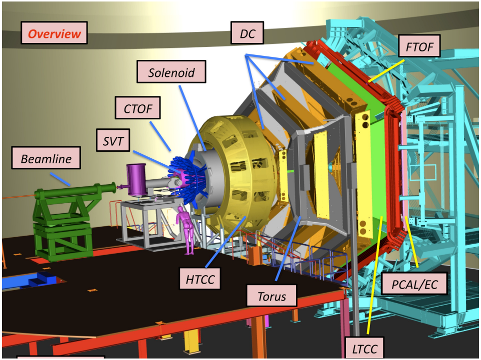
CLAS12 PID and Machine Learning

Daniel Lersch, Michael C. Kunkel

03.10.2017

Introduction

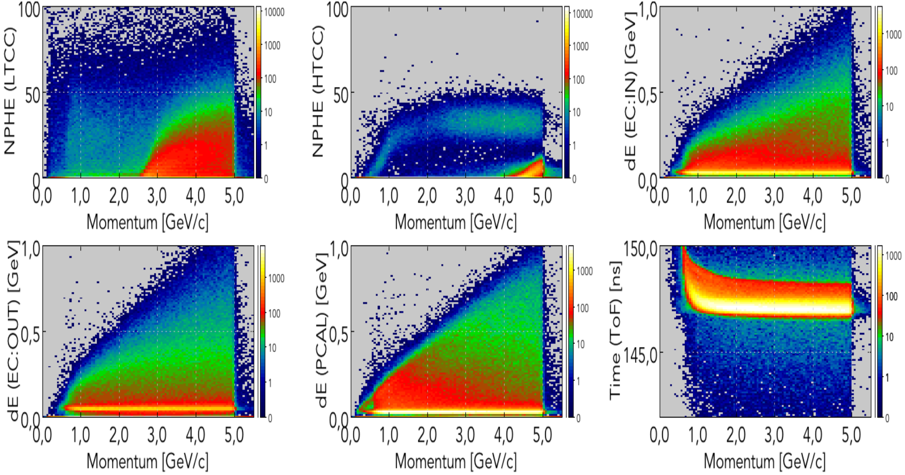


- **Wanted:** Particle masses $m_{e^\pm}$, $m_{\pi^\pm}$, m_p, $m_{K^\pm}$, …
- **Given:** Information from CLAS12 sub detector systems
- **Approach:** Combine detector information and feed them into a machine learning algorithm
  $\Rightarrow$ Get particle type
- All results shown here are based on:
  - Simulated single particle tracks
  - GEMC 4a.2.1
  - COATJAVA 4a.8.1
  - Apache-Spark 2.20 framework

$\Rightarrow$ **Todays focus: Separation of e^- / π^-**

i) Which variables (or combination of variables) to chose for PID?
ii) Which machine learning algorithm / classifier is "the best"?
iii) Quality of results $\Leftrightarrow$ systematic checks?

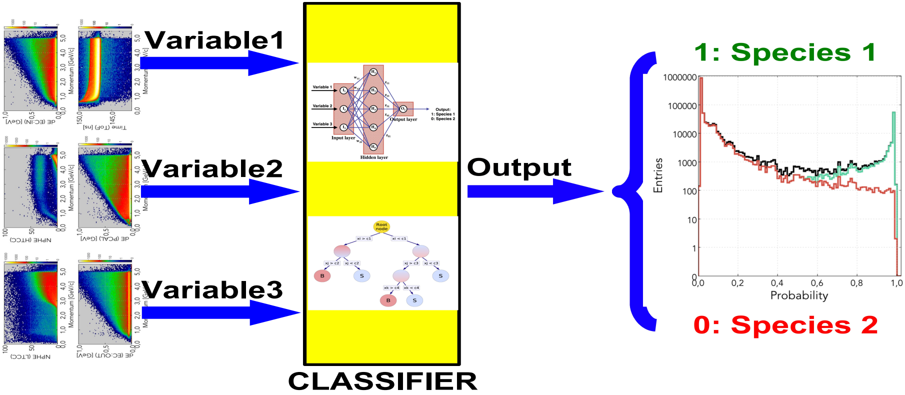
The $e^-\pi^-$ Data Set



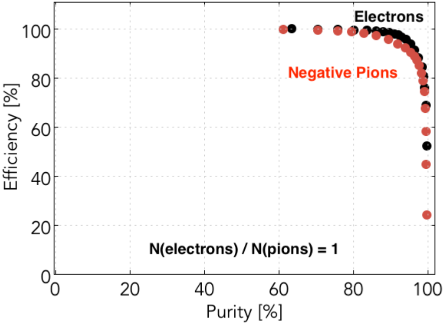
- Possible PID-variables:
  p, **nphe(LTCC)**, **nphe(HTCC)**, $\Delta E(pcal)$, $\Delta E(ec:in)$, $\Delta E(ec:out)$, $\Delta E(cal)$, **t**
- Ratio: $N(\pi^-)/N(e^-) = 10$
- Request particles to have: $p > 0$ and at least one other sub detector fired

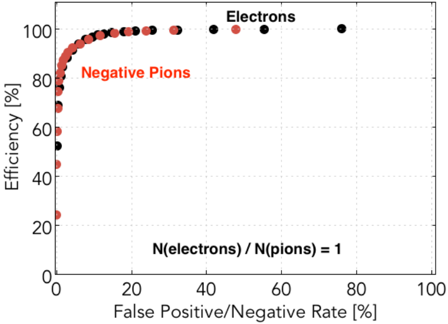
Solving multivariate Classification Problems

- Several classifier algorithms available
  (neural networks, boosted decision trees, support vector machines,...)
- Classifier internal parameters (weights, thresholds,...) are determined via training

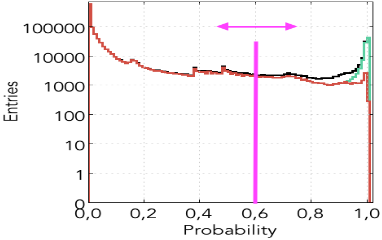


Evaluation of a trained Classifier: The ROC-Curve

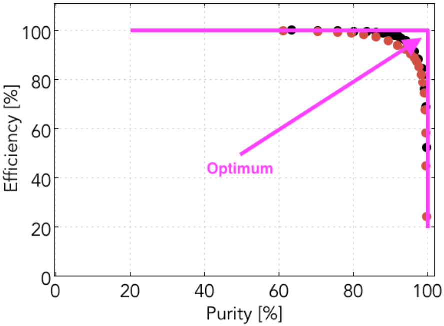


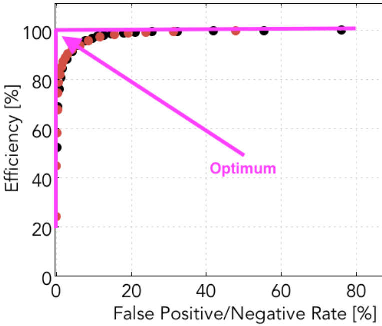


- <u>Receiver-Operating-Characteristics:</u>
  - **Efficiency**: $\epsilon_S = \frac{N_S^{acc}}{N_S^{all}}$, $\epsilon_B = \frac{N_B^{rej}}{N_B^{all}}$
  - **Purity**: $P_S = \frac{N_S^{acc}}{N_S^{acc}+N_B^{acc}}$, $P_B = \frac{N_B^{acc}}{N_S^{acc}+N_B^{acc}}$
  - **False Positive Rate**: $FPR = \frac{N_B^{acc}}{N_B^{all}}$
  - **False Negative Rate**: $FPR = \frac{N_S^{rej}}{N_S^{all}}$

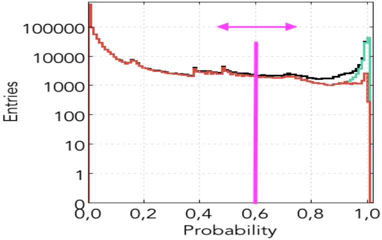


Evaluation of a trained Classifier: The ROC-Curve

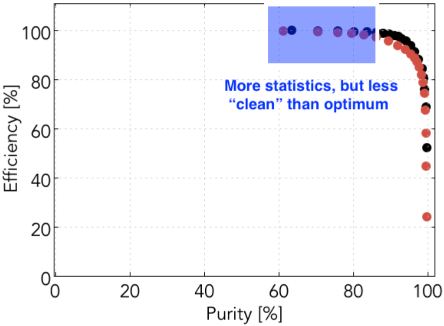


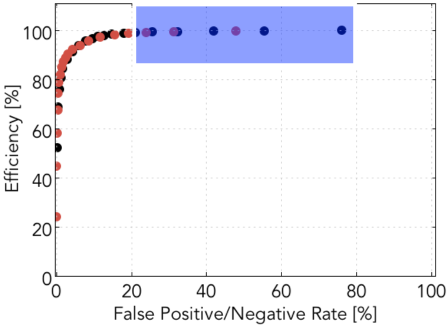


- Receiver-Operating-Characteristics:
  - **Efficiency**: $\epsilon_S = \frac{N_S^{acc}}{N_S^{all}}$, $\epsilon_B = \frac{N_B^{rej}}{N_B^{all}}$
  - **Purity**: $P_S = \frac{N_S^{acc}}{N_S^{acc}+N_B^{acc}}$, $P_B = \frac{N_B^{acc}}{N_S^{acc}+N_B^{acc}}$
  - **False Positive Rate**: $FPR = \frac{N_B^{acc}}{N_B^{all}}$
  - **False Negative Rate**: $FPR = \frac{N_S^{rej}}{N_S^{all}}$

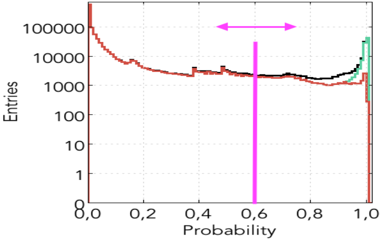


Evaluation of a trained Classifier: The ROC-Curve

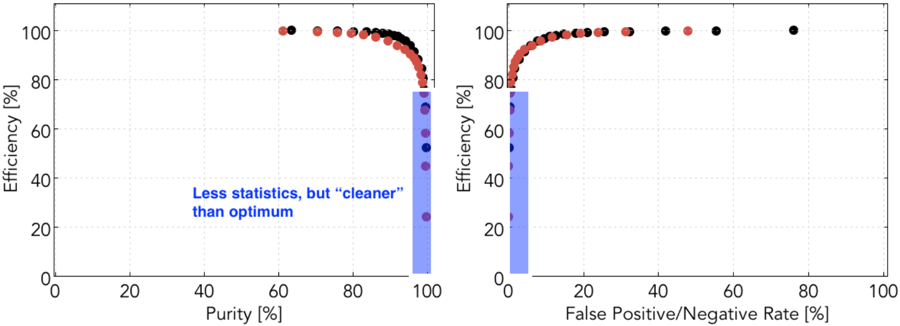




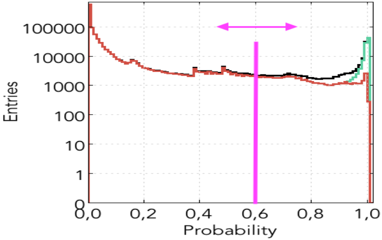
- Receiver-Operating-Characteristics:
  - **Efficiency**: $\epsilon_S = \frac{N_S^{acc}}{N_S^{all}}$, $\epsilon_B = \frac{N_B^{rej}}{N_B^{all}}$
  - **Purity**: $P_S = \frac{N_S^{acc}}{N_S^{acc}+N_B^{acc}}$, $P_B = \frac{N_B^{acc}}{N_S^{acc}+N_B^{acc}}$
  - **False Positive Rate**: $FPR = \frac{N_B^{acc}}{N_B^{all}}$
  - **False Negative Rate**: $FPR = \frac{N_S^{rej}}{N_S^{all}}$



Evaluation of a trained Classifier: The ROC-Curve

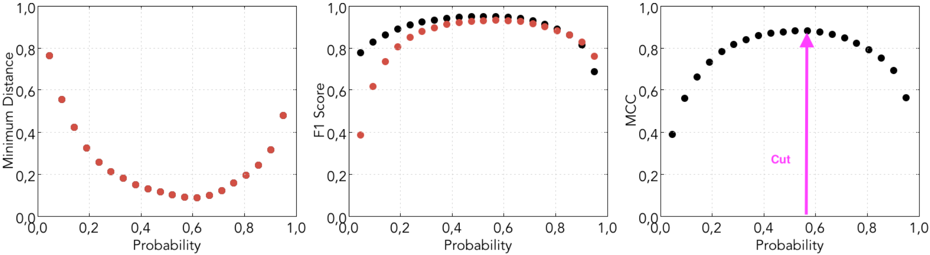


- <u>Receiver-Operating-Characteristics:</u>
  - **Efficiency**: $\epsilon_S = \frac{N_S^{acc}}{N_S^{all}}$, $\epsilon_B = \frac{N_B^{rej}}{N_B^{all}}$
  - **Purity**: $P_S = \frac{N_S^{acc}}{N_S^{acc}+N_B^{acc}}$, $P_B = \frac{N_B^{acc}}{N_S^{acc}+N_B^{acc}}$
  - **False Positive Rate**: $FPR = \frac{N_B^{acc}}{N_B^{all}}$
  - **False Negative Rate**: $FPR = \frac{N_S^{rej}}{N_S^{all}}$

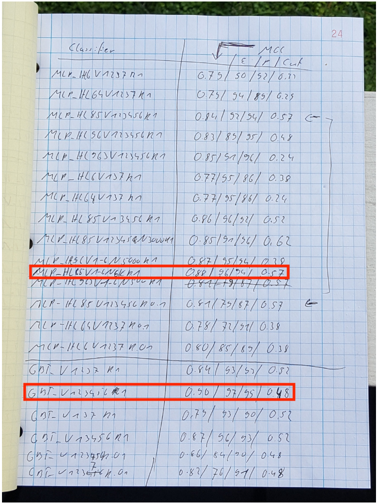


Choosing the Classification Parameters: ROC-Metrics

- Use ROC-curve to:
  i) Select probability cut
  ii) Choose classifier
- Apply metric:
  - Distance: $d \equiv \sqrt{(1-\epsilon_S)^2 + (0 - FPR)^2} \in [0, \infty)$ (left plot)
  - F1 score: $f_1 \equiv 2\frac{\epsilon_S \times P_S}{\epsilon_S + P_S} \in [0, 1]$ (centre plot)
  - Mathews Correlation Coefficient: $MCC \equiv \frac{\epsilon_S \times \epsilon_B - FPR \times FNR}{\sqrt{\frac{\epsilon_S}{P_S} \times \frac{\epsilon_B}{P_B}}} \in [-1, 1]$ (right plot)

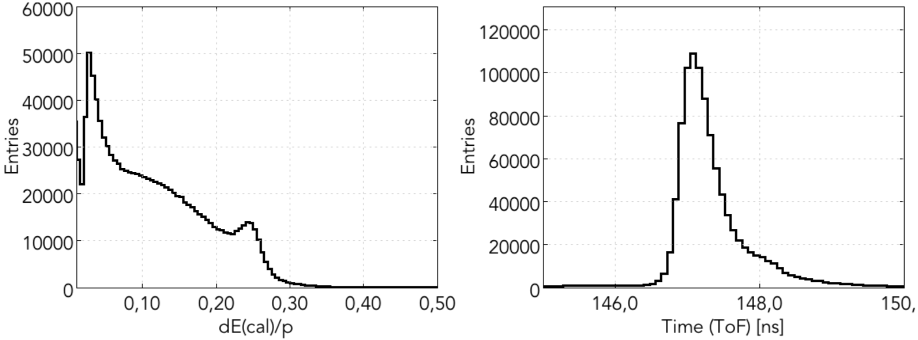


Choosing the Classification Parameters: Result



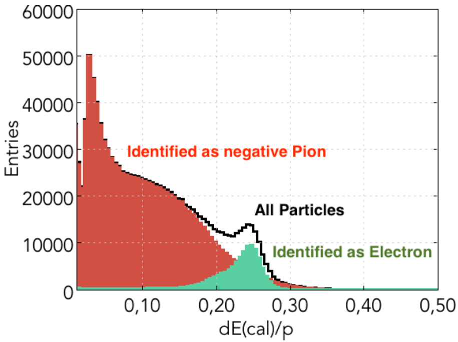
- Use MCC to choose a classifier, PID Variables and probability threshold:
  - Boosted Decision Tree with 100 Trees and probability threshold: 0.48
  - Neural Network with architecture $\{8:5\}$ and probability threshold: 0.57
  - Both classifier trained with:
    - p
    - nphe(LTCC)
    - nphe(HTCC)
    - $\Delta E(pcal)$
    - $\Delta E(ec:in)$
    - $\Delta E(ec:out)$
    - $N(\pi^-)/N(e^-) = 1$
- Room for improvement: Automated Parameter Scan as a function of MCC (Or any other metric)

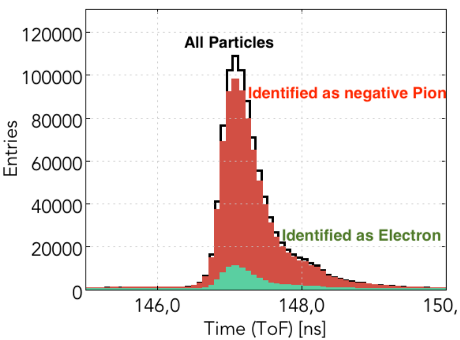
Application of the Classifier on the $e^-\pi^-$ Data Set



Classifier	Efficiency ϵ_S [%]	Purity P_S [%]	FPR [%]	$[N(\pi^-)/N(e^-)]_{rec}$
	—	—	—	—
	—	—	—	—

Application of the Classifier on the $e^-\pi^-$ Data Set



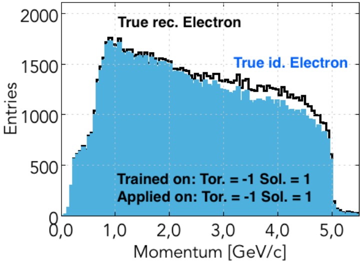


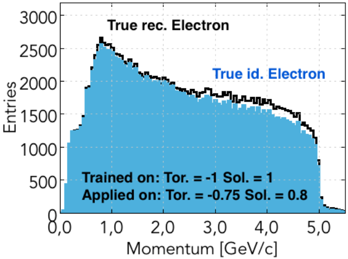
Classifier	Efficiency ϵ_S [%]	Purity P_S [%]	FPR [%]	$[N(\pi^-)/N(e^-)]_{rec}$
Decision Tree	96	58	7	6
Neural Network	94	61	6	6

Changing the Magnetic Field

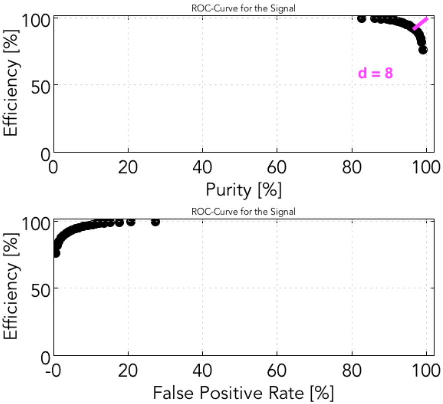
- Checked influence of different magnetic field settings on classifier performance
- Kept ratio: $N(\pi^-)/N(e^-) = 10$
- Three numbers in each cell of the table are: $\epsilon_S[\%]$ / $P_S[\%]$ / $FPR[\%]$

Applied Field $\Rightarrow$ **Trained Field** $\Downarrow$	$Tor. = -1$ $Sol. = 1$	$Tor. = -0.75$ $Sol. = 0.6$	$Tor. = -0.75$ $Sol. = 0.8$
$Tor. = -1$ $Sol. = 1$	94 / 61 / 6	95 / 62 / 6	95 / 61 / 6
$Tor. = -0.75$ $Sol. = 0.6$	94 / 53 / 8	95 / 57 / 8	95 / 55 / 8
$Tor. = -0.75$ $Sol. = 0.8$	95 / 56 / 6	95 / 59 / 6	95 / 57 / 7



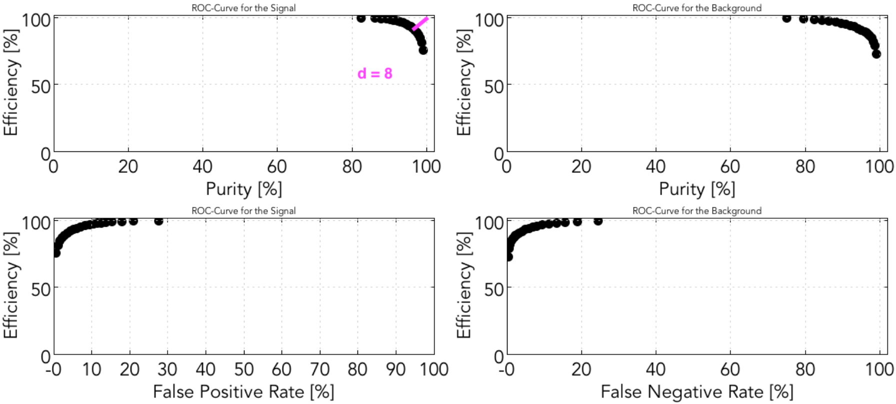


Resolution Effects for $N(\pi^-)/N(e^-) = 1$



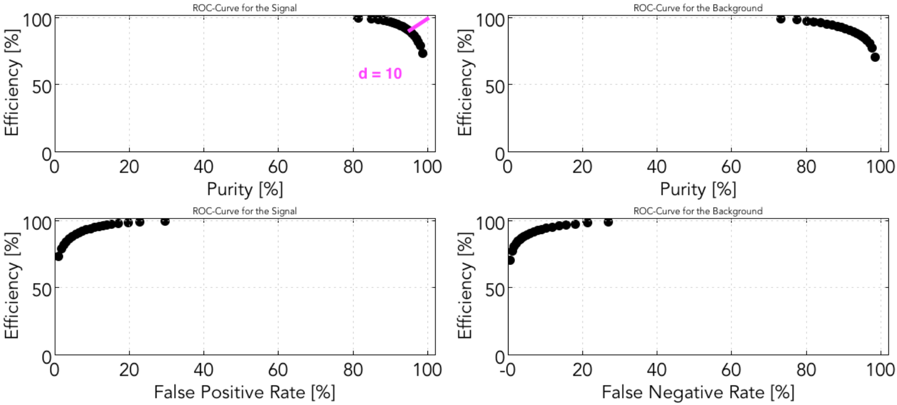
- **Basic question:** What happens if resolution in measured data is different from the resolution the classifier has been trained on?
- **Simple Test:** Apply trained classifier on a data set with different resolution than the training set
- Smeared all PID variables by a Gaussian distribution: $x \mapsto x + \text{Gauss}(0, \delta \cdot x)$
- Shown above: ROC-curves for $\delta = 0.0$

Resolution Effects for $N(\pi^-)/N(e^-) = 1$



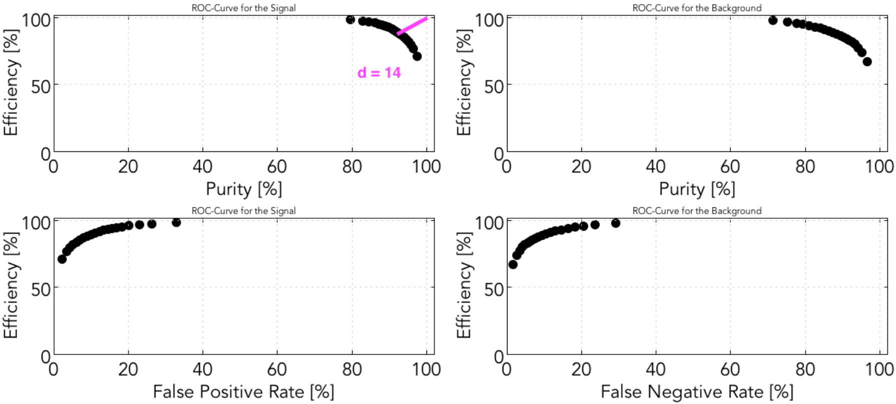
- **Basic question:** What happens if resolution in measured data is different from the resolution the classifier has been trained on?
- **Simple Test:** Apply trained classifier on a data set with different resolution than the training set
- Smeared all PID variables by a Gaussian distribution: $x \mapsto x + \text{Gauss}(0, \delta \cdot x)$
- Shown above: ROC-curves for $\delta = 5\%$

Resolution Effects for $N(\pi^-)/N(e^-) = 1$



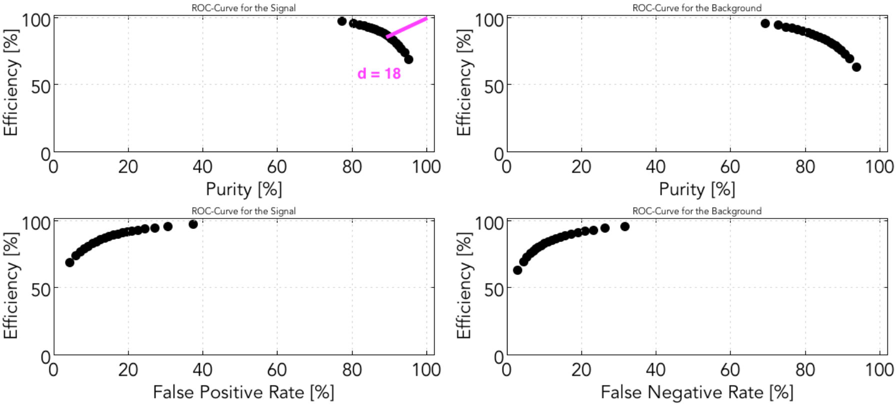
- **Basic question:** What happens if resolution in measured data is different from the resolution the classifier has been trained on?
- **Simple Test:** Apply trained classifier on a data set with different resolution than the training set
- Smeared all PID variables by a Gaussian distribution: $x \mapsto x + \text{Gauss}(0, \delta \cdot x)$
- Shown above: ROC-curves for $\delta = 15\%$

Resolution Effects for $N(\pi^-)/N(e^-) = 1$



- **Basic question:** What happens if resolution in measured data is different from the resolution the classifier has been trained on?
- **Simple Test:** Apply trained classifier on a data set with different resolution than the training set
- Smeared all PID variables by a Gaussian distribution: $x \mapsto x + \text{Gauss}(0, \delta \cdot x)$
- Shown above: ROC-curves for $\delta = 25\%$

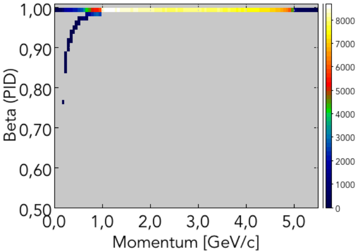
Resolution Effects for $N(\pi^-)/N(e^-) = 1$



- **Basic question:** What happens if resolution in measured data is different from the resolution the classifier has been trained on?
- **Simple Test:** Apply trained classifier on a data set with different resolution than the training set
- Smeared all PID variables by a Gaussian distribution: $x \mapsto x + \text{Gauss}(0, \delta \cdot x)$
- Shown above: ROC-curves for $\delta = 35\%$

Summary and Outlook

- ☑ Set up and use Machine Learning for PID
- ☑ Performed identification of e^- in a π^--dominated data set
  - Key variables: p, nphe(LTCC), nphe(HTCC), $\Delta E(pcal)$, $\Delta E(ec:in)$, $\Delta E(ec:out)$
  - Use two classifier $\Rightarrow$ chosen by ROC-metric $\Rightarrow$ Both performed equally: $\epsilon_{e^-} \sim 95\%$
  - Checked influence of magnetic field settings $\Rightarrow$ No significant impact indicated
  - Studied resolution effects $\Rightarrow$ In this analysis: managable performance for $\delta < 15\%$
- ☐ Test methods on KPP-Data
- ☐ Perform PID studies for e^+, π^+, $K^\pm$ and p (ongoing)
- ☐ Estimation of Variable Importance (ongoing)
- ☐ Include further classifier algorithms (e.g. SVM, Likelihood,...) (ongoing)
- ☐ Possible X-check of PID-results (e.g. compare β(PID) with β(ToF))

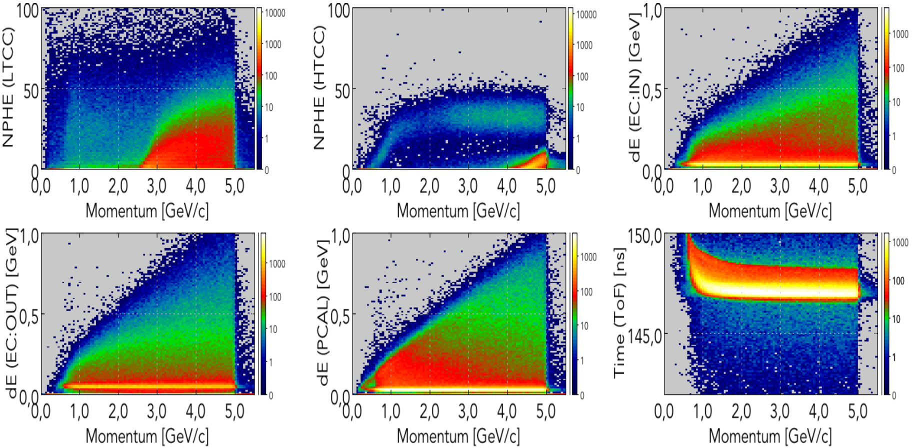


Content

Backup Stuff

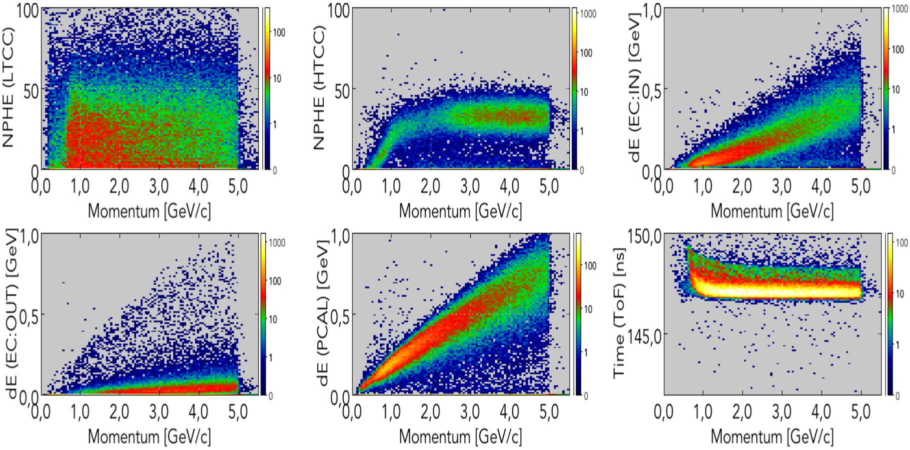
Backup: The $e^-\pi^-$ Data Set

All Particles



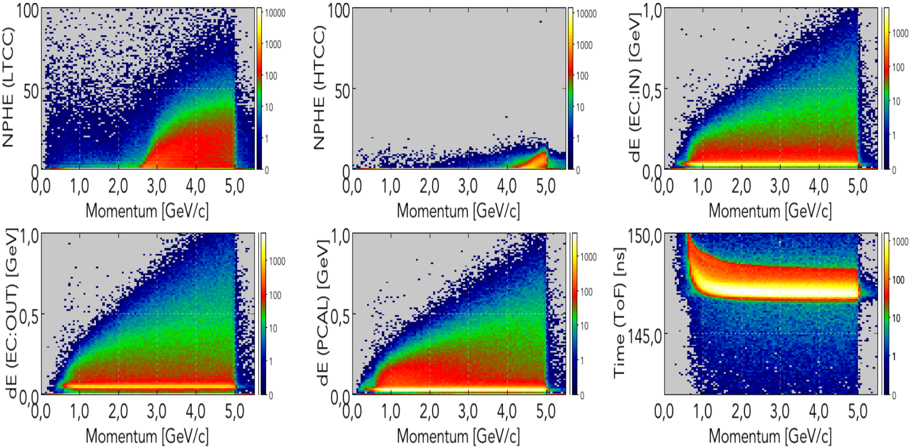
Backup: The $e^-\pi^-$ Data Set

Electrons

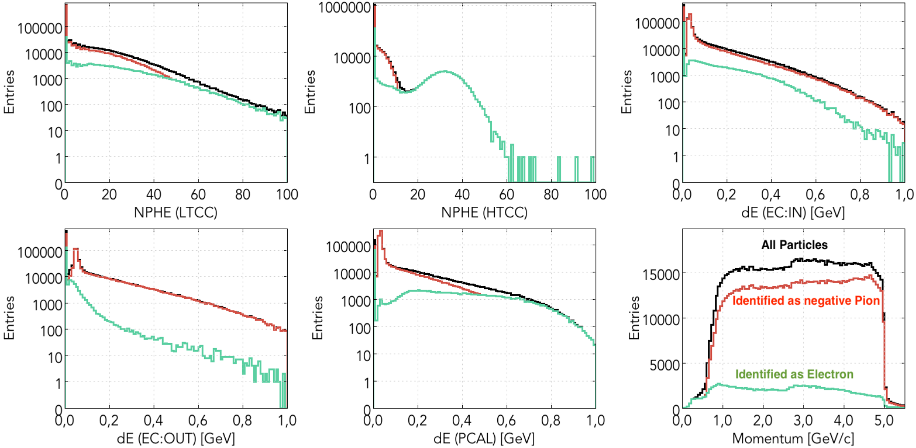


Backup: The $e^-\pi^-$ Data Set

Negative Pions



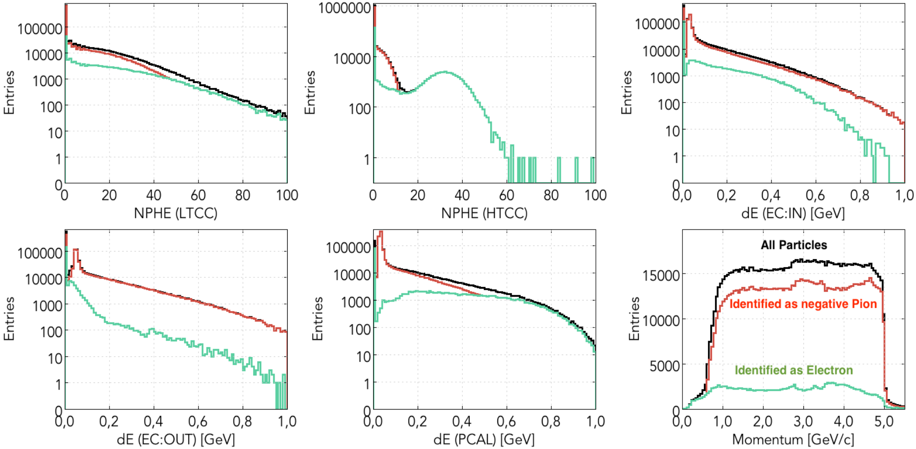
Backup: 1D Distributions before and after PID



Apply neural network algorithm on $e^-\pi^-$ data set:

- $\epsilon_S = 94\%$
- $P_S = 62\%$

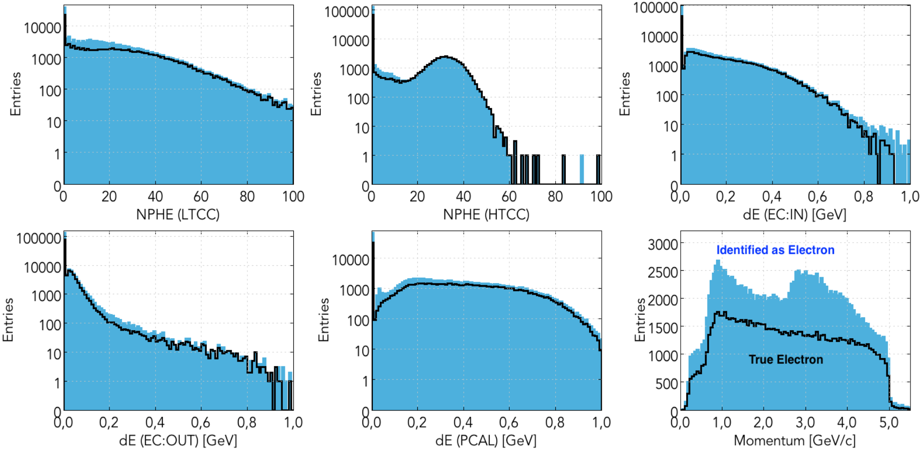
Backup: 1D Distributions before and after PID



Apply boosted decision tree algorithm on $e^{-}\pi^{-}$ data set:

- $\epsilon_S = 96\%$
- $P_S = 58\%$

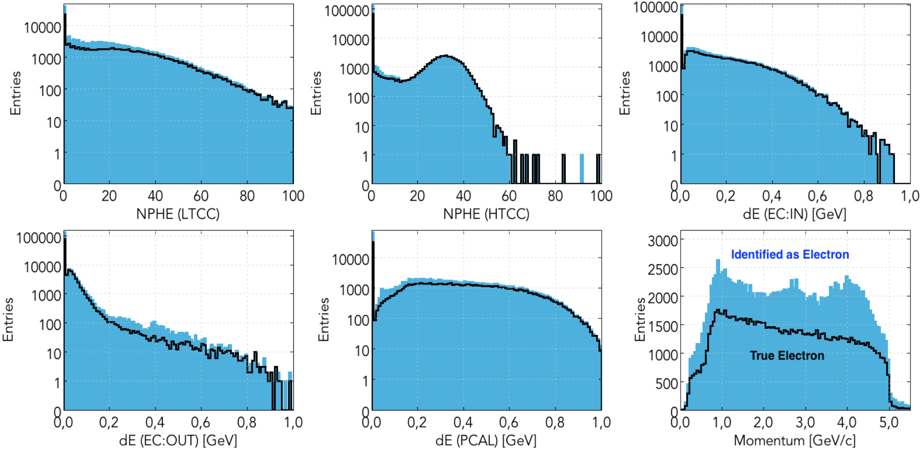
Backup: Compare True and ID Distributions



Apply neural network algorithm on $e^{-}\pi^{-}$ data set:

- $\epsilon_S = 94\%$
- $P_S = 62\%$

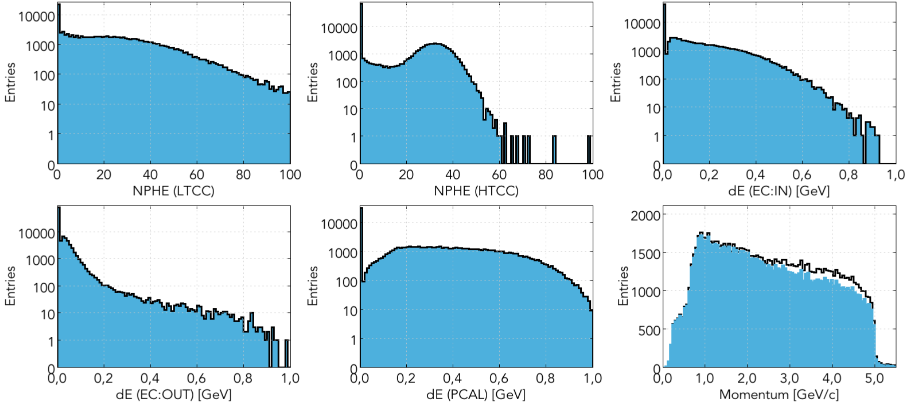
Backup: Compare True and ID Distributions



Apply boosted decision tree algorithm on $e^-\pi^-$ data set:

- $\epsilon_S = 96\%$
- $P_S = 58\%$

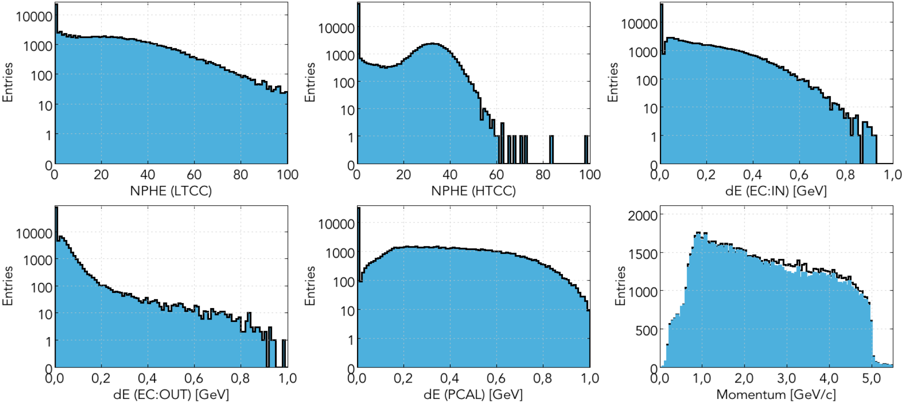
Backup: Compare True and ID Distributions for true Electrons



Apply neural network algorithm on $e^{-}\pi^{-}$ data set:

- $\epsilon_S = 94\%$
- $P_S = 62\%$

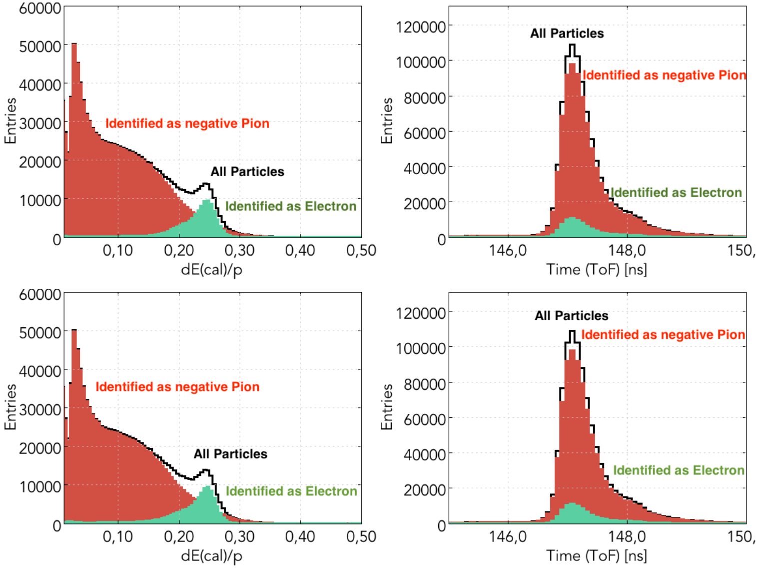
Backup: Compare True and ID Distributions for true Electrons



Apply boosted decision tree algorithm on $e^-\pi^-$ data set:

- $\epsilon_S = 96\%$
- $P_S = 58\%$

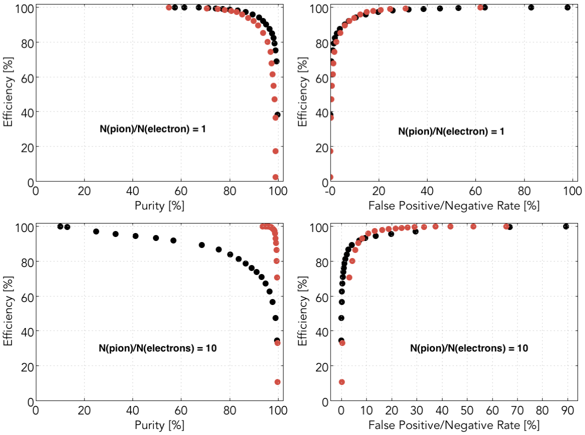
Backup: Comparison of PID-Plots



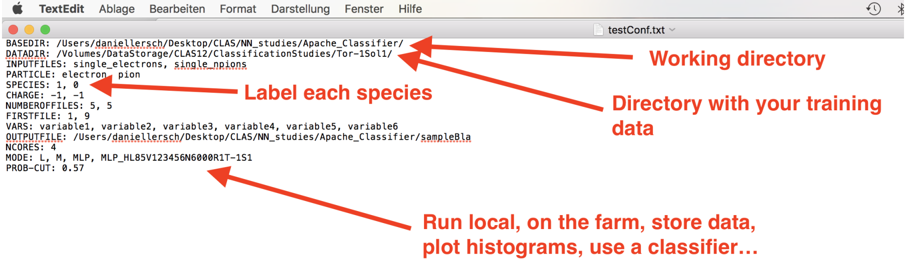
Top: MLP / Bottom: GBT

Backup: Useful Relations

- ϵ_S and FPR are features of the classifier and (ideally) independent of the ratio between the species
- $\left(\frac{N_B^{all}}{N_S^{all}}\right)_{rec} = \left[\frac{P_S}{\epsilon_S} \times \frac{\epsilon_B}{P_B}\right] \times \left(\frac{N_B^{all}}{N_S^{all}}\right)_{true}$
- $P_S = \epsilon_S \times \left[\epsilon_S + FPR \times \left(\frac{N_B^{all}}{N_S^{all}}\right)_{true}\right]^{-1}$



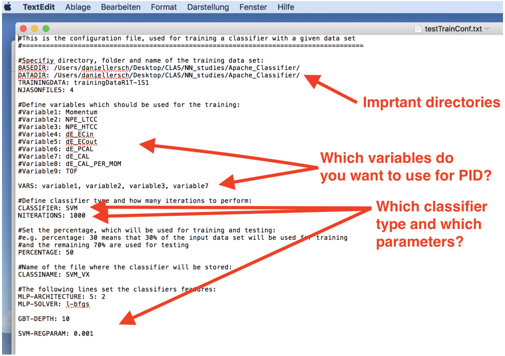
Backup: Generating single Track Training Data




```
BASEDIR: /Users/daniellersch/Desktop/CLAS/NN_studies/Apache_Classifier/
DATADIR: /Volumes/DataStorage/CLAS12/ClassificationStudies/Tor-1Sol1/
INPUTFILES: single_electrons, single_npions
PARTICLE: electron, pion
SPECIES: 1, 0
CHARGE: -1, -1
NUMBEROFFILES: 5, 5
FIRSTFILE: 1, 9
VARS: variable1, variable2, variable3, variable4, variable5, variable6
OUTPUTFILE: /Users/daniellersch/Desktop/CLAS/NN_studies/Apache_Classifier/sampleBla
NCORES: 4
MODE: L, M, MLP, MLP_HL85V123456N6000R1T-1S1
PROB-CUT: 0.57
```

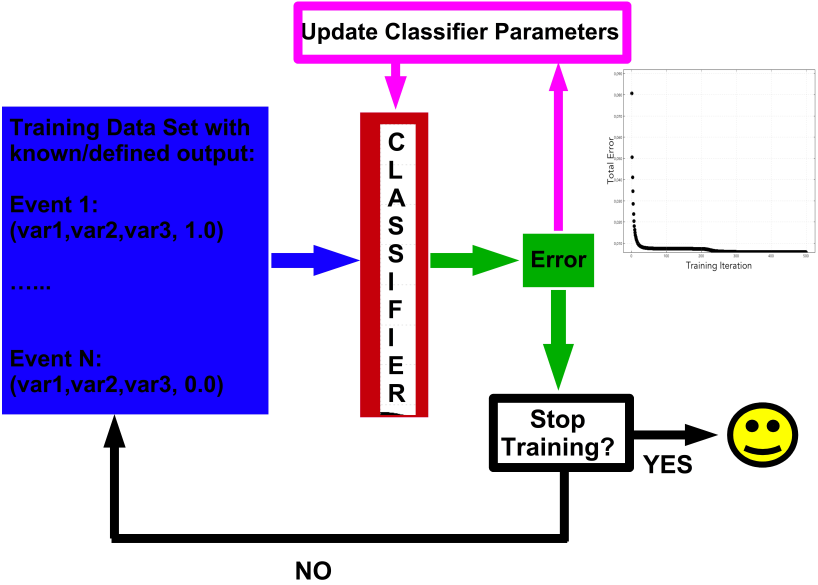
- Java-based Module:
  - Generate training data from reconstructed CLAS12 data
  - Test a classifier on reconstructed CLAS12 data
  - Can be run by/with multiple users/configurations
- Module parameters are specified in a txt-file (see top figure)
- Txt-file generation and submission to farm are run by a single script

Backup: Training and Choice of the Classifier

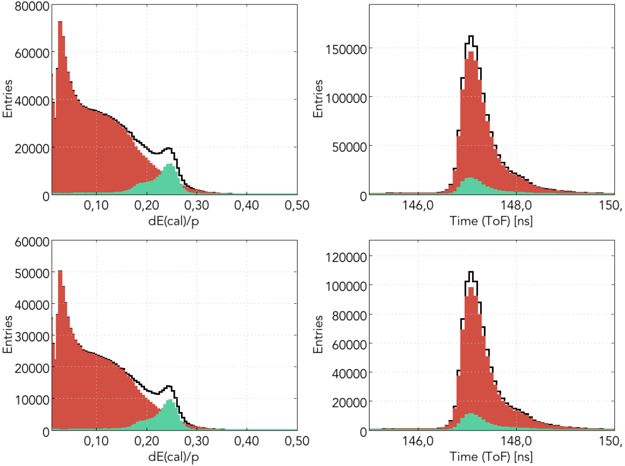


- Java-based Module:
  - Specify the variables used for training
  - Select the classifier that shall be trained
  - Define how the classifier shall be trained
  - Can be run by/with multiple users/configurations
- Module parameters are specified in a txt-file (see top figure)
- Txt-file generation and submission to farm are run by a single script

Backup: Training a Classifier



Backup: Changing the Magnetic Field



- Top: Classifier trained on: Tor. $= -0.75$ and Sol. $= 0.8$ and applied on: Tor. $= -0.75$ and Sol. $= 0.8$
- Bottom: Classifier trained on: Tor. $= -0.75$ and Sol. $= 0.8$ and applied on: Tor. $= -1$ and Sol. $= 1$