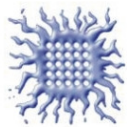
Towards an MVA PID

S. Lukić

Vinča institute of nuclear sciences, University of Belgrade

ILD analysis/software workshop 22-26. Feb. 2016

HEP GROUP VINCA



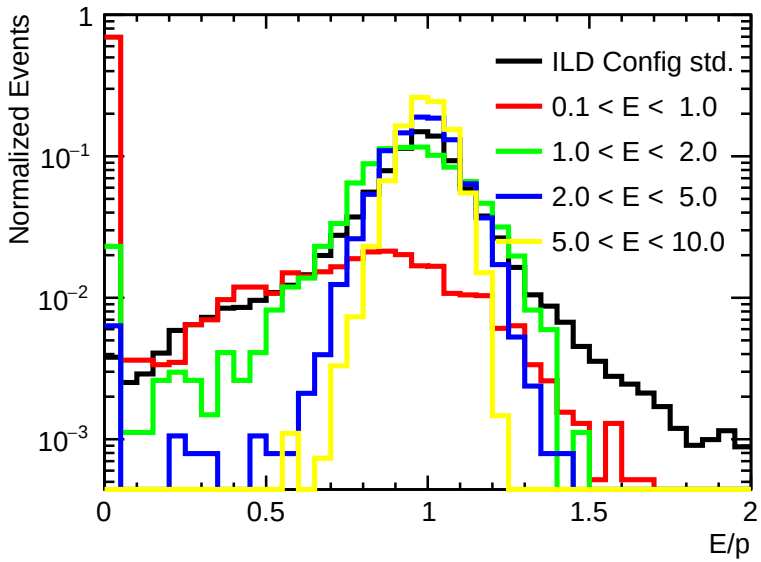
- `PIDTools` contains a set of processors in an advanced state of development.
- `LikelihoodPIDProcessor` combines three PID algorithms:
  - "basic" (using total deposits from ECAL, HCAL and muon system)
  - Cluster shapes (shape data written by a separate processor )
  - $\mathrm{d}E/\mathrm{d}x$ in the tracker (data written by a separate processor )
- Helper classes developed to organise the *hypotheses* and the *data* (Not yet in the trunk)

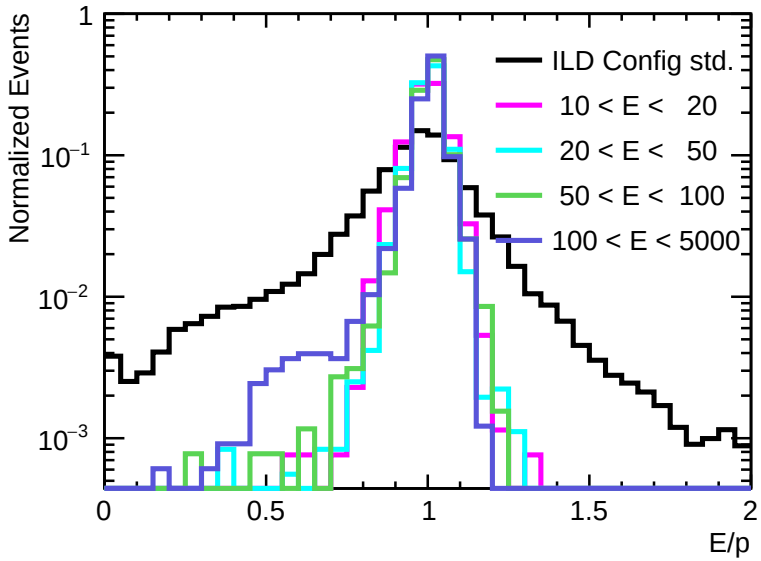
Momentum dependence of the sensitive variables

Distributions of (ECAL+HCAL)/p for ranges of p measured in the tracker

Electrons

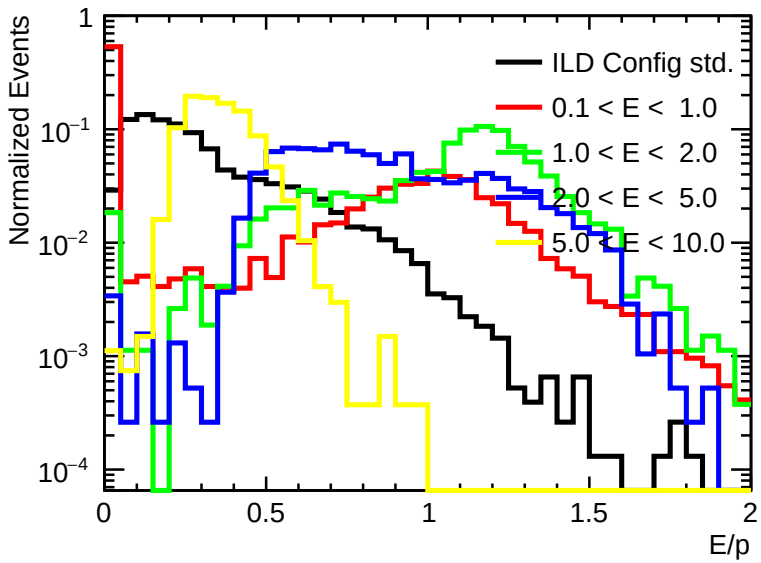
Also shown the overall distribution from ILD Standard config





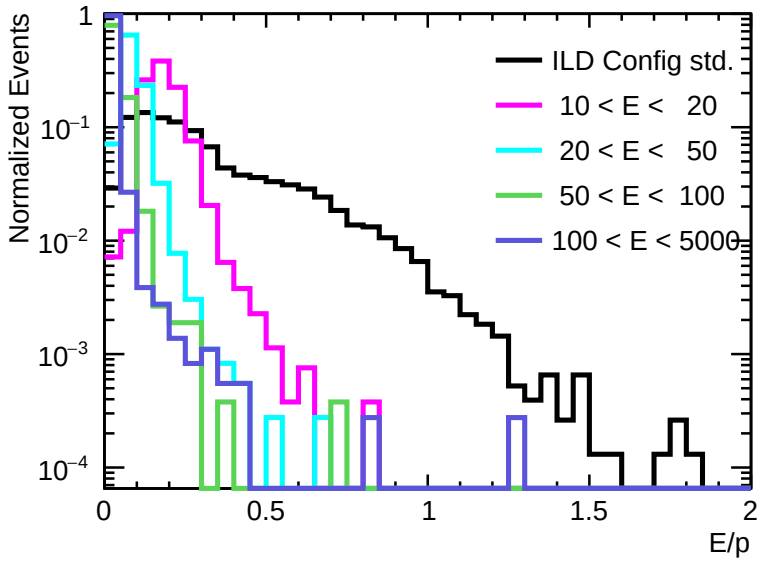
Momentum dependence of the sensitive variables

Distributions of (ECAL+HCAL)/p for ranges of p measured in the tracker



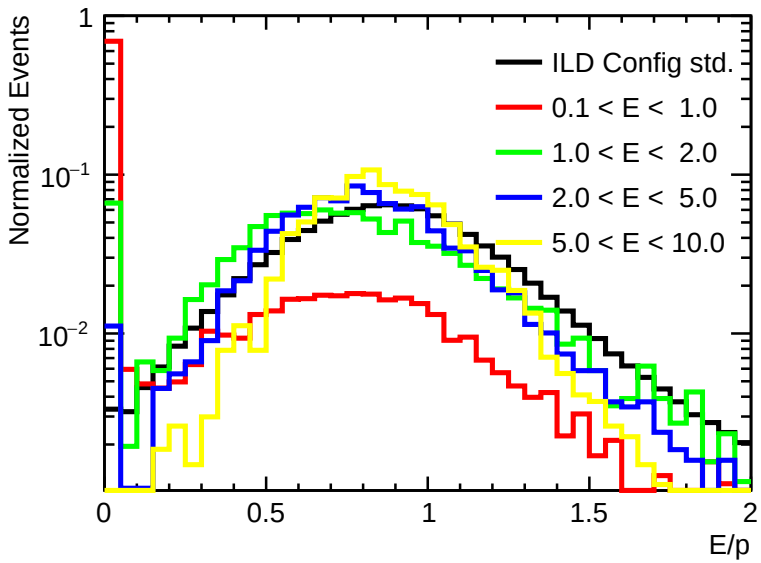
Muons

Also shown the overall distribution from ILD Standard config



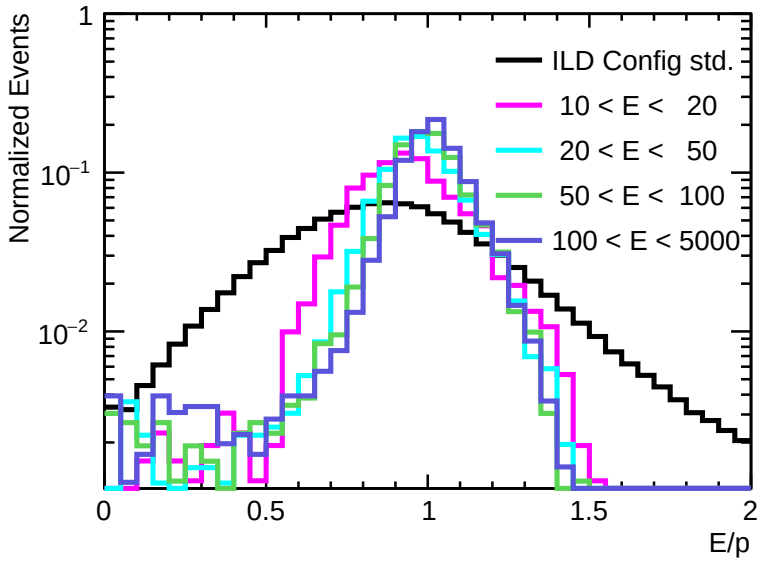
Momentum dependence of the sensitive variables

Distributions of (ECAL+HCAL)/p for ranges of p measured in the tracker



Pions

Also shown the overall distribution from ILD Standard config

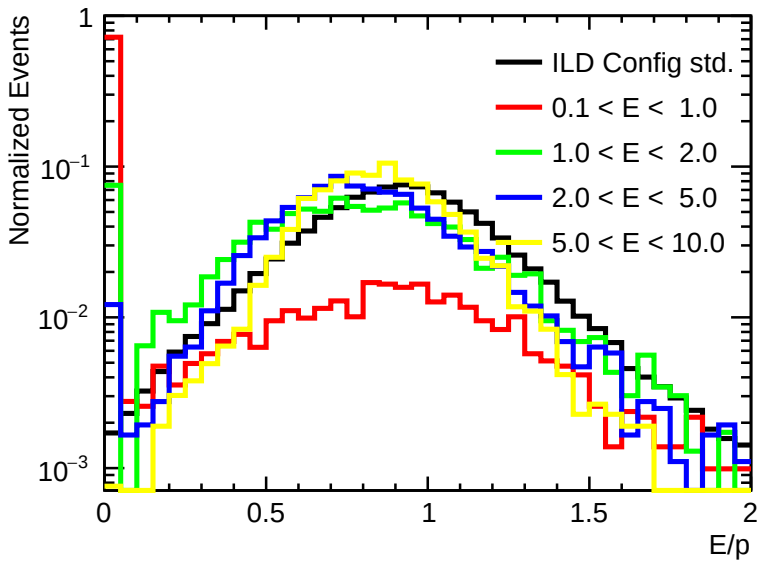


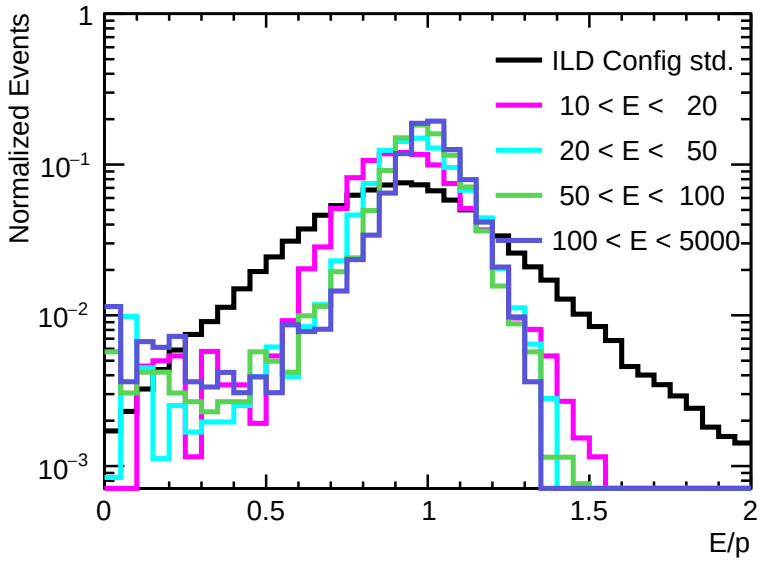
Momentum dependence of the sensitive variables

Distributions of (ECAL+HCAL)/p for ranges of p measured in the tracker

Kaons

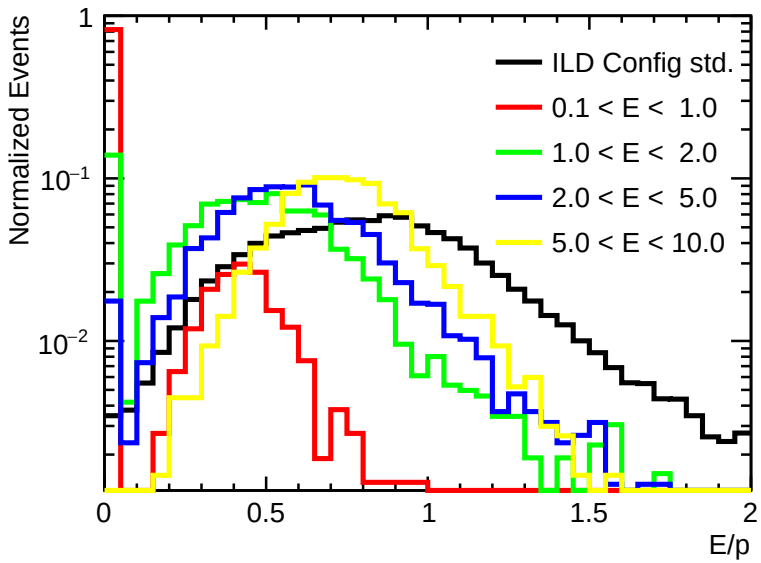
Also shown the overall distribution from ILD Standard config





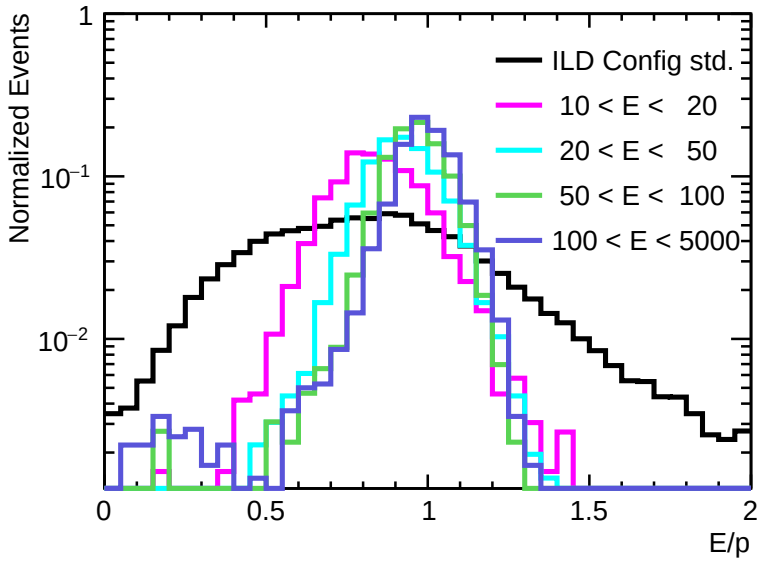
Momentum dependence of the sensitive variables

Distributions of (ECAL+HCAL)/p for ranges of p measured in the tracker



Protons

Also shown the overall distribution from ILD Standard config



Momentum dependence of the sensitive variables

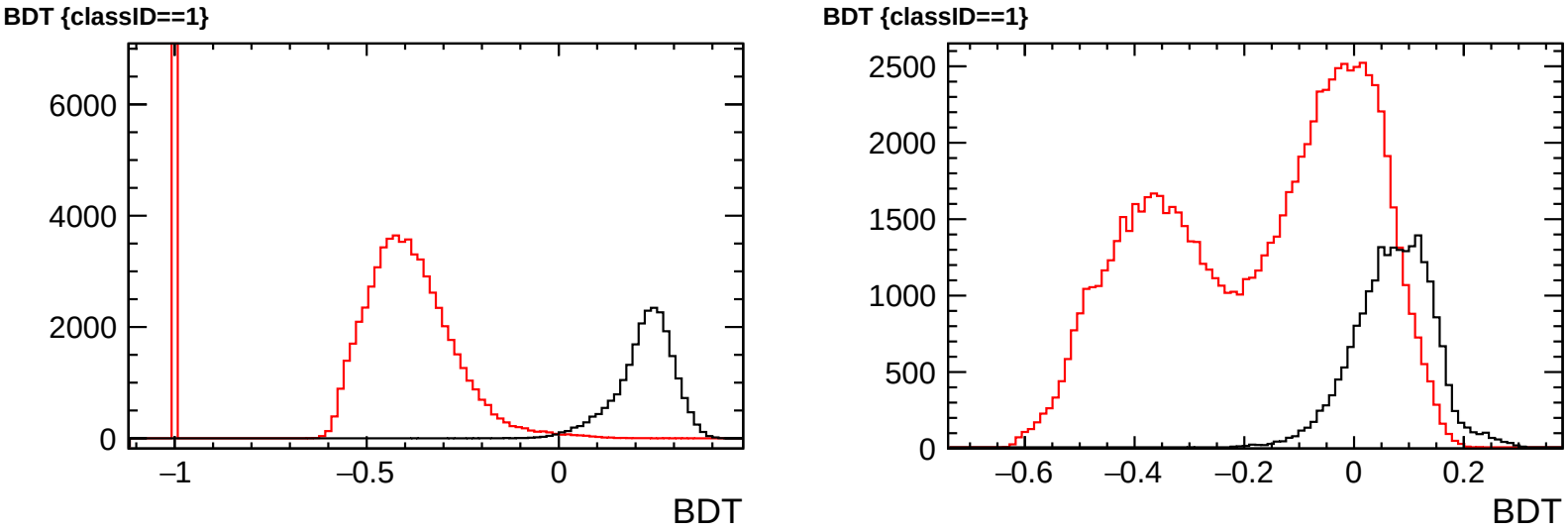
Summary:

- Calorimetric sensitive variables evolve with p – higher sensitivity should be possible if measured variables were compared to p-dependent distributions
- Suggestion by Junping – Move to MVA (should solve the p-dependence and correlations automatically)

MVA Strategy issues

- Not a simple signal/background classification. Multiple categories – different possible ways to proceed (see following slides)
- Training sample: Single particles or physics samples
  - Single particles: What relative weights should one give to different particle types?
  - Physics sample: Is the sample representative of the "real life" situation

A possible classification strategy



a) b)

Signal and background BDT output for a) electrons, b) kaons
Only particles with associated clusters and with $dE/dx > 0$ used for training

- Separate training for each hypothesis vs. all others
- Define a cut by some criterion (e.g. 99% of bkg removed) for each training
- How many hypotheses pass the cut?
  - One – job done
  - Several – make decision by a test statistic $Q_>$
  - None – make decision by a test statistic $Q_<$

More about the test statistics

- The metric of te MVA output is arbitrary – only the ordering of particles has a sense
- Usable quantities:

$$\int_x^1 f_s(x')\,\mathrm{d}x'$$
$$\int_x^1 f_b(x')\,\mathrm{d}x'$$

Helper classes – PIDParticles

- PIDParticles namespace with tools to organise the hypotheses (Currently: *electron*, *muon*, *pion*, *kaon*, *proton*):
  - Map of `PIDParticle_base` objects containing basic info about particles PDG code, mass, prior, posterior, logL, Bethe-Bloch curve parameters, etc. – easier and safer coding + easier to include additional particles.
  - Maps of objects of derived classes `LLPIDHypothesis` and `MVAPIDHypothesis` adding info required by specific PID algorithms
  - Loops over particle types can now use map iterators...

Helper classes – PIDVariables

- **Background:** PID processors use total calorimetry deposits, as well as output from processors for cluster shapes and $\mathrm{d}E/\mathrm{d}x$ to calculate a set of 10-12 sensitive variables (*data*) for the distinction between particle types.
- Same set of variables is calculated when training MVA PID, or making histograms of characteristic distributions for Likelihood PID.
- `PIDVariables` class organises the data:
  - Provides a map of sensitive variables.
  - `void Update(EVENT::ReconstructedParticle*);` calculates the values of the variables for a `ReconstructedParticle` object (cluster shapes and $\mathrm{d}E/\mathrm{d}x$ processors have already been applied).
  - Other relevant variables are also stored (e.g. $\mathrm{d}E/\mathrm{d}x$, track energy) needed for debugging, testing, development etc.
  - Can be used by every processor or code that needs it
    Advantage: Unique code for the calculation of variables.

Outlook

- Working on a MVA PID
- Strategy details to be optimised (and made steerable?)
- Separate training for PFO that lack part of information (calorimetry or $\mathrm{d}E/\mathrm{d}x$)

Thanks!