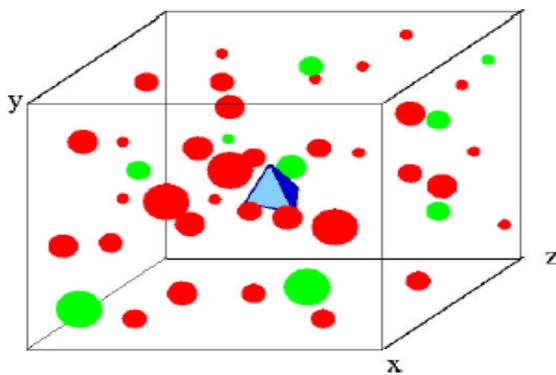
- **Applied to both track-seeded and calo-seeded candidates**
- Base-line method for calo-seeded candidates
- Good performance, low memory and CPU consumption
- Transparent and insensitive to small changes in the training sample.



# PDE\_RS method

Parzen estimation (1960s) – approximation of the unknown probability as a **sum of kernel functions** placed at the points  $x_n$  of the training sample.

- Make it faster - count signal ( $n_s$ ) and background ( $n_b$ ) events in N-dim hypercube around the event classified – only few events from the training sample needed (PDE\_RS).
- Hypercube dimensions are free parameters to be tuned.
- Discriminator  $D(x)$  given by signal and background event densities:



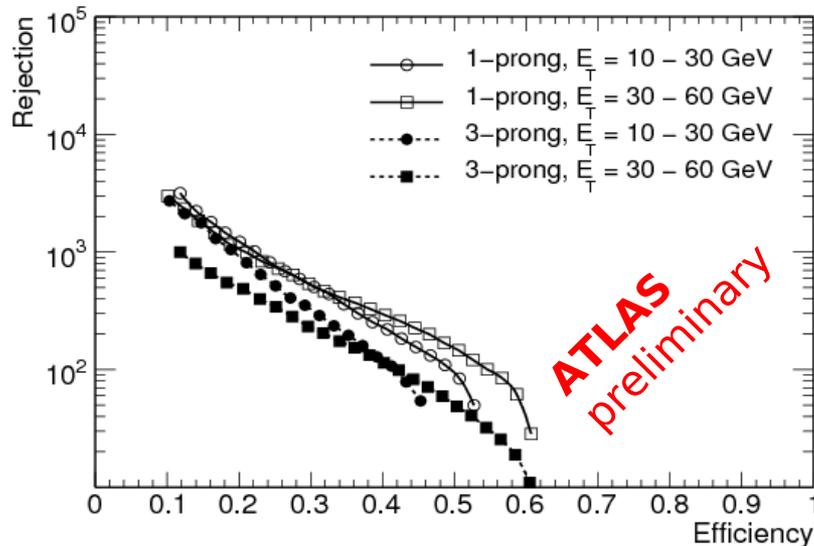
$$D(x) = \frac{n_s}{n_s + n_b}$$

- Events stored in the binary tree – easy and fast finding of neighbor events.
- **Adaptive version** – size of the hypercube adapts to the event densities, a kernel function instead of sharp hypercube boundaries.

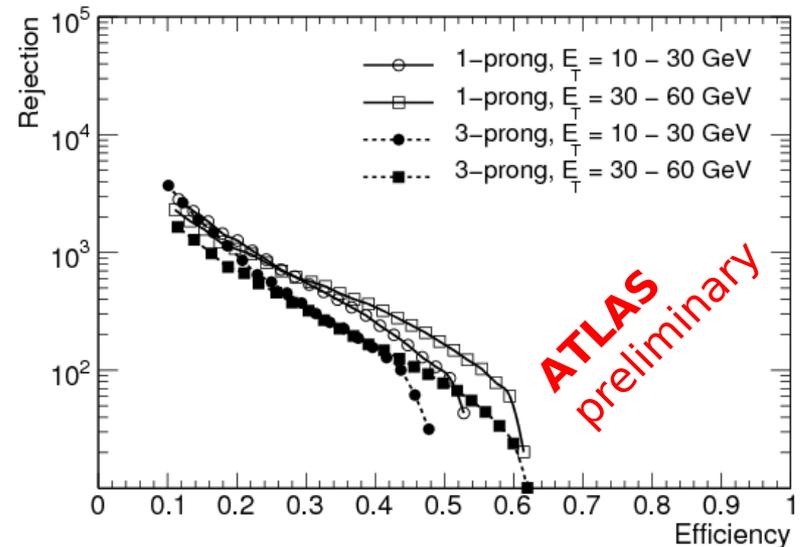
# PDE\_RS method

Applied to track-seeded candidates

## Standard



## Adaptive

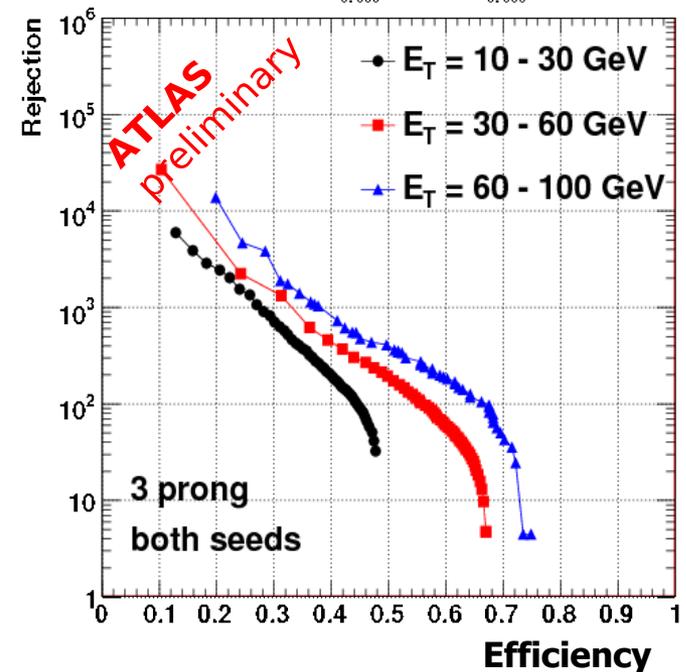
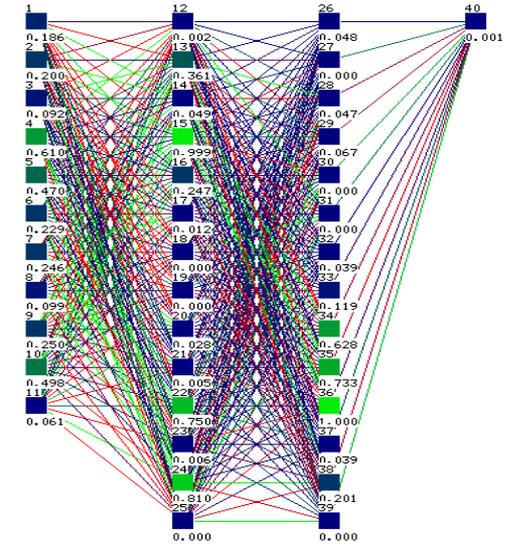


- **Advantages:** Good performance, takes correlations into account
- **Disadvantages:** High memory and CPU consumption, quite big training data samples needed

Use of adaptive PDERS increases background rejection by about 5% compared to standard PDE-RS.

# Neural Network

- Uses track-based variables
- Impact parameter significance added (1-prong candidates)
- **8 neural networks:** 1,2,3-prong candidates, with/without impact parameter (1-prong candidates), with/without  $\pi^0$  cluster.
- Good performance (uses correlations between variables, impact parameter information added).
- Stuttgart Neural Network Simulator (SNNS) used, feed-forward NN with two hidden layers. Trained network is converted to the C code – very fast, low memory consumption.



# Decision Trees

- **Sequential application of “cuts”, the final nodes have an associated classifier value based on their purity**
- **Training: growing a decision tree:**

- Start with Root node
- Split training sample according to cut on best variable at this node
- Splitting criterion: e.g., maximum

“Gini-index”  $G = P \times (1 - P)$

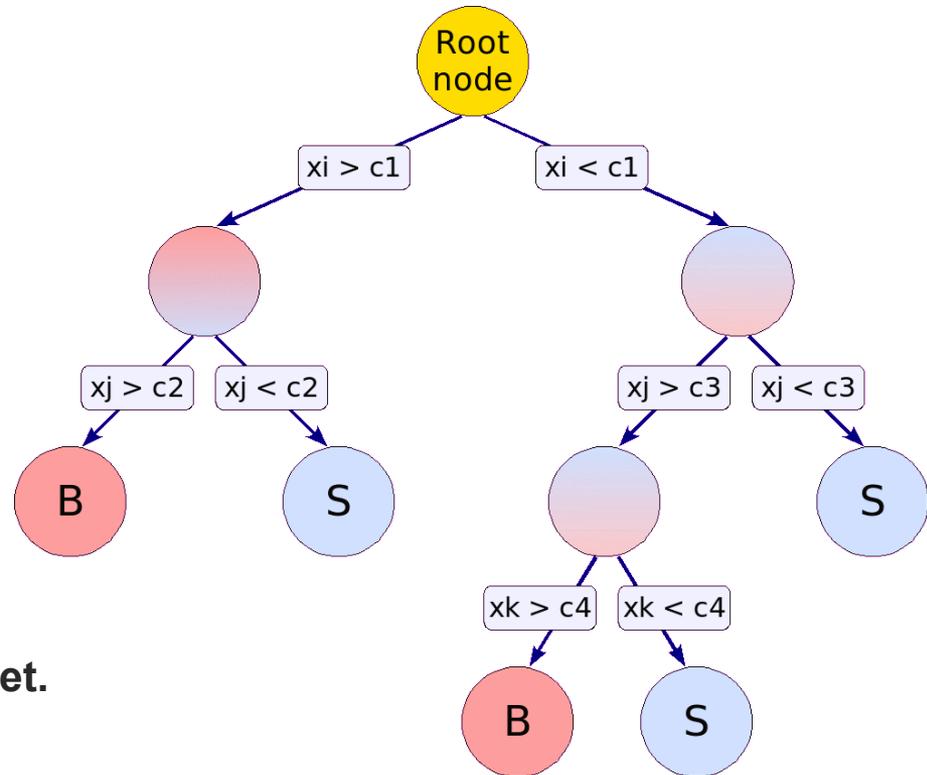
Purity  $P = \sum W_s / (\sum W_s + \sum W_B)$

- Continue splitting until min. number of events or max. purity reached

- **Advantages:**

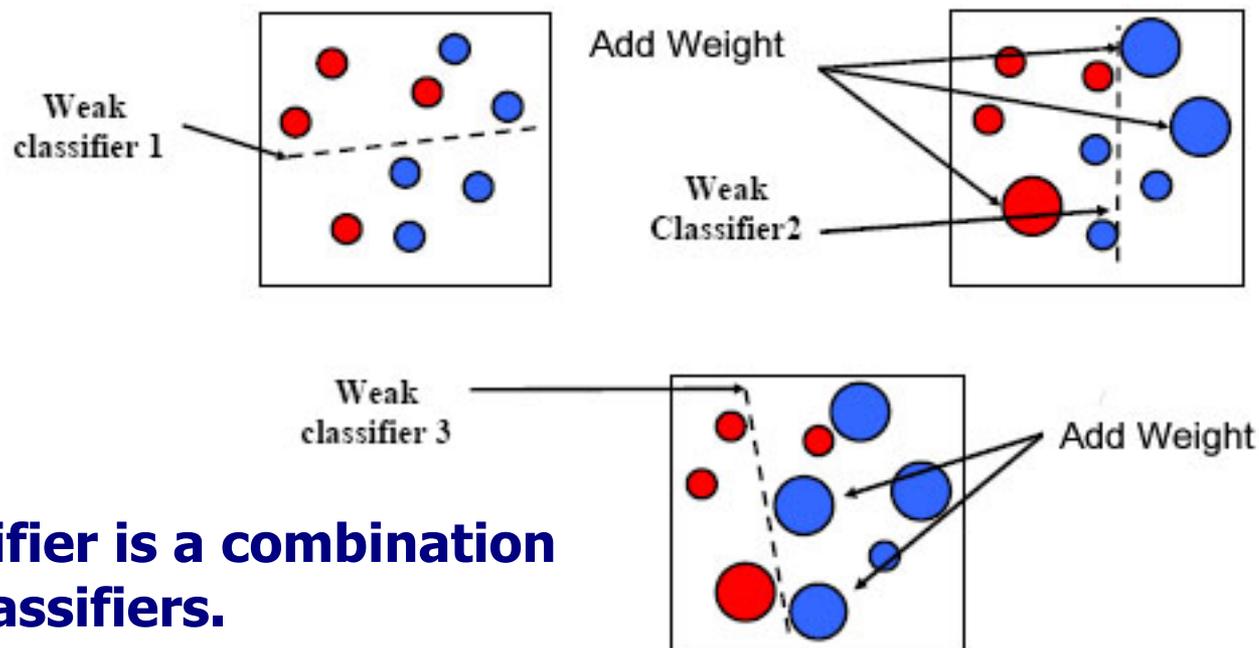
- fast and easy to train, easy to interpret.
- useless/weak variables are ignored

- **Disadvantages:** small changes in training sample can give large changes in tree structure



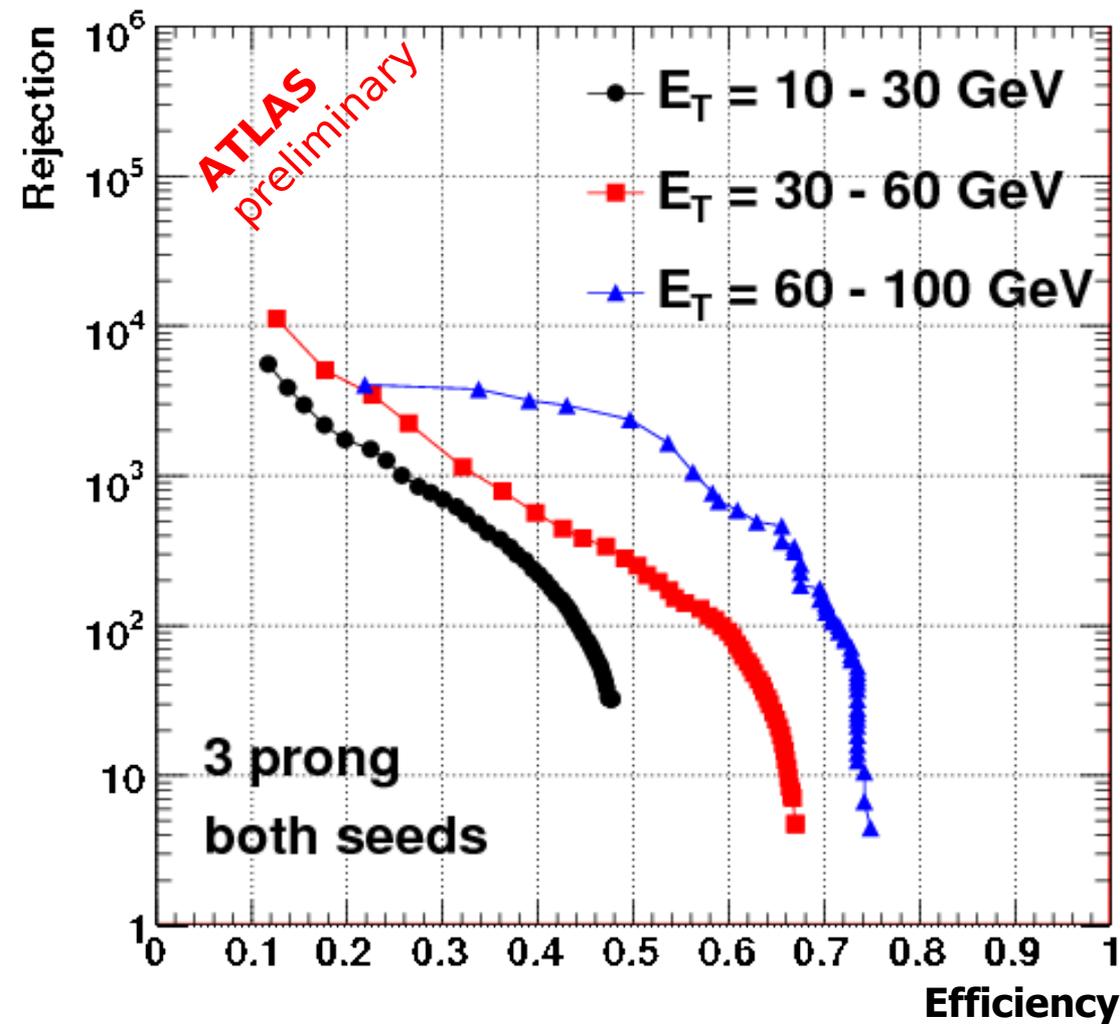
# Boosted Decision Trees

- **Boosting:** a general technique for improving the performance of any weak classifier.
- **Boosted Decision Trees (1996, R.Shapire & Y.Freund):** combining several decision trees (forest) derived from one training sample via the application of event weights into ONE multivariate event classifier by performing “majority vote”
  - ➔ **AdaBoost:** wrongly classified training events are given a larger weight.

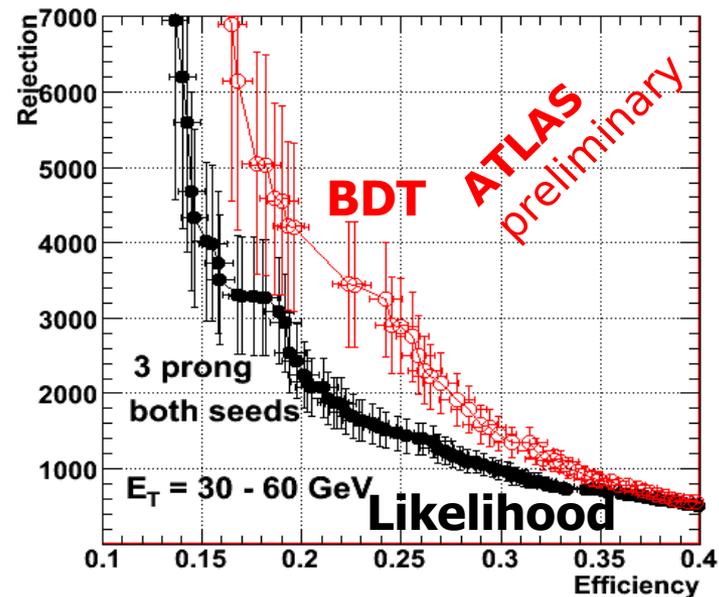


**Final classifier is a combination of weak classifiers.**

# Boosted Decision Tree



- D0 experiment BDT software used for training
- 10 trees for boosting (AdaBoost)
- Similar set of variables as used by Likelihood (few additional)
- Use of correlations between variables – BDT has a potential to improve the performance reached by Likelihood



# Summary

- Tau identification significantly improved by using multivariate analysis tools.
- All of the presented classification methods are performing well:
  - **Cuts** – fast, robust, transparent for users.
  - **Projected Likelihood** – a popular and well performing tool
  - **PDE\_RS** – robust and efficient, but large samples of reference candidates needed.
  - **Neural network** – fast classification while converted to the C function after training,
  - **BDT** - fast and simple training, insensitive to outliers, good performance.  
Relatively new in HEP
- Multivariate analysis is necessary, if it is important to extract as much information from the data as possible.
- For classification problems no single “best” method exists. What matters - is also simplicity and speed of learning and fast (and robust) classification.

# Future plans

## We should prepare for real data

- **With Monte Carlo:**
  - Variable ranking and reduction – optimal set of variables
  - Focus on flexibility and robustness
  - PARADIGM approach should help...
- **With real data:**
  - Comparison of Monte Carlo with data (validation)
  - Choice of trusted variables
  - Systematics, systematics...

# Backup slides

Algorithm	Tracks	eff [%]	rej	eff [%]	rej
Tau1p3pDiscriCut	1 prong	38	148 ± 5	49	103 ±4
	3 prong	34	111 ± 3	57	22 ±0.4
Tau1p3pDiscriCutTMVA	1 prong	31	322 ±18	58	108 ±4
	3 prong	37	127 ±4	51	31 ±0.5
Tau1p3pDiscriLL	1 prong	30	455 ±31	30	806 ±94
	3 prong	30	319 ±18	31	310 ±22
Tau1p3pDiscriPDRS	1 prong	30	532 ±39	30	733 ±81
	3 prong	30	363 ±22	30	444 ±38
Tau1p3pDiscriNN	1 prong	30	724 ±63	30	1050 ±140
	3 prong	30	712 ± 61	30	1386 ±212
1Tau3pEfficNN	1 prong	30	306 ±17	30	735 ±82
	3 prong	30	355 ±21	30	943 ±119
TauCommonLikelihood	1 prong	30	450 ±30	30	899 ±111
	3 prong	30	547 ±41	30	984 ±127
TauLikelihood2007	1 prong	30	272 ±14	30	967 ±124
	3 prong	30	233 ±11	30	215 ±13
TauBDTAnalysis	1 prong	30	528 ±39	30	1046 ±139
	3 prong	30	707 ± 60	30	1465 ±230
TauBDT	1 prong	30	188 ±8	30	576 ±56
	3 prong	30	196 ±8	30	451 ±39

# Tau decay branching ratios

Table 1: The  $\tau$  decay branching ratios, based on  $10^8$  simulated  $\tau$  decays from  $Z \rightarrow \tau\tau$  events.

Decay modes	TAUOLA-CLEO
$\tau \rightarrow e\nu_e \nu_\tau$ ,	17.8 %
$\tau \rightarrow \mu\nu_\mu \nu_\tau$	17.4 %
$\tau \rightarrow h^\pm \text{ neutr. } \nu_\tau$ (single-prong)	49.5 %
$\tau \rightarrow \pi^\pm \nu_\tau$	11.1 %
$\tau \rightarrow \pi^0 \pi^\pm \nu_\tau$	25.4 %
$\tau \rightarrow \pi^0 \pi^0 \pi^\pm \nu_\tau$	9.2 %
$\tau \rightarrow \pi^0 \pi^0 \pi^0 \pi^\pm \nu_\tau$	1.1 %
$\tau \rightarrow K^\pm \text{ neutr. } \nu_\tau$	1.6 %
$\tau \rightarrow h^\pm h^\pm h^\pm \text{ neutr. } \nu_\tau$ (three-prong)	14.6 %
$\tau \rightarrow \pi^\pm \pi^\pm \pi^\pm \nu_\tau$	9.0 %
$\tau \rightarrow \pi^0 \pi^\pm \pi^\pm \pi^\pm \nu_\tau$	4.3 %
$\tau \rightarrow \pi^0 \pi^0 \pi^\pm \pi^\pm \pi^\pm \nu_\tau$	0.5 %
$\tau \rightarrow \pi^0 \pi^0 \pi^0 \pi^\pm \pi^\pm \pi^\pm \nu_\tau$	0.1 %
$\tau \rightarrow K_S^0 X^\pm \nu_\tau$	0.9 %
$\tau \rightarrow (\pi^0) \pi^\pm \pi^\pm \pi^\pm \pi^\pm \pi^\pm \nu_\tau$ (five-prong)	0.1 %
other modes with K	1.3 %
others	0.03 %