Data Analysis

Mike Williams
MIT
July 28, 2016

Some Basics of Statistical Inference

Hypothesis Testing

Let's denote some test statistic T with PDF g(T). We then define a p-value as

$$p = \int_T^\infty g_{f_0}(T')dT'$$

The p-value is the probability of finding a T-value corresponding to lesser agreement than the observed T-value if the hypothesis is true -- it is NOT the probability that the hypothesis is true!

If true, the p-value distribution is uniform on (0,1). One can then reject the hypothesis at CL a if $p > 1-a$; e.g., a test hypothesis is rejected at 95% CL if $p < 0.05$. If true, 5% of experiments should be rejected!

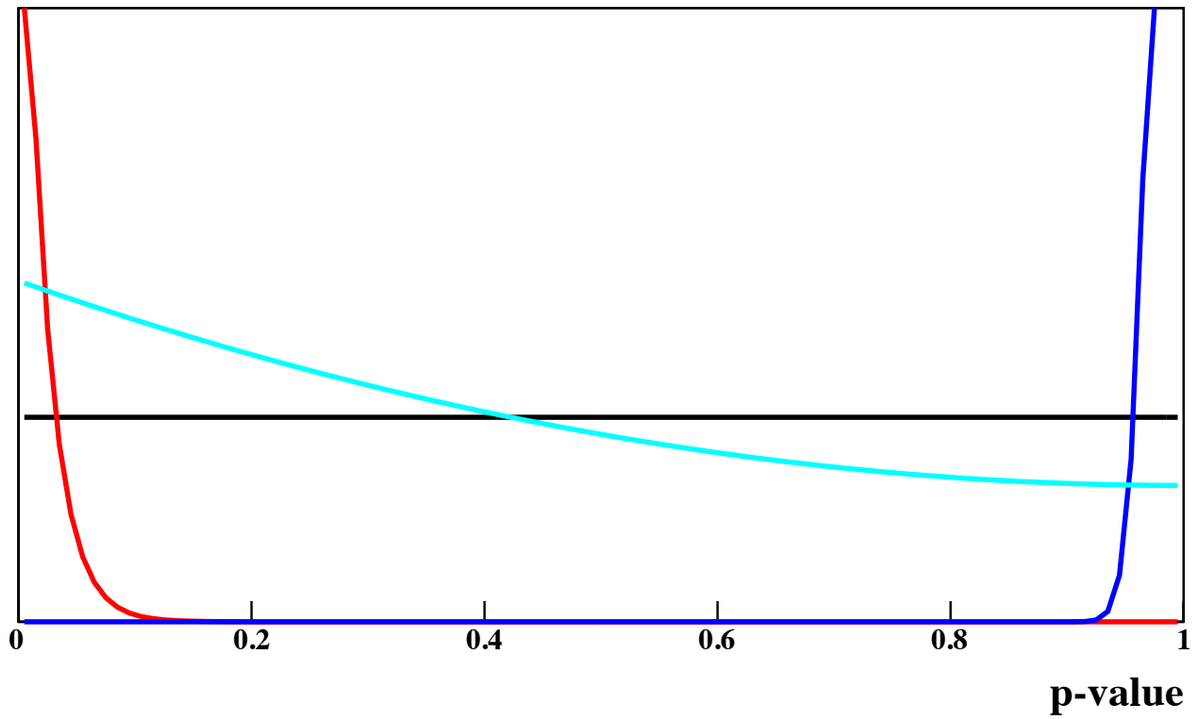
N.b., X^2 is so popular because it's easy to calculate and g(T) does not depend on the hypothesis (technically only true in the n→infinity limit).

See: MW, How good are your fits? Unbinned multivariate GOF tests in higher energy physics, JINST 5 (2010) P09004.

p-values

Consider a simple example: I give a physics test to 1M physics students and build g(T) where T=test score. Then, I give this test to a student I don't know and try to determine whether they are a physics student.

physics students, art students, math students, professors



"Good" p-values don't mean my hypothesis was true. The test may be insensitive to differences in hypotheses.

There is no uniformly most powerful test for all problems. Choose wisely for each application you encounter.

Significance

The statistical significance is the probability that a given result would occur by chance assuming a “null” hypothesis is true. Physicists tend to quote this in “units” of σ, but this does NOT mean the problem is assumed to be Gaussian.

Wilks’ Theorem: For “nested” models, $-2\Delta\log(L)$ is asymptotically X^2 distributed with n(dof) = Δn(par). Physicists screw this up regularly:

- the models must be nested (e.g., signal Gaussian with mean and/or width free cannot be nested with a no signal model!);
- converting to nσ using $\sqrt{(-2\Delta\log(L))}$ is only (possibly) valid if Δn(par)=1;
- this is only asymptotically valid.

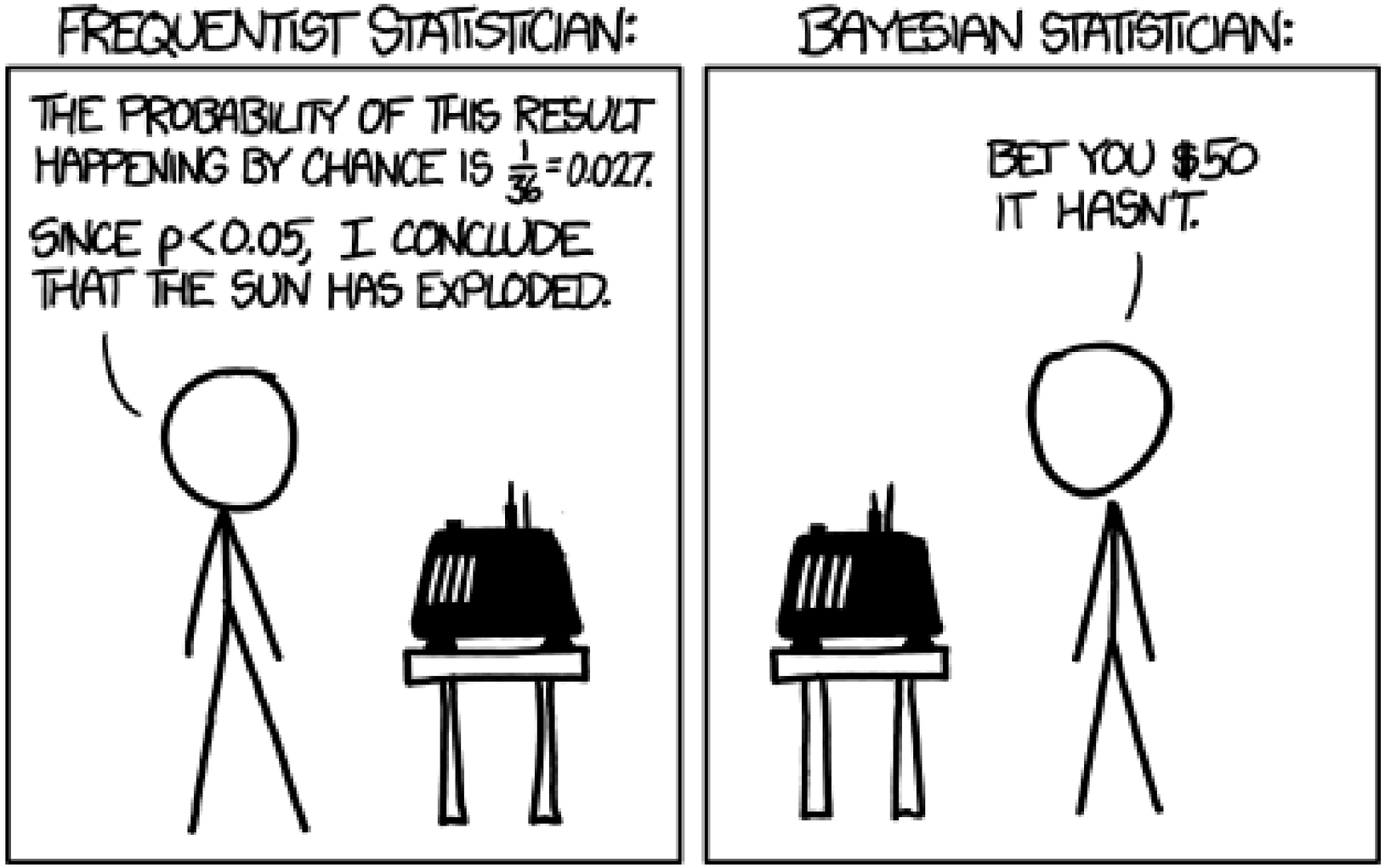
Alternative is to generate MC data sets and extract $-2\Delta\log(L)$ on each to get its PDF. P.S. Don’t forget the trials factor (look elsewhere effect).

Frequentist vs Bayesian

You will inevitably encounter this debate: Frequentist vs Bayesian. What do these terms mean? Is one of them pseudo-science/non-sense/witchcraft?

$$\text{Bayes' Theorem: } P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Has the sun exploded? Ask a machine (from a dark room) that first rolls dice, then if it gets two 6's it lies; otherwise, it tells the truth.



Frequentist vs Bayesian

Let's change the problem slightly: Replace the dice with a random number generator that throws numbers between 1 and 3.4M. If it gets 666, it lies; otherwise, it tells the truth. We try and it says "yes"!

Frequentist Physicist

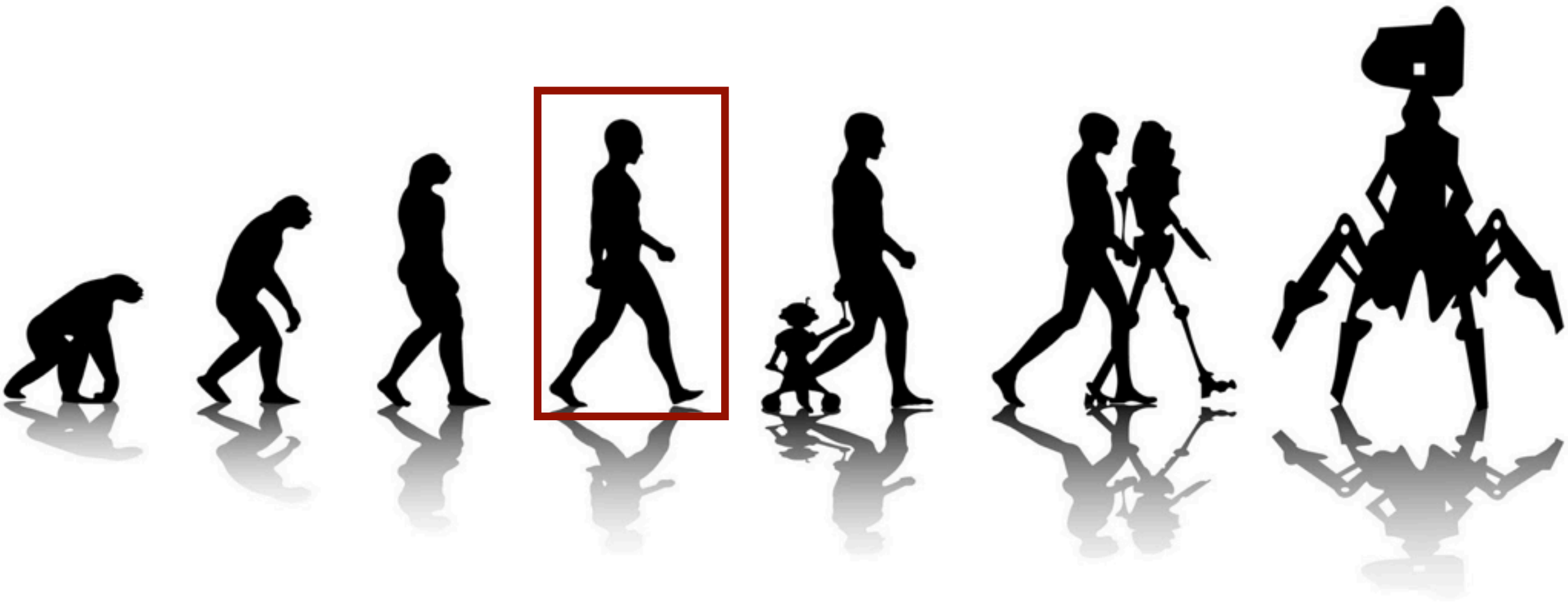
Since $p < 5\sigma$, rejects the "sun has not exploded" hypothesis but (should) accept there is a 1 in 3.4M chance of error in this claim. Should not claim the sun has exploded (but likely does anyway).

Bayesian Physicist

Bayes' Theorem says P(explode I yes) = P(yes I explode) P(explode) / P(yes). Note that we need to input a "prior belief" for P(explode). We could base this on historical astrophysics observations and solar models -- but surely it's a very small number (ε). Let's also define P(lie) = λ.

So, P(explode I yes) = $(1-\lambda)\varepsilon/[(1-\lambda)\varepsilon+\lambda(1-\varepsilon)] \sim \varepsilon/(\varepsilon+\lambda)$. If $\varepsilon \ll \lambda$, then we have P(explode I yes) $\sim \varepsilon/\lambda$; i.e., our belief that the sun has exploded will be 3.4M times bigger but still very very small.

Some Modern Developments



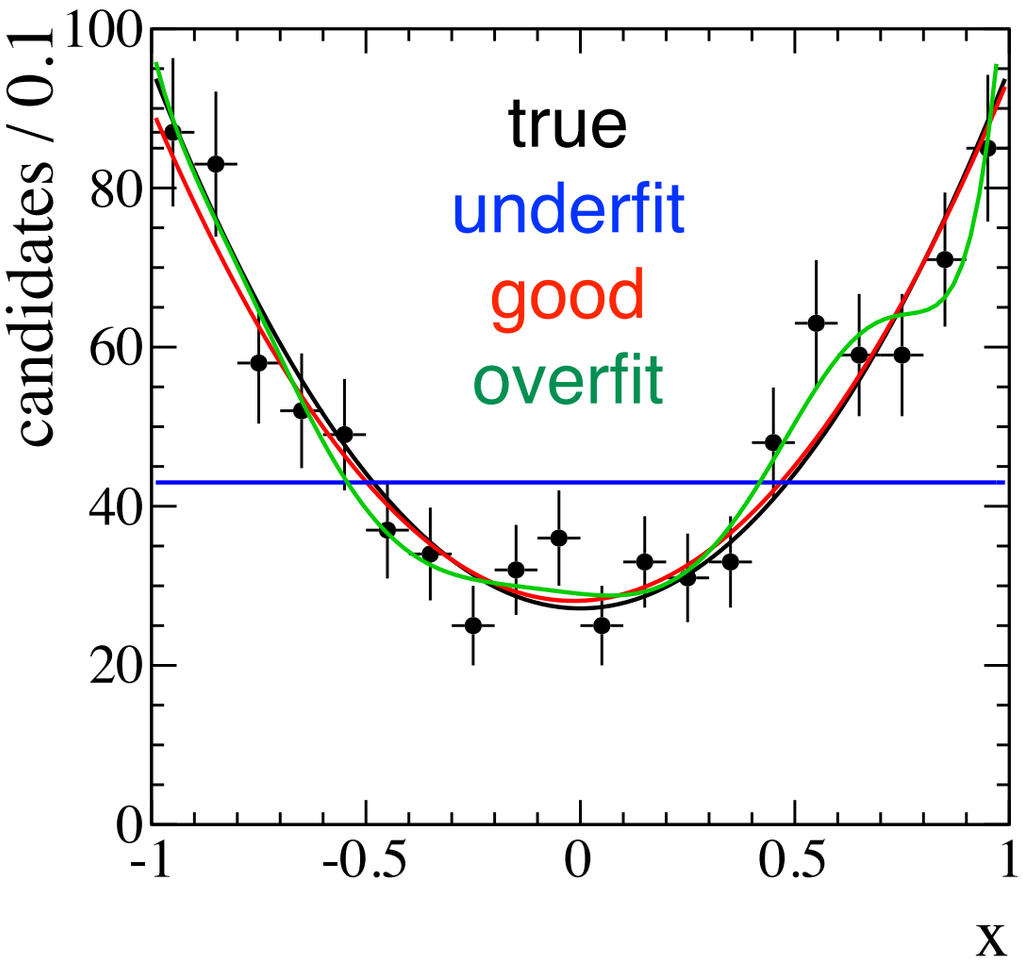
Information Criteria

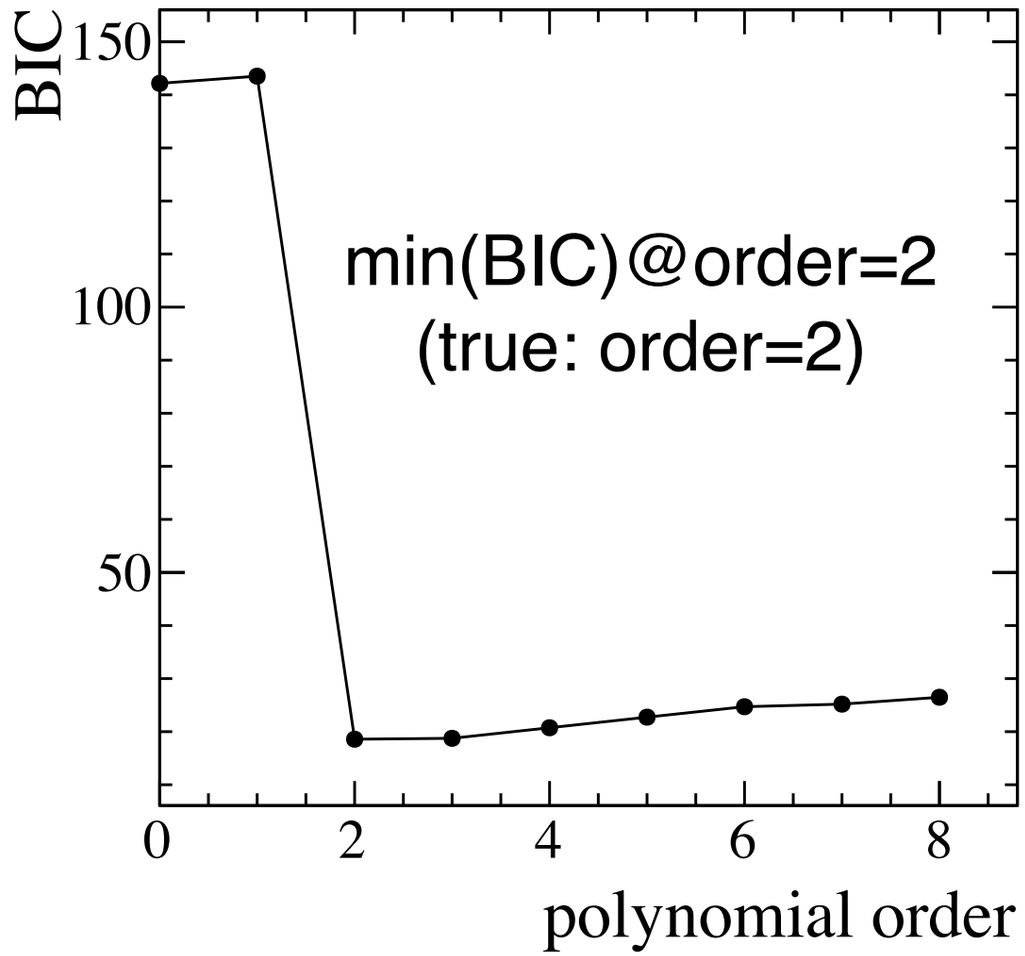
Adding complexity to a model leads to an improved goodness of fit; however, adding unnecessary complexity decreases the predictive power of the model.

N.b., technically these chi2 should be -2log(L), but for approx. Gaussian errors we can substitute the chi2.

$$\mathrm{BIC} = \chi^2 + n_p \log N$$

$$\mathrm{AIC} = \chi^2 + 2n_p$$



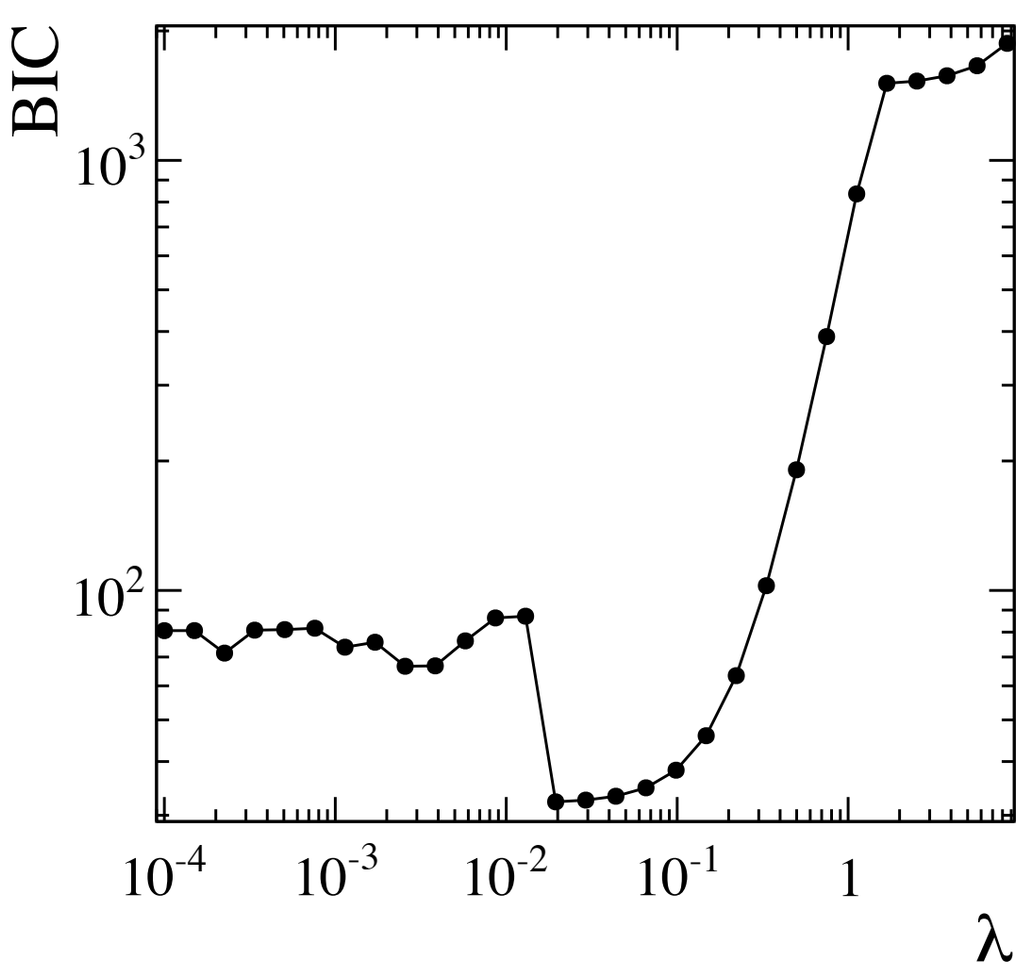


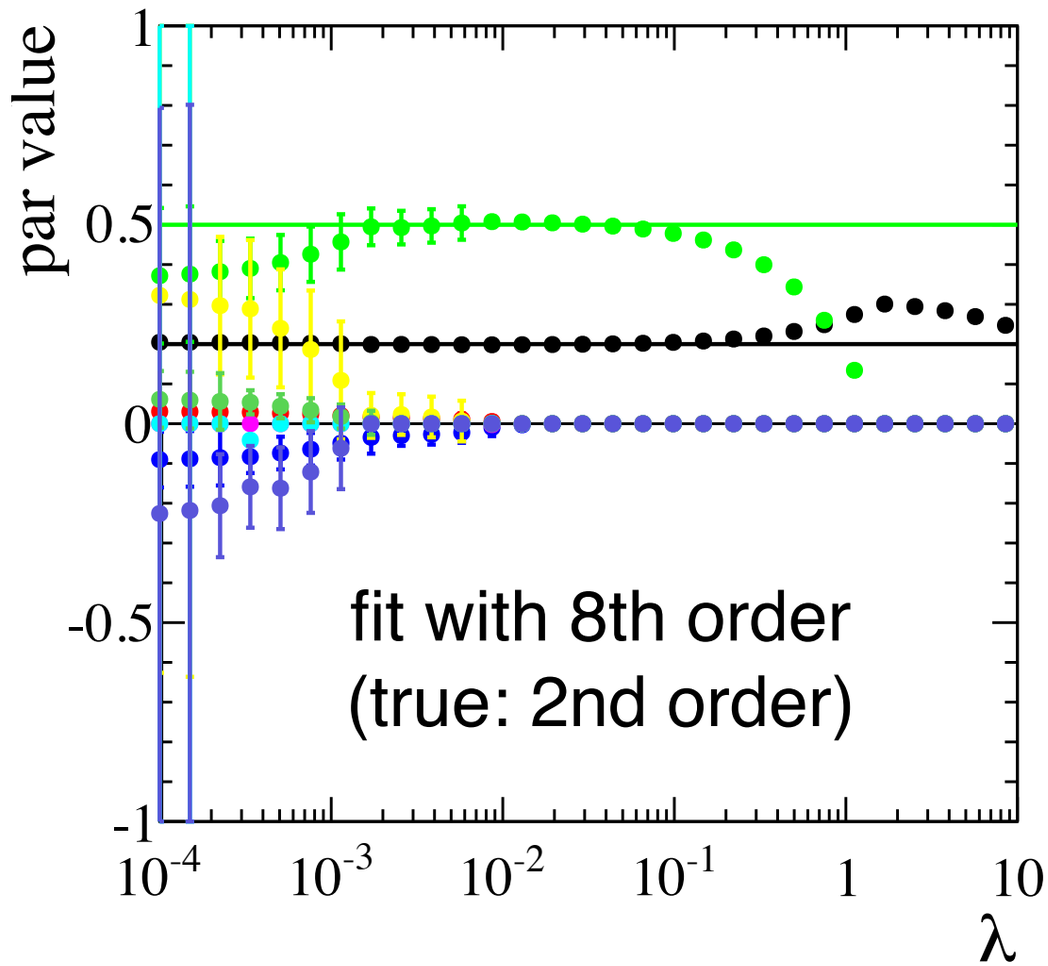
Information criteria try to select the most predictive model by balancing improved GOF with increases in complexity -- and they're simple to use!

Model Selection (LASSO)

Sometimes we don't just want a predictive model, we want to obtain estimates for model parameters -- including whether they're significantly non-zero.

$$\chi^2 \rightarrow \chi^2 + \lambda \sum |\alpha_i|$$



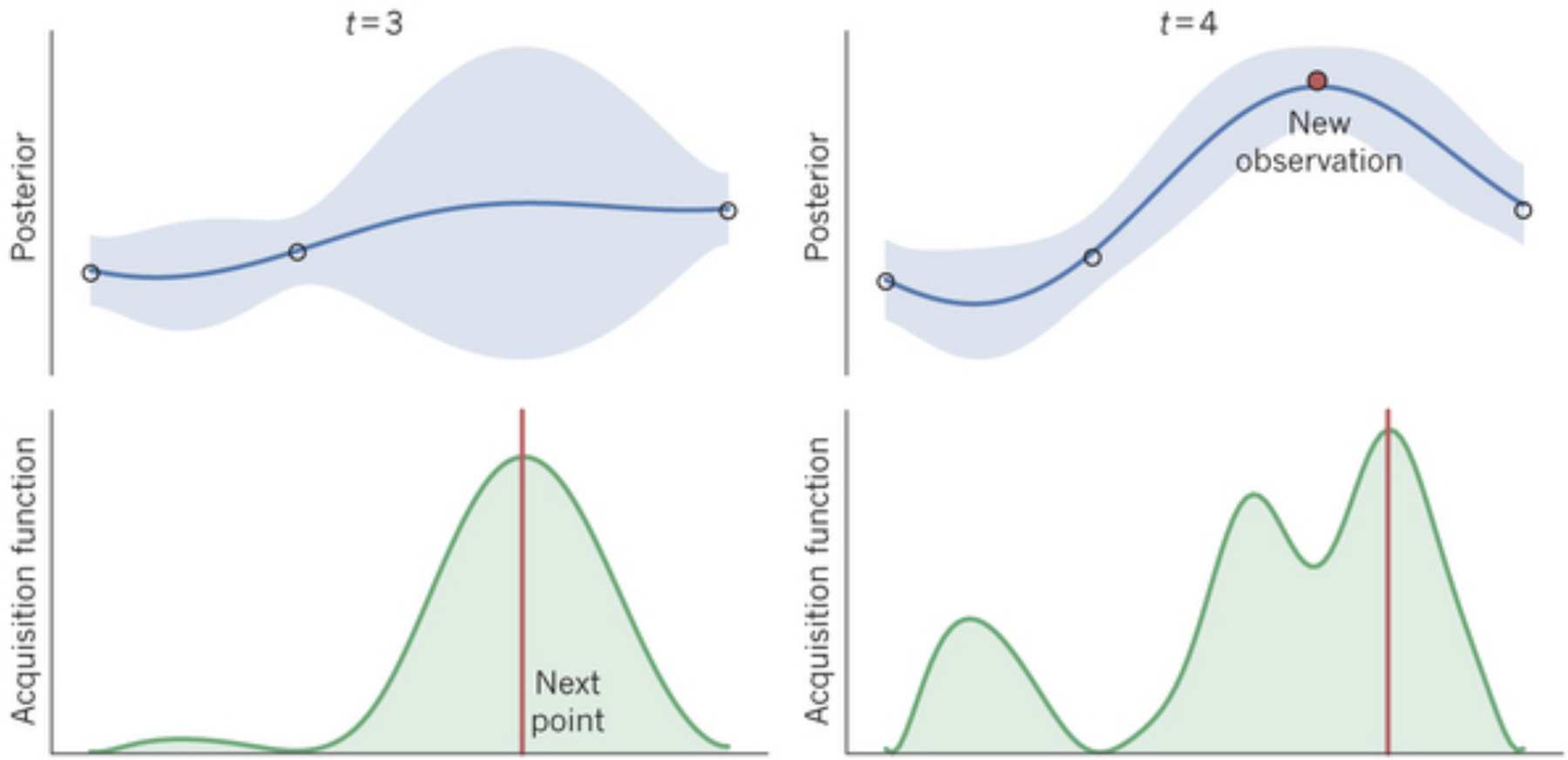


The LASSO is a simple, fast way of performing model selection without possibly introducing human bias into the process.

See: Hardin, Guegan, Stevens, MW, Model selection for amplitude analysis, JINST 10 (2015) P09002.

Bayesian Optimization

Bayesian optimization refers to a family of methods that do global optimization of black-box functions (no derivatives required).

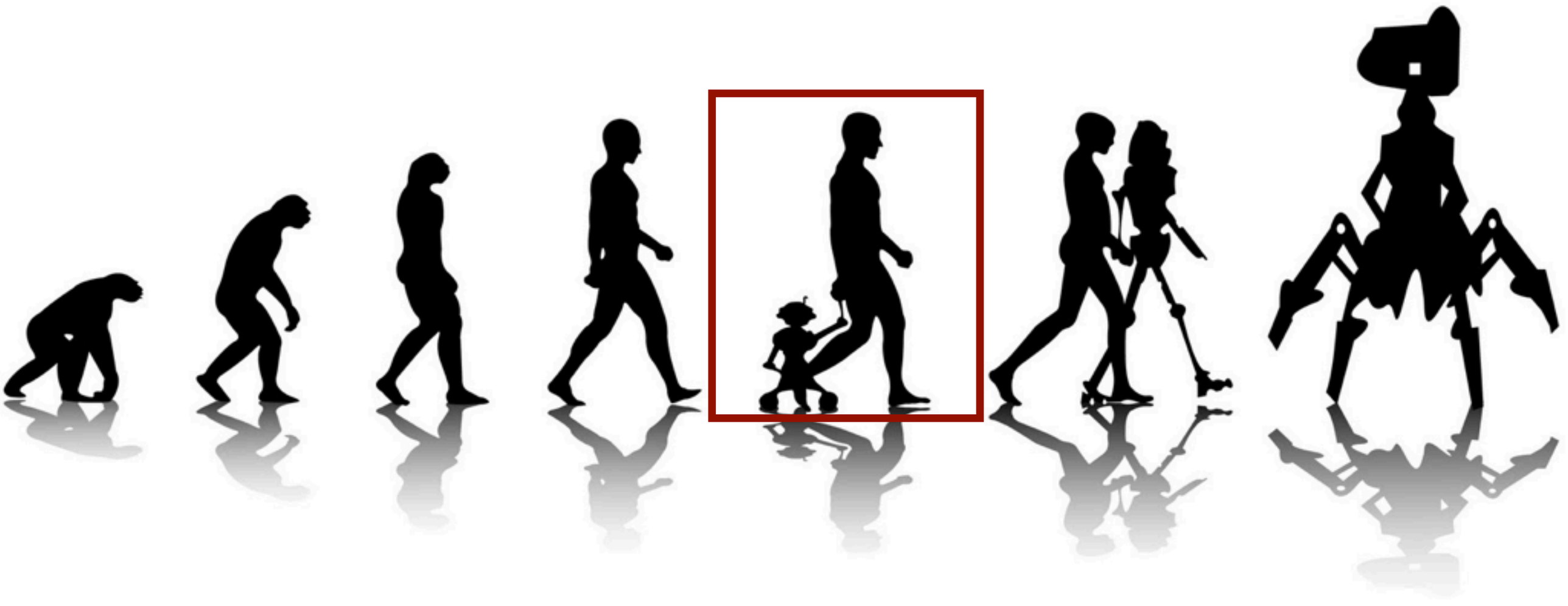


Start from prior for objective function, treat evaluations as data and produce a posterior used to determine the next point to sample.

See https://github.com/HIPS/Spearmint for an excellent package in python (also, scikit-optimize coming soon).

Example app: Ilten, MW, Yang, Monte Carlo tuning using Bayesian Optimization, to appear August 2016.

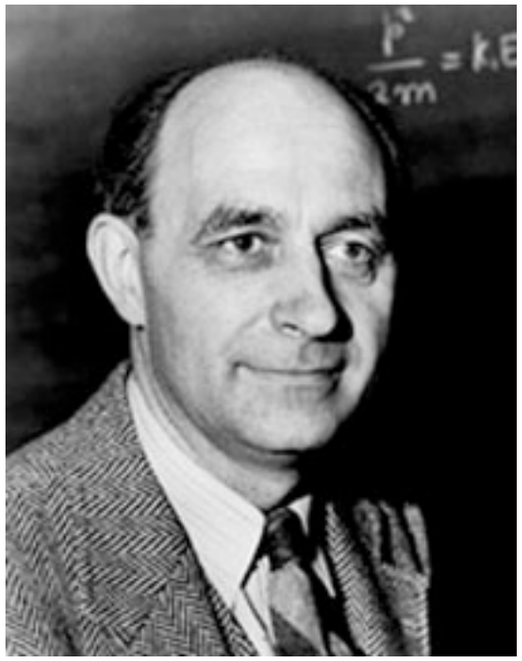
(Supervised) Machine Learning



Machine Learning

Supervised machine learning is a broad and evolving field. The most common usage in physics is training algorithms to classify data as signal or background by studying existing labeled (possibly MC) data.

teacher

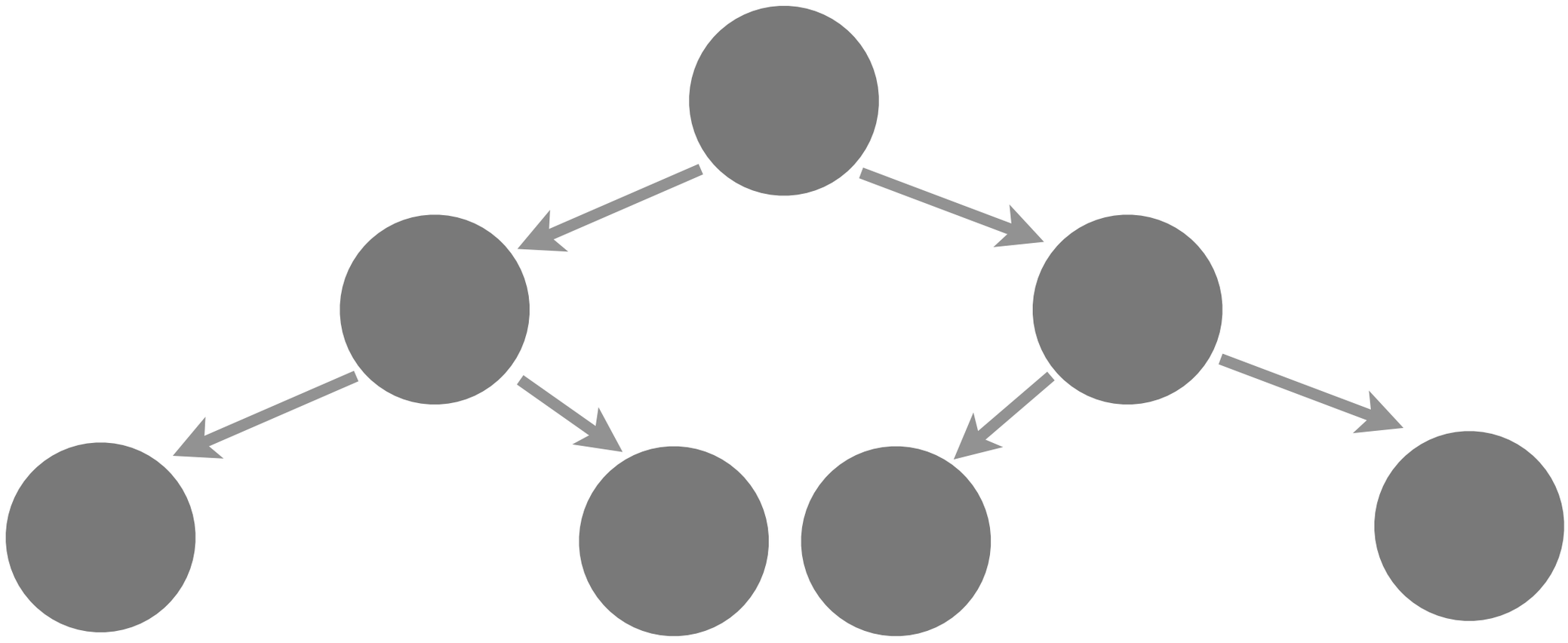


student

There are too many categories of algorithms to even attempt to list them here. In physics, most usage is either a boosted decision tree (BDT) or artificial neural network (ANN) -- so I'll briefly describe these.

Decision Trees

Start with a labeled training data sample. These may be obtained from simulation, data sidebands, control modes, etc.

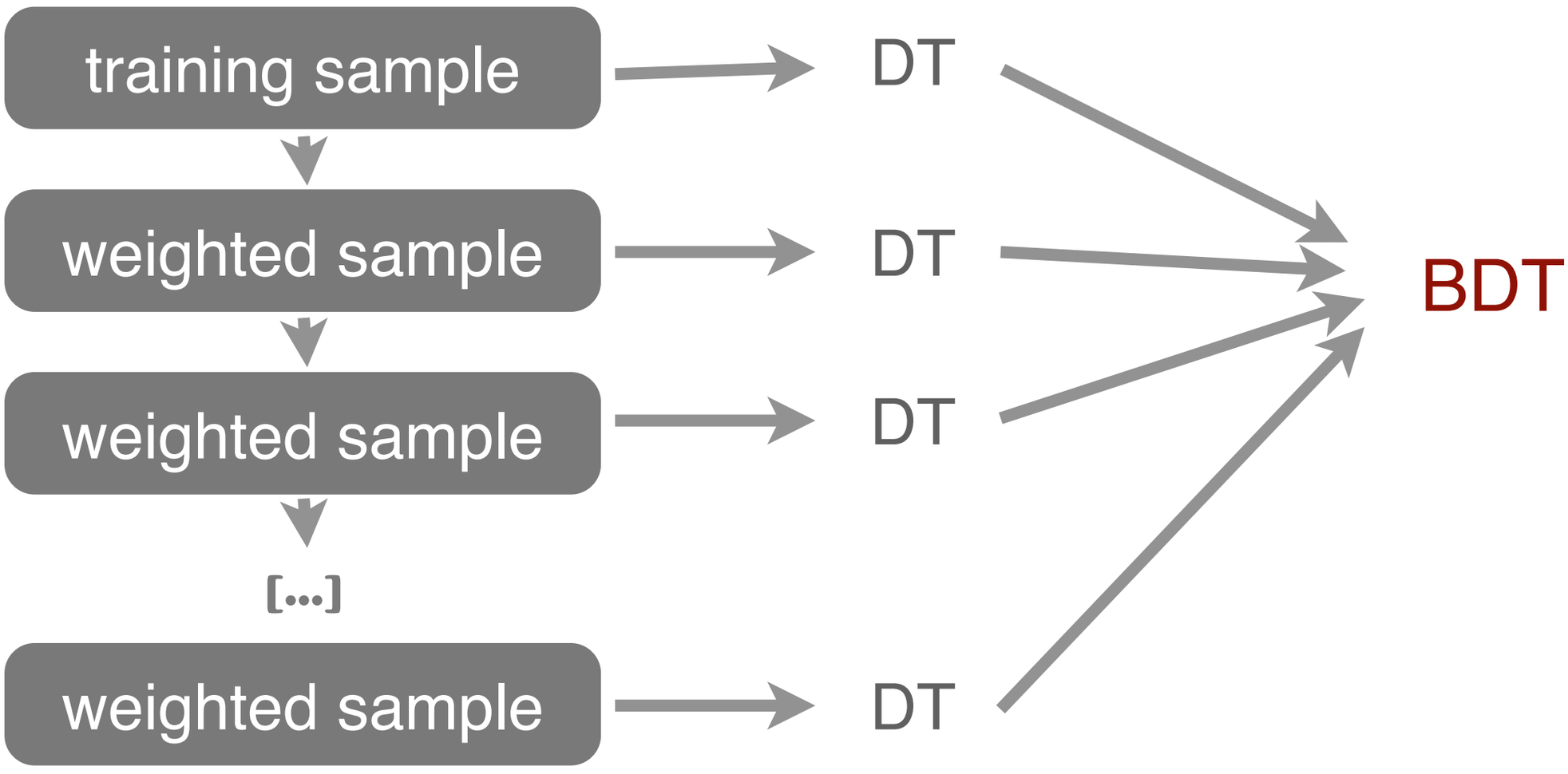


[etc]

Repeatedly split the data to maximize some FOM trying to produce pure B or S “leaves”. Stop when can’t improve FOM, or when reaching some stopping criteria (a subset of algorithm-specific hyper-parameters).

Boosting

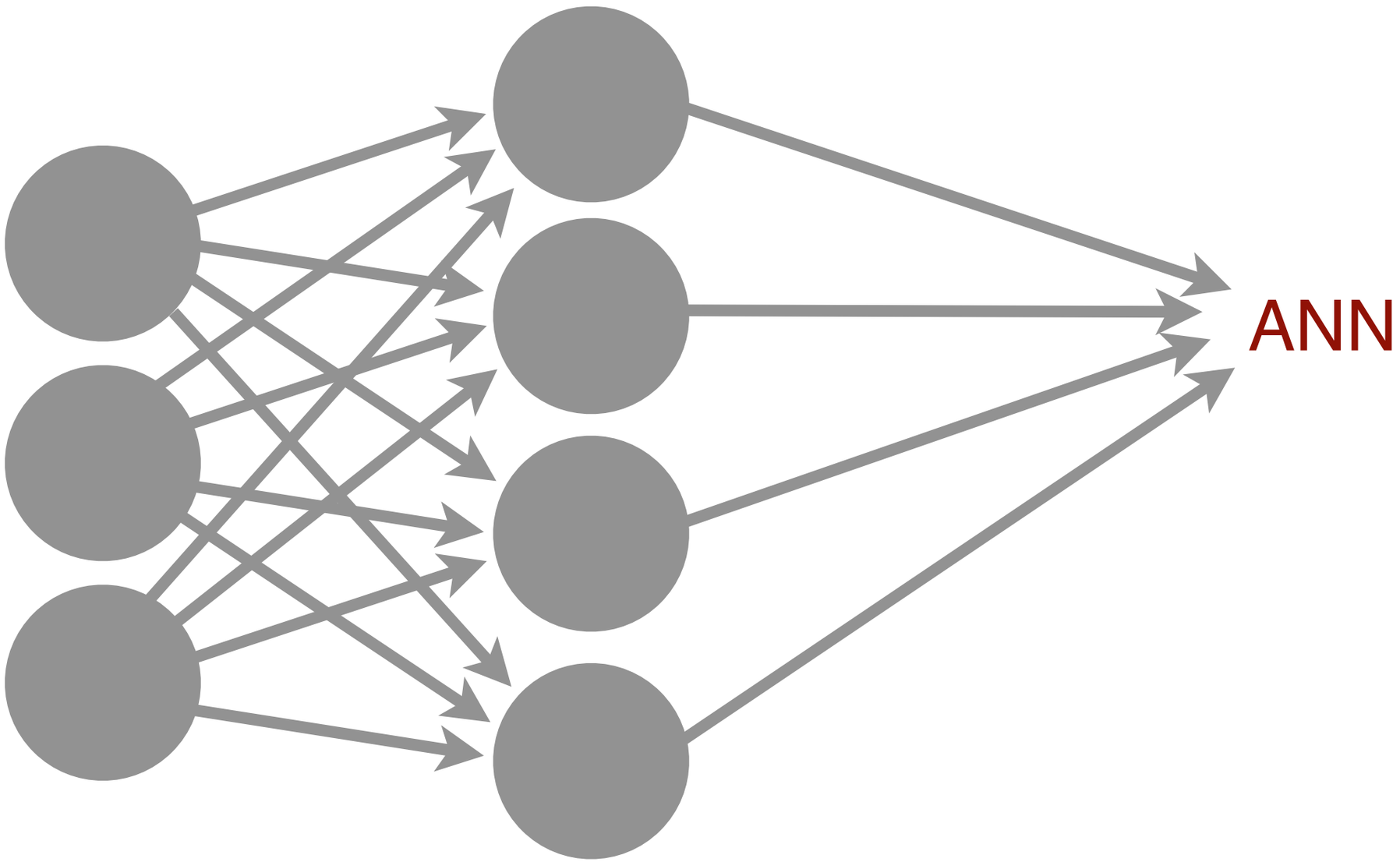
Boosting is a family of methods that produce a series of “weak” classifiers that when combined are extremely powerful.



One common approach: Each DT in the series “boosts” the weight of events based on trying to minimize some loss function.

Neural Networks

ANNs send data from input neurons via synapses to a hidden layer (or layers) of neurons, and then to the output neurons via more synapses.



Learning is typically done via back-propagation of the cost-function gradient w.r.t. the NN parameters.

Overtraining

Want to avoid learning specifics of the training data set (overtraining). Same idea as overfitting, this results in a less predictive algorithm.

It is easy to spot severe overtraining by comparing the performance on the training data set to an independent (validation) data set.

If sample sizes are limited, consider k-folding.

In short, most modern algorithms are pretty good at avoiding this provided “good” hyper-parameters are chosen (these can be problem specific however).

An enlightened way of optimizing machine learning hyper-parameters is Bayesian Optimization (e.g. the spearmint package is a very good option).

Tools

Most physicists are using TMVA in ROOT; however, the rest of the world is using the python scikit-learn package (sklearn for short) or Keras.

Basics: Adaboost DT or Multilayer Perceptron NN (MLP); State-of-the-Art: XGBoost DT or Deep NN (e.g. Tensorflow).

“

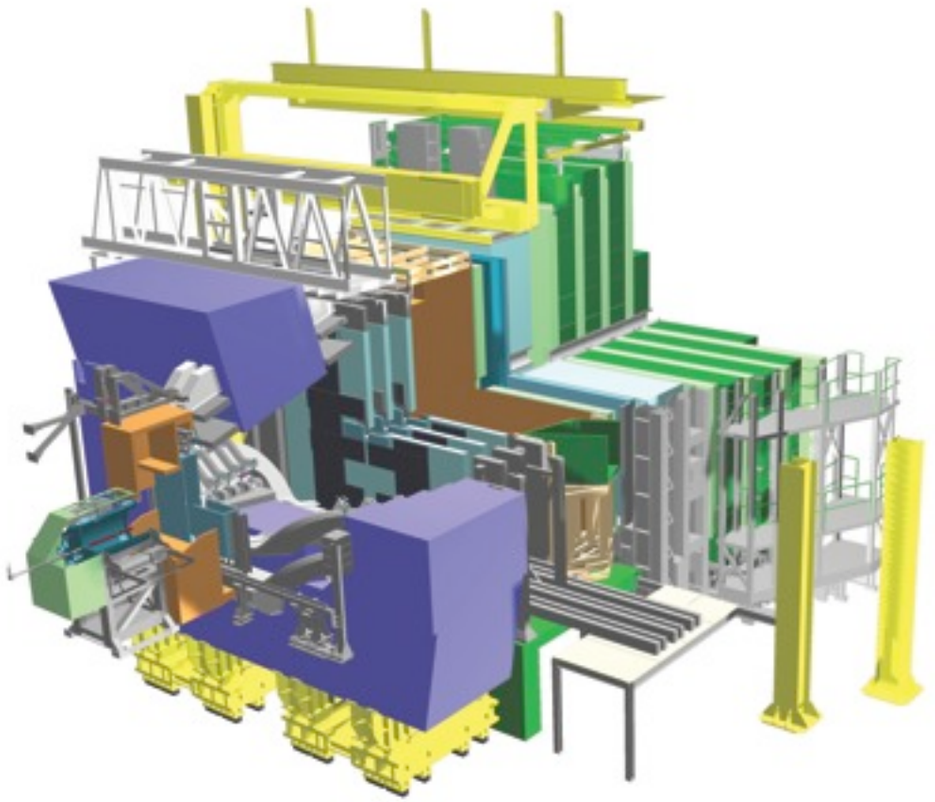
In theory there’s no difference between theory and practice. In practice there is.

Yogi Berra

”

Real-Time Processing

LHCb currently implements both hardware and software triggering stages, using the standard cascading feature-building approach.

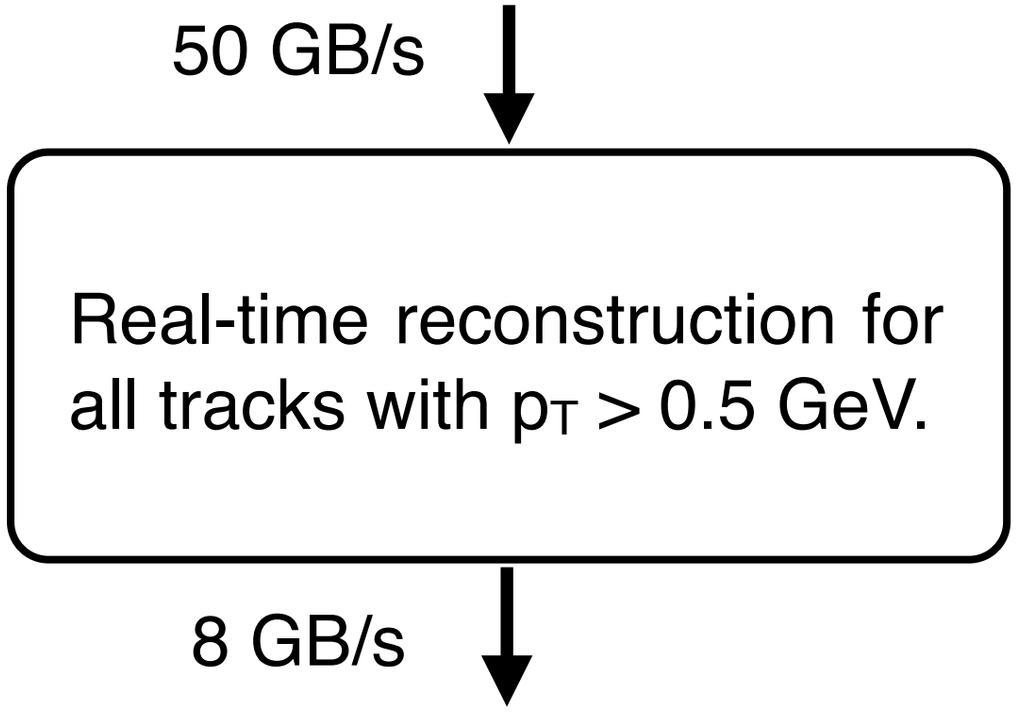


~TB/s
(post zero
suppression)

~50 GB/s

Feature-building in custom electronics (e.g. FPGAs) used to greatly reduce rate, but online computing systems still have herculean tasks.

Real-Time Processing



27,000 physical CPU cores (>50k logical cores) are used to run the full offline reconstruction in real time.

Two ML algorithms used in this stage select about 70% of the output BW.

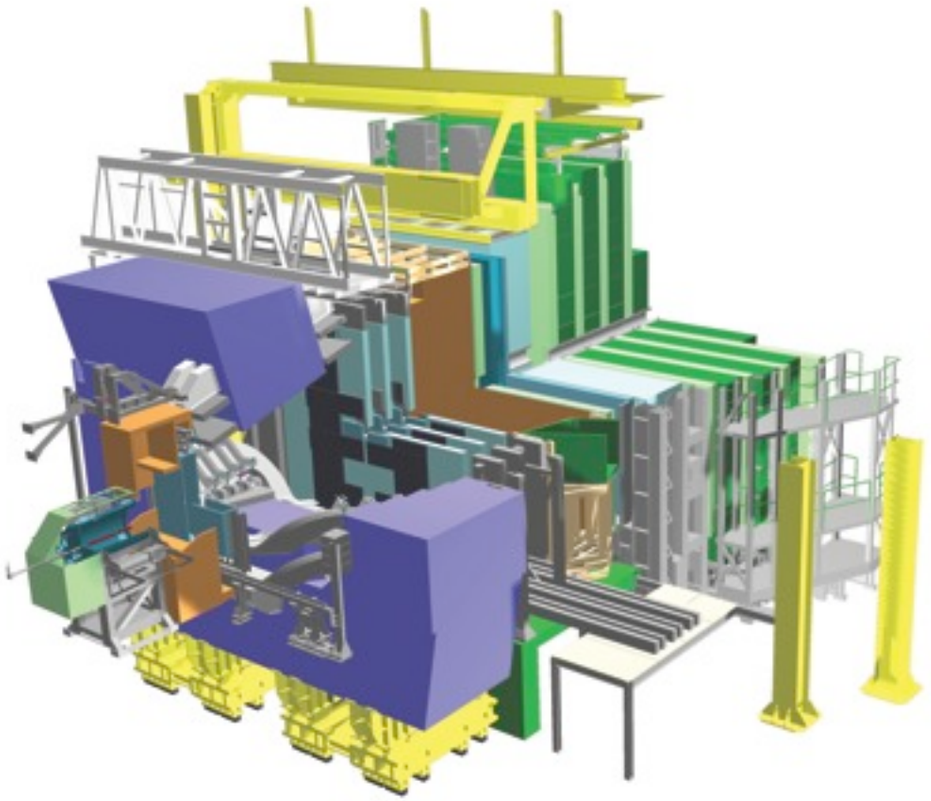
10 PB of disk used for the buffering while final alignment/calibration done.

Final event selection is currently 30% ML based (first introduced into the main trigger algorithm for the start of 2011 data taking).

See V. Gligorov, MW, JINST 8 (2012) P02013.

Real-Time Processing

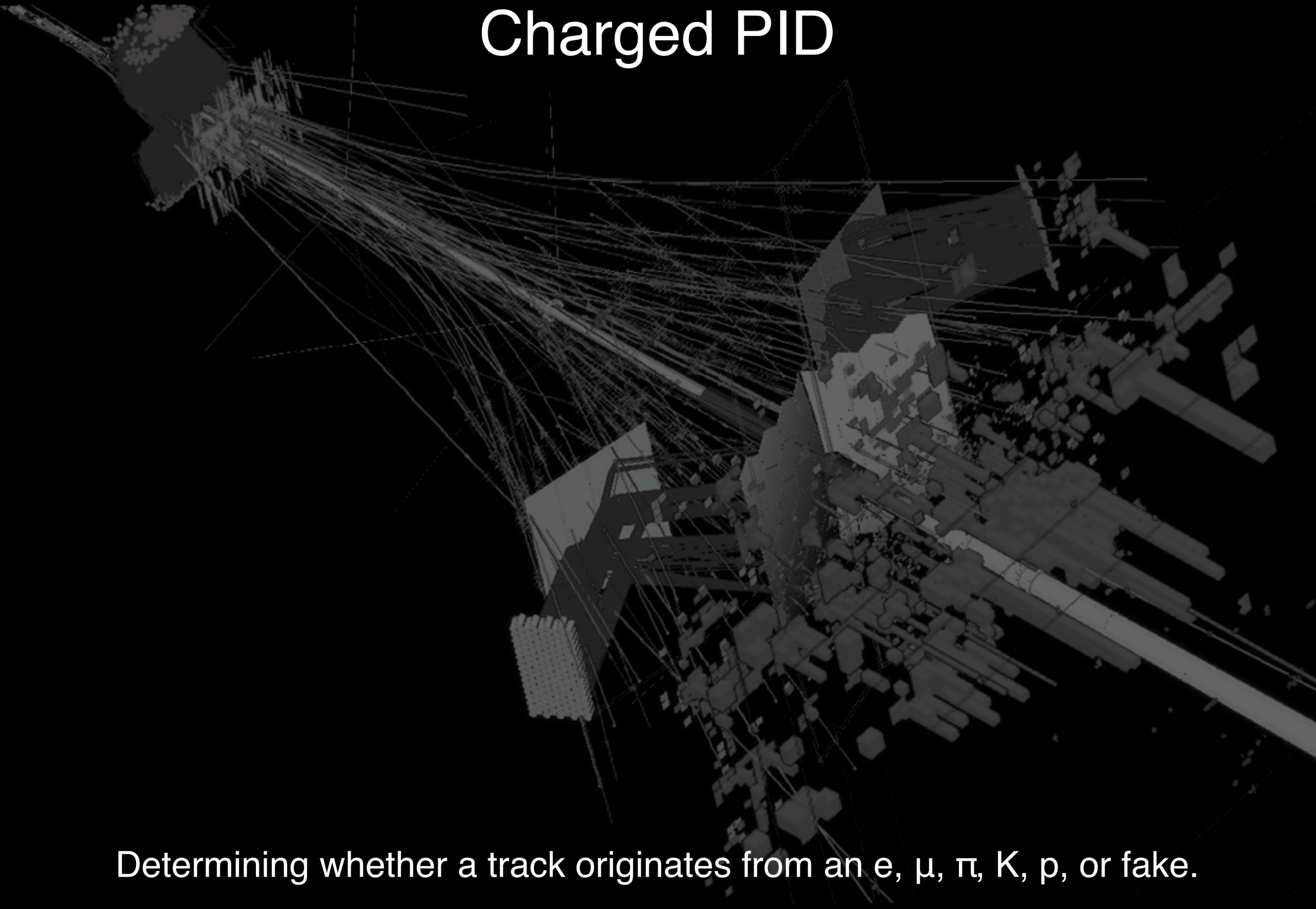
In Run 3, LHCb will increase the post-zero-suppression data rate to 5 TB/s, but also remove the hardware trigger. The full reconstruction will be done in real time on all events.



5 TB/s
(post zero
suppression)

Machine learning algorithms will play even more important roles in making it possible to maximize the physics output of this big-data challenge!

Charged PID



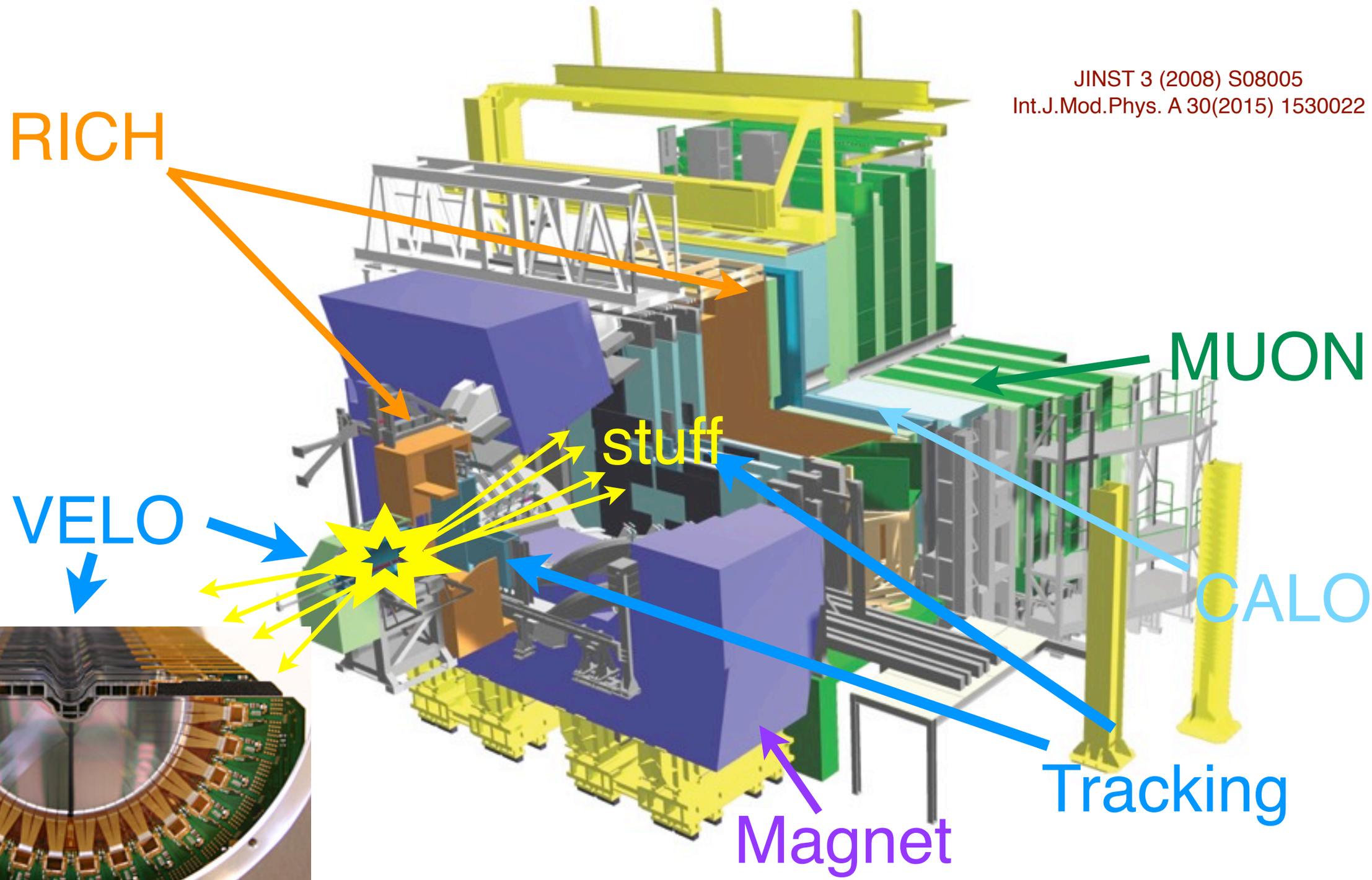
Determining whether a track originates from an e, μ, π, K, p, or fake.

LHCb Detector

LHCb is a forward Spectrometer ($2 < \eta < 5$)

(roughly 1-15^o)

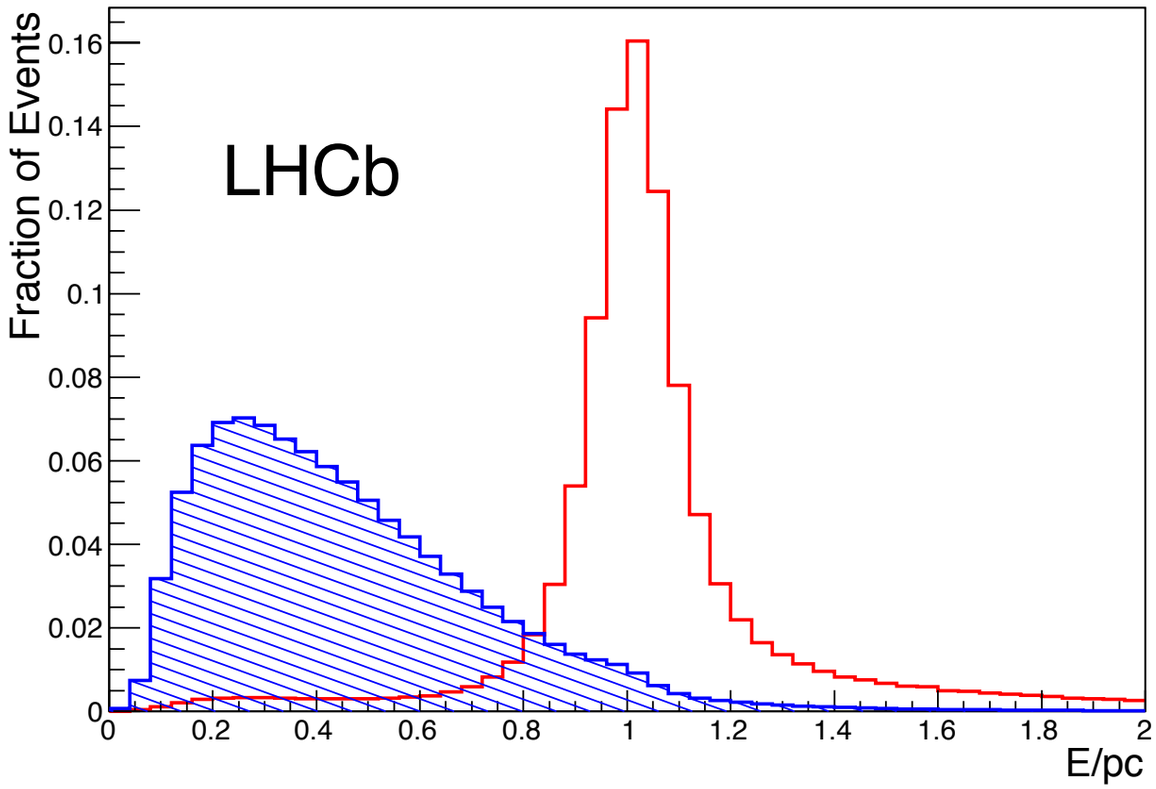
JINST 3 (2008) S08005
Int.J.Mod.Phys. A 30(2015) 1530022

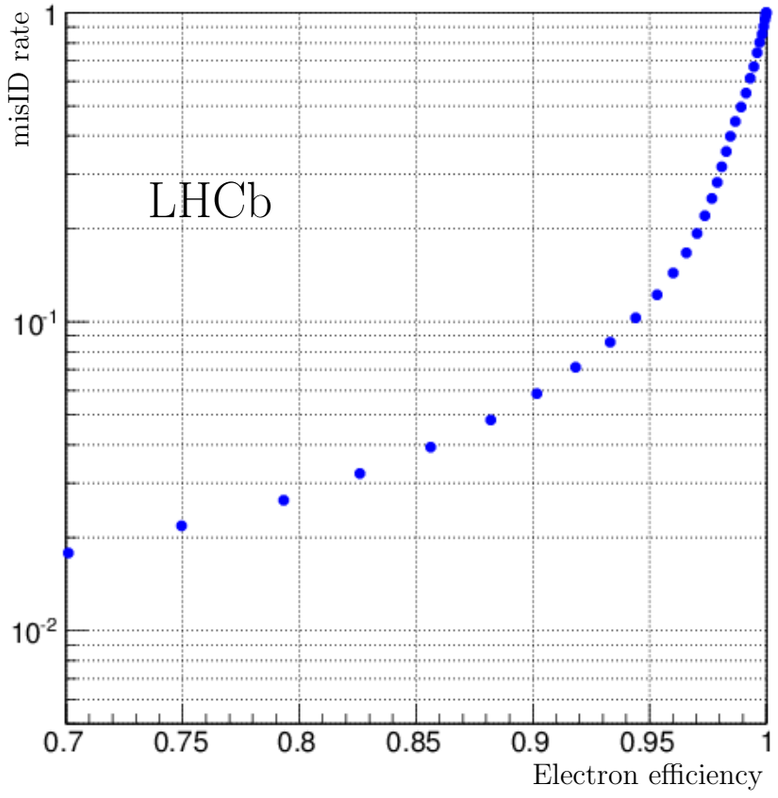


Calorimeters

The primary use of the calorimeters for charged PID is in identifying electrons.

$$\Delta\log\mathcal{L}^{\mathsf{CALO}}(e-h) = \Delta\log\mathcal{L}^{\mathsf{ECAL}}(e-h) + \Delta\log\mathcal{L}^{\mathsf{HCAL}}(e-h) + \Delta\log\mathcal{L}^{\mathsf{PS}}(e-h)$$

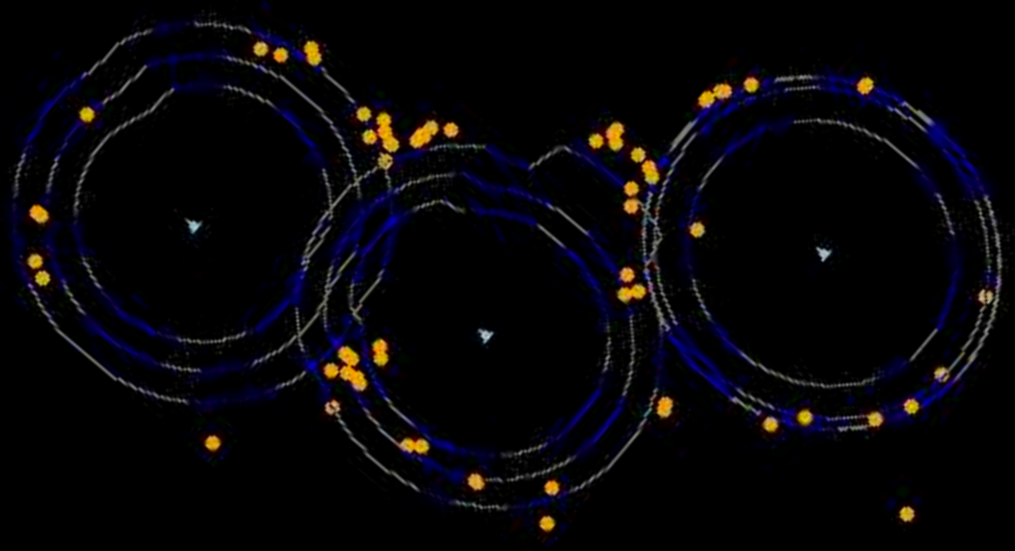


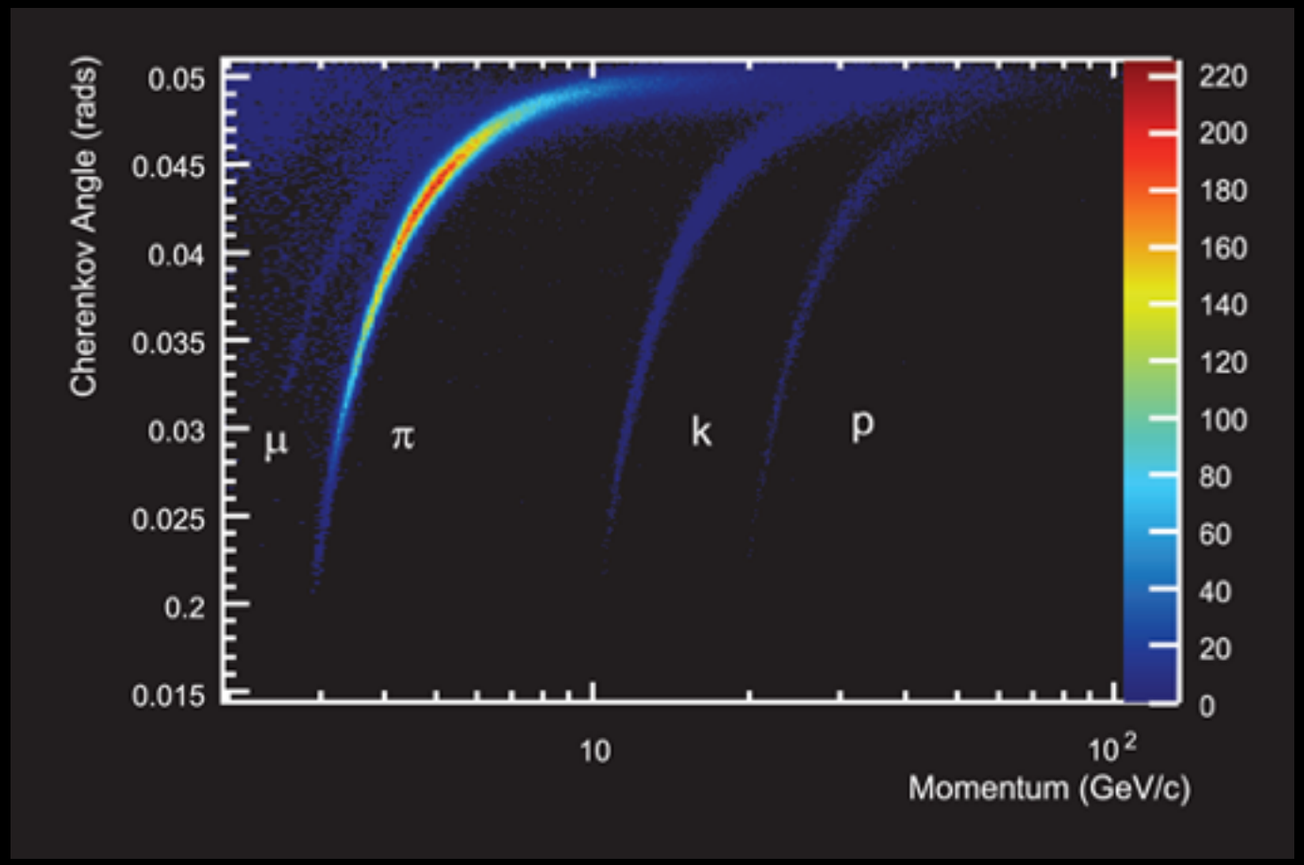


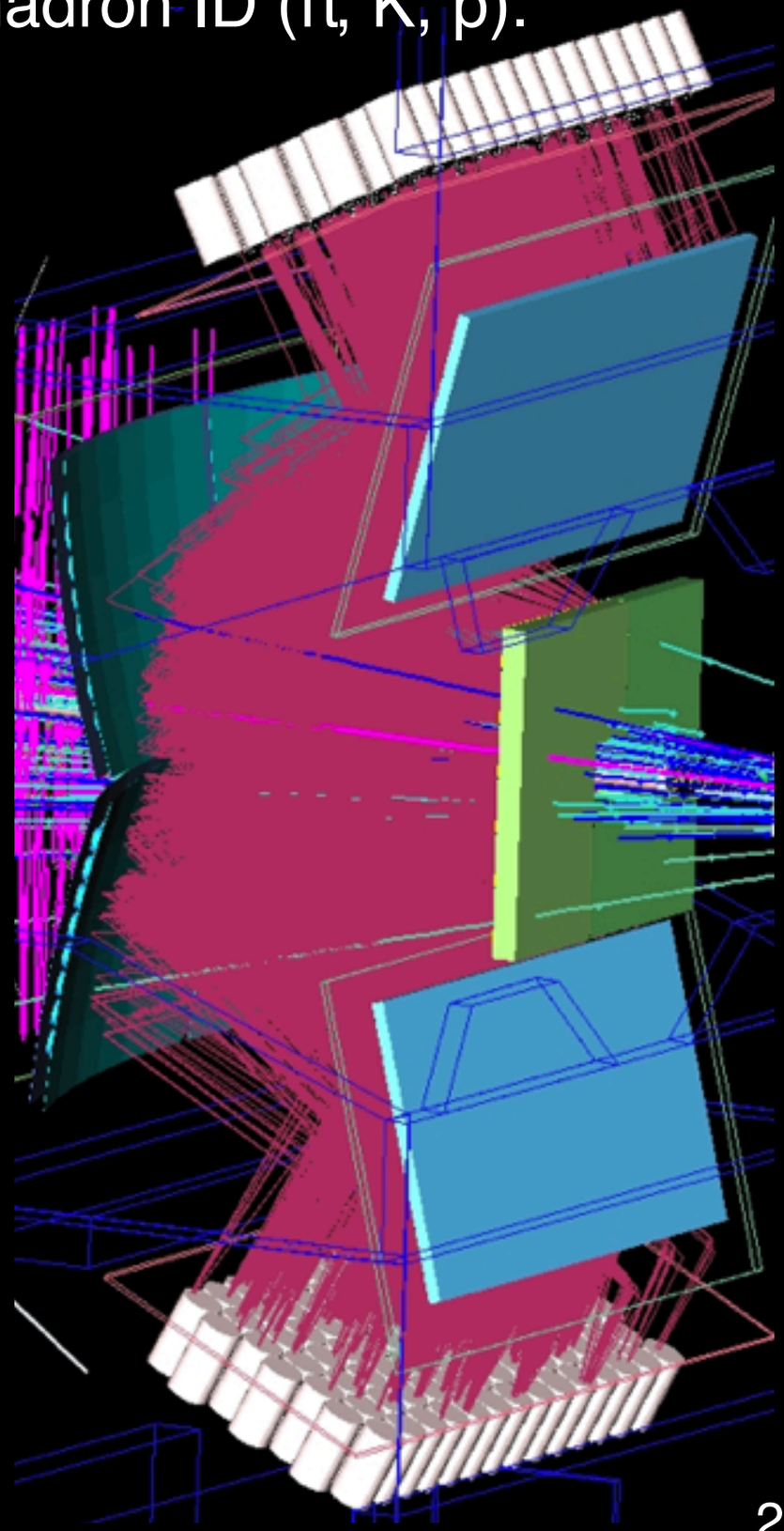
Using electrons from photon conversions and hadrons from D^0 decays, e and h PDFs are constructed from data vs track 3 momentum.

RICHs

The primary role of the RICHs is charged-hadron ID (π, K, p).

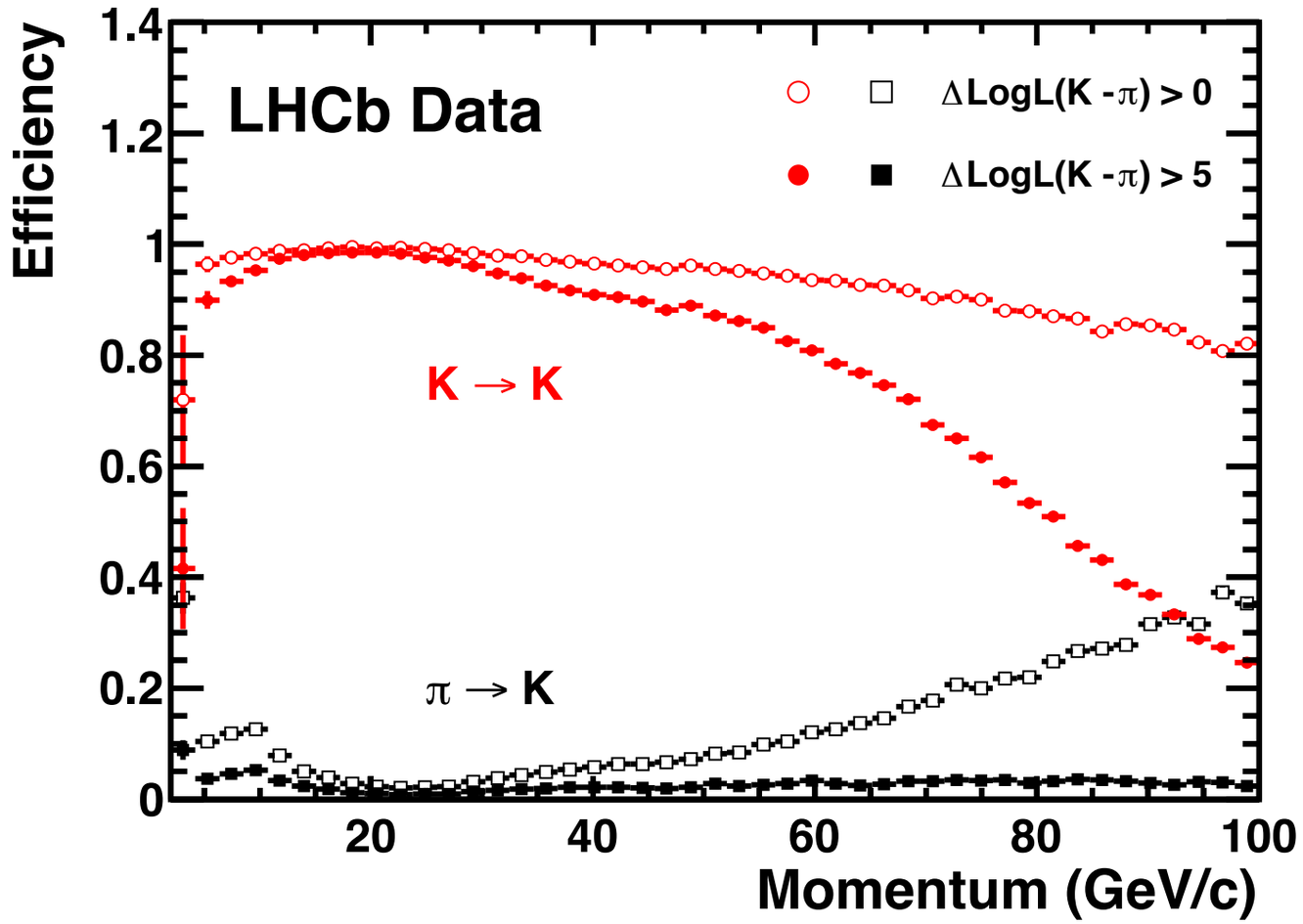






RICHs

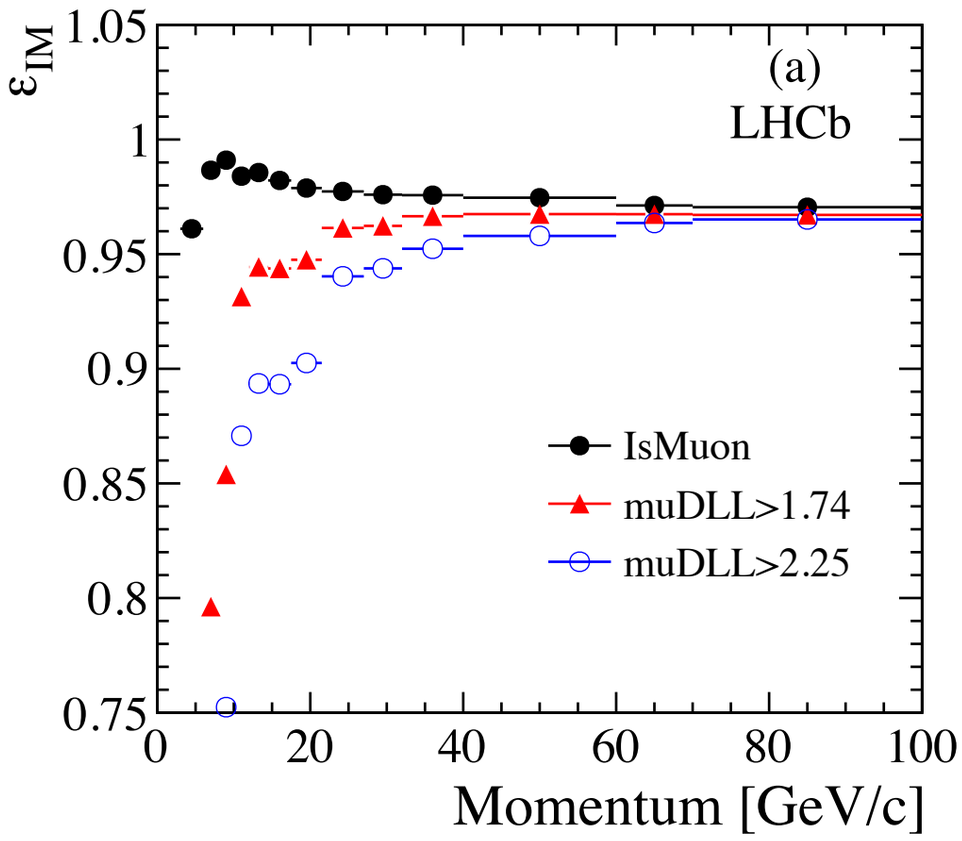
The primary role of the RICHs is charged-hadron ID (π, K, p).

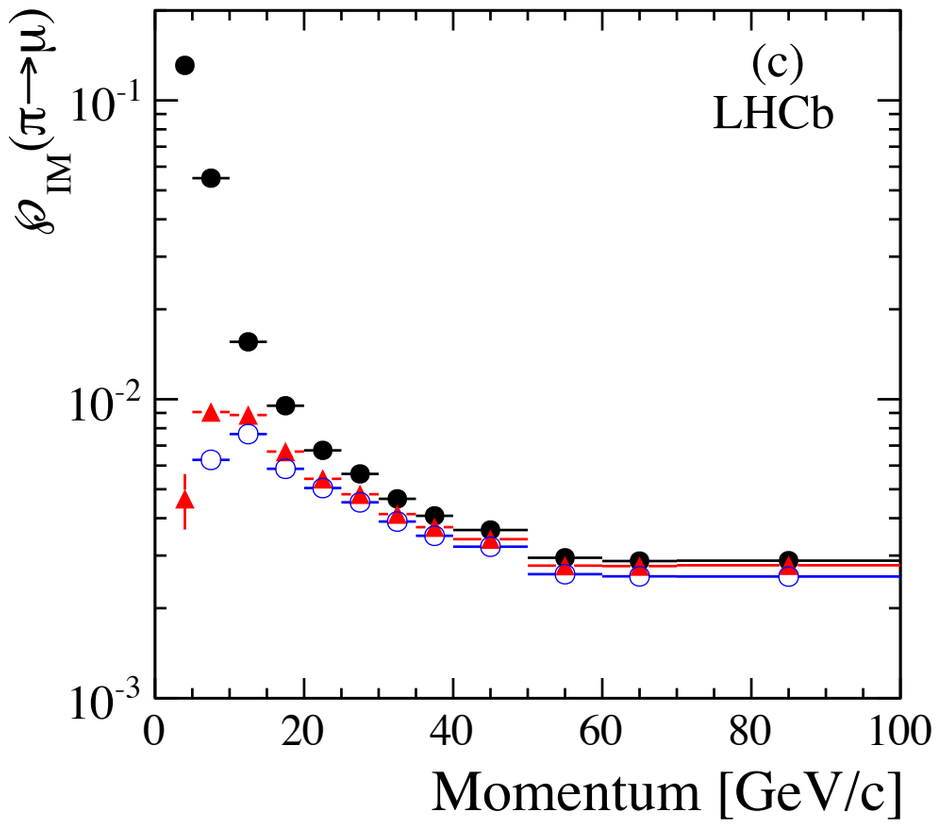


Calculate the likelihood of each RICH ring pattern observed under various PID hypotheses, then use "DLL" to arbitrate (calibrate/validate using $K_S \rightarrow \pi\pi$, $\Lambda \rightarrow p\pi$, and $D^0 \rightarrow K\pi$ data samples).

Muon System

Muons are identified by looking for hits in the muon system, which is shielded by both the ECAL, HCAL, and whose stations are interleaved with iron absorbers.

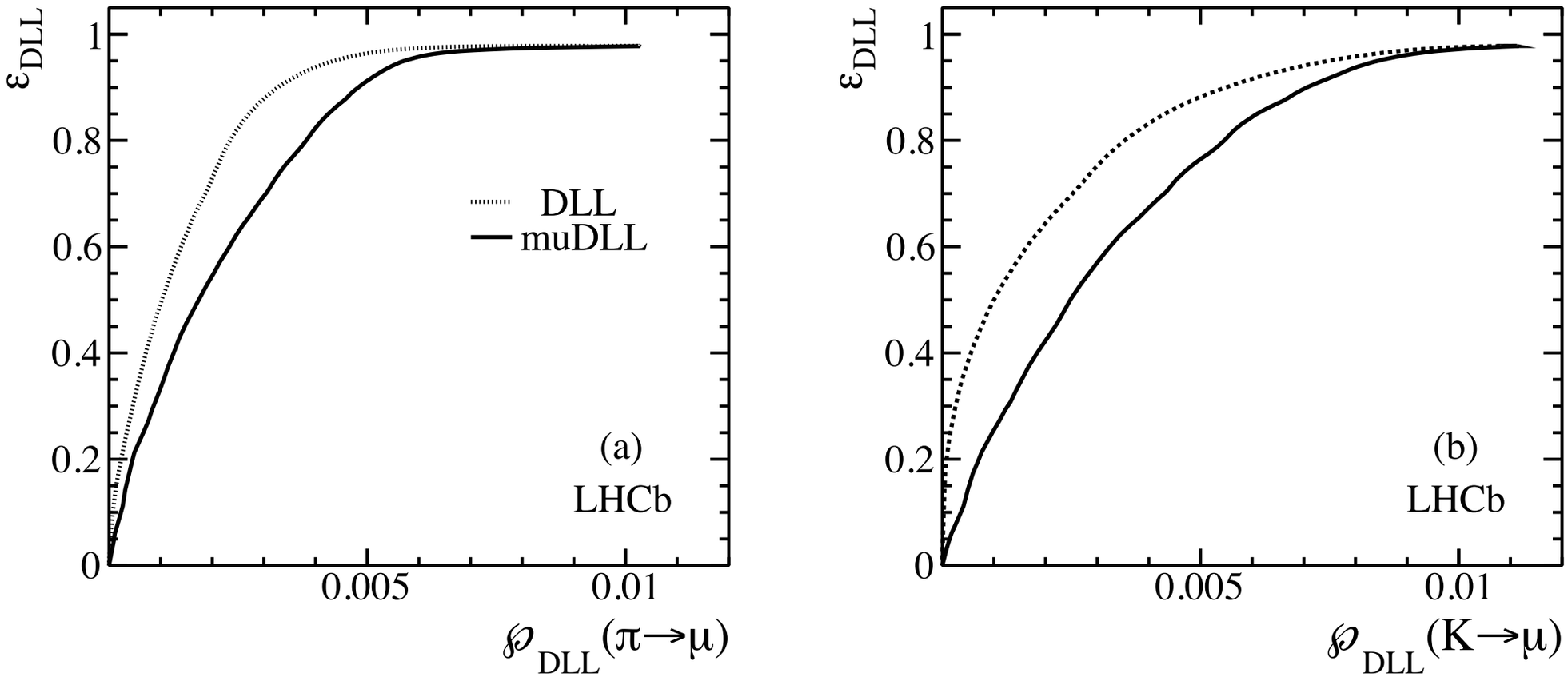




MisID from π,K→μ in flight, shared hits with a real muon, punch through, etc.

Combined DLLs

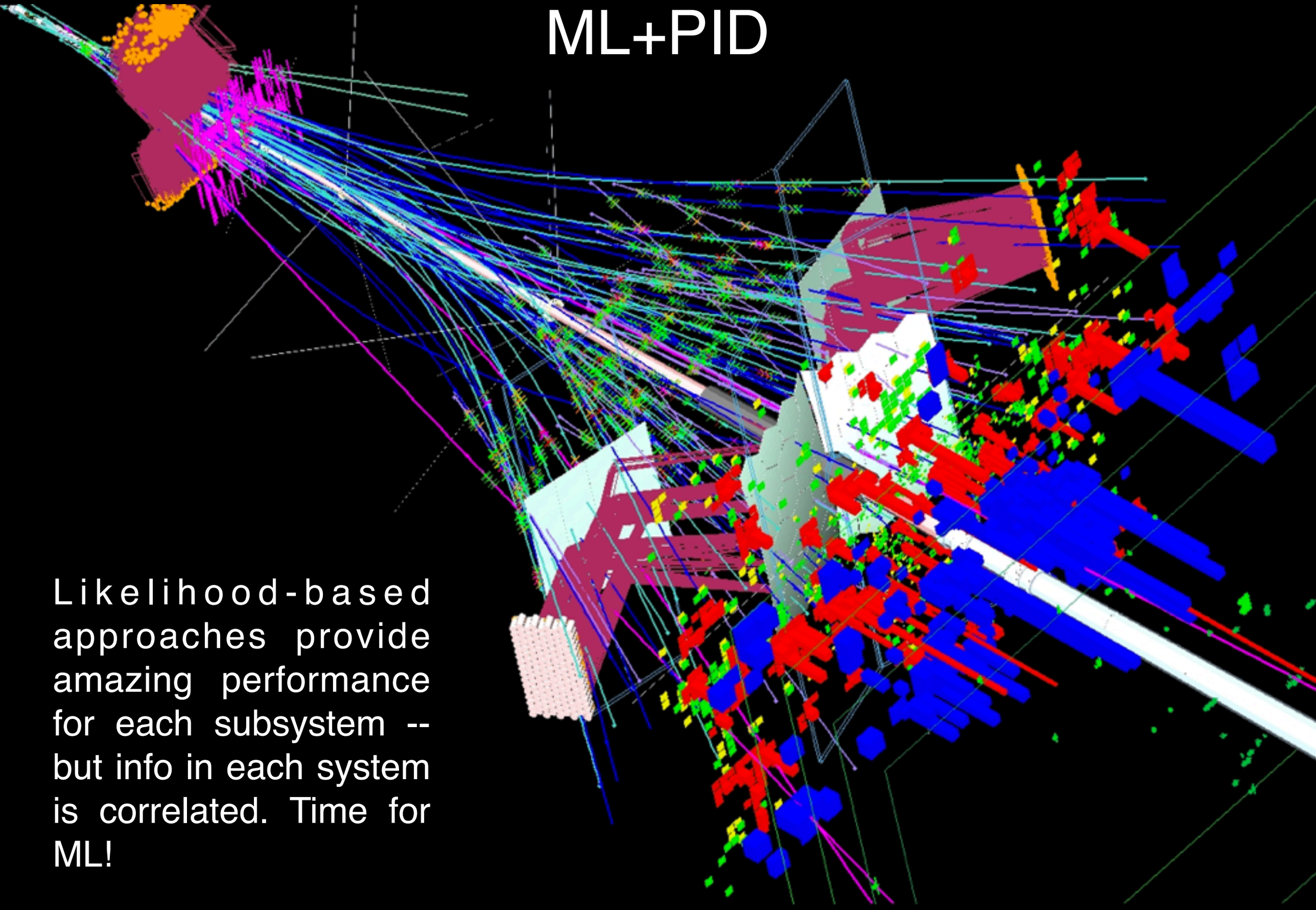
By combining the likelihoods from the RICHs, calorimeter system, and the muon system, LHCb obtains even better PID performance.



Consider the common case of K→μ decay in flight. If it was still a kaon when it passed through the RICH, then the RICH likelihood will show this.

E.g., CombDLL reduces B→hh misID by 6x for loss of 3% of B_s→μμ signal.

ML+PID



Likelihood-based approaches provide amazing performance for each subsystem -- but info in each system is correlated. Time for ML!

PID NNs

Drawbacks of CombDLL:

- Hard to build MVA PDFs from the data only. Could use MC -- but then not all subsystems may be “scaled” properly.

- Discrete info doesn’t fit well (or at all).

- DLL is of no use when neither hypothesis is correct.

E.g., if the track is below Cherenkov threshold for both a pion and kaon, then their DLL = 0 by construction -- but what if there’s a nice ring in the data?

Advantages of an ML-based approach:

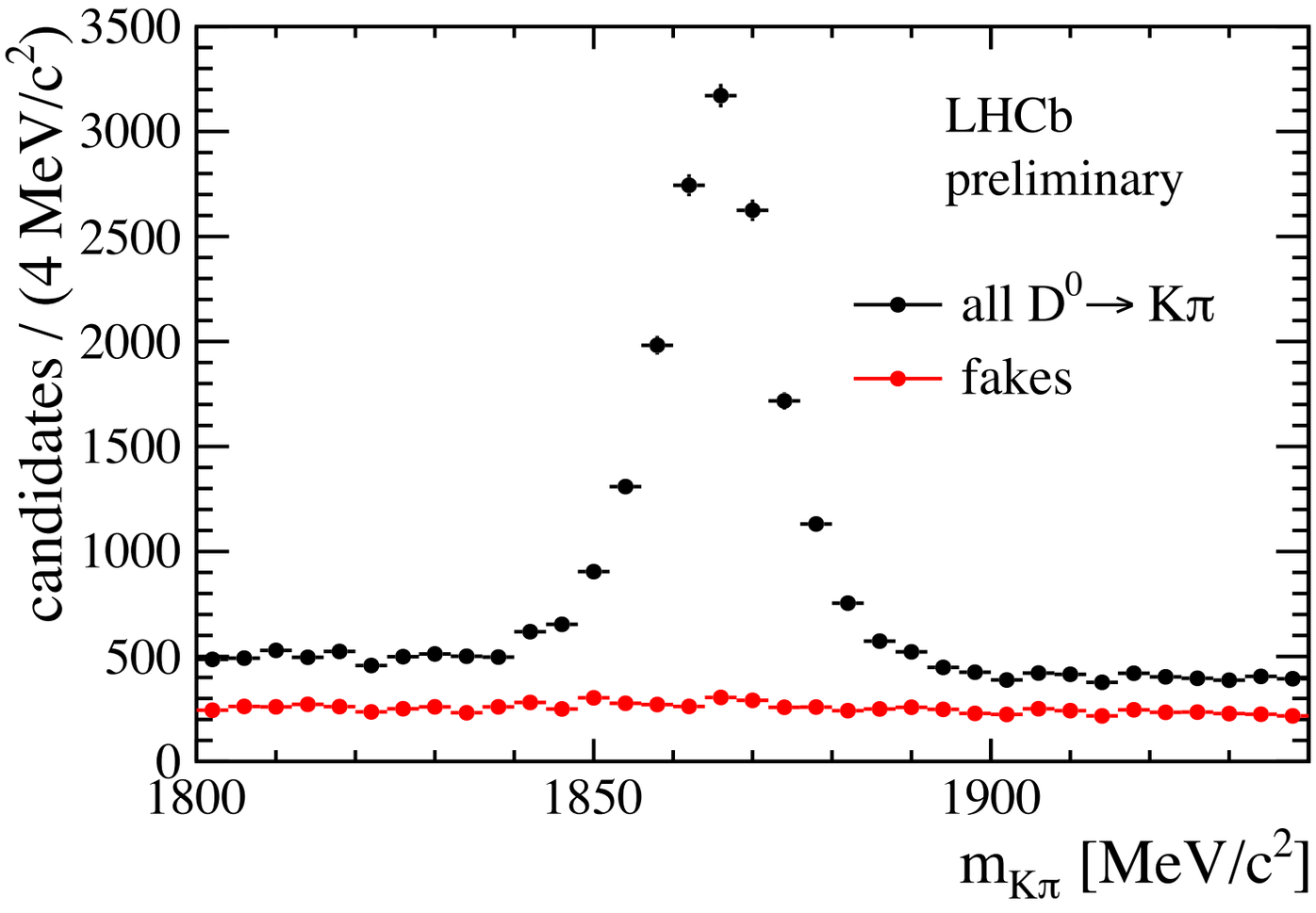
- Trivial to use discrete and continuous features.

- Train one algorithm for each track type where anything else is BKGD.

- Dimensional reduction makes validation/calibration in data much easier.

Fake Tracks

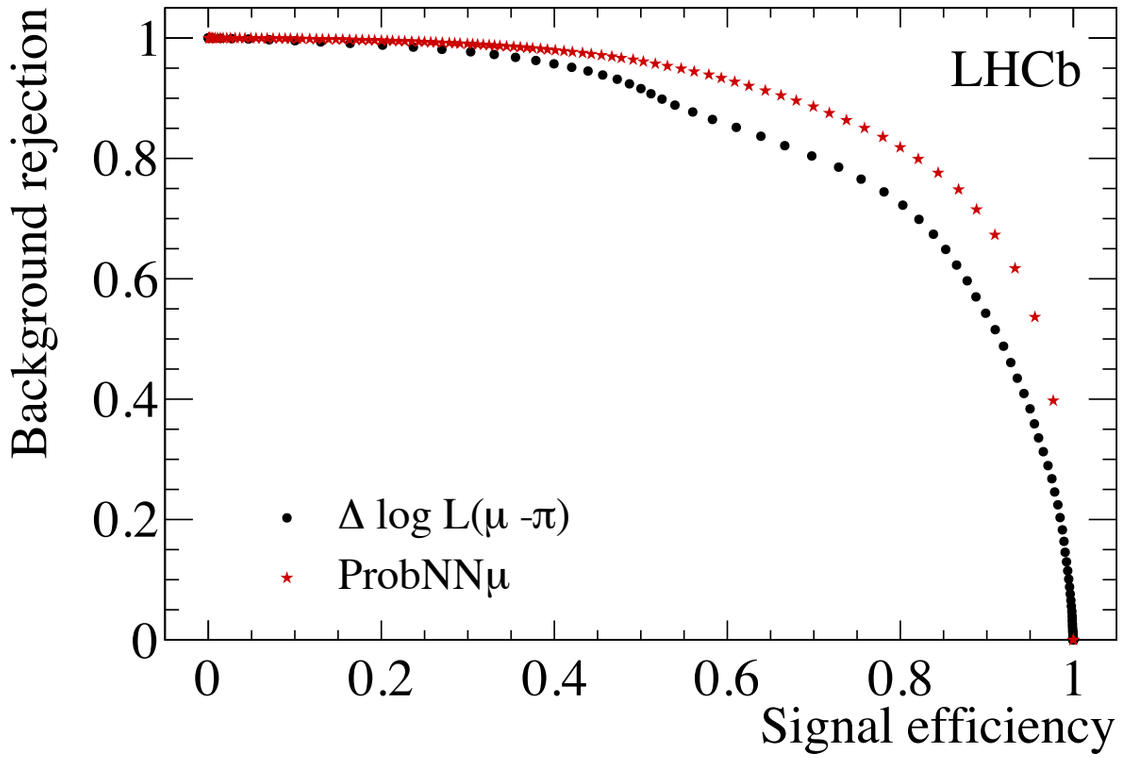
Tracks are built using a Kalman-filter algorithm and selected based on several quantities, one of which is the x^2/n_{DOF}.

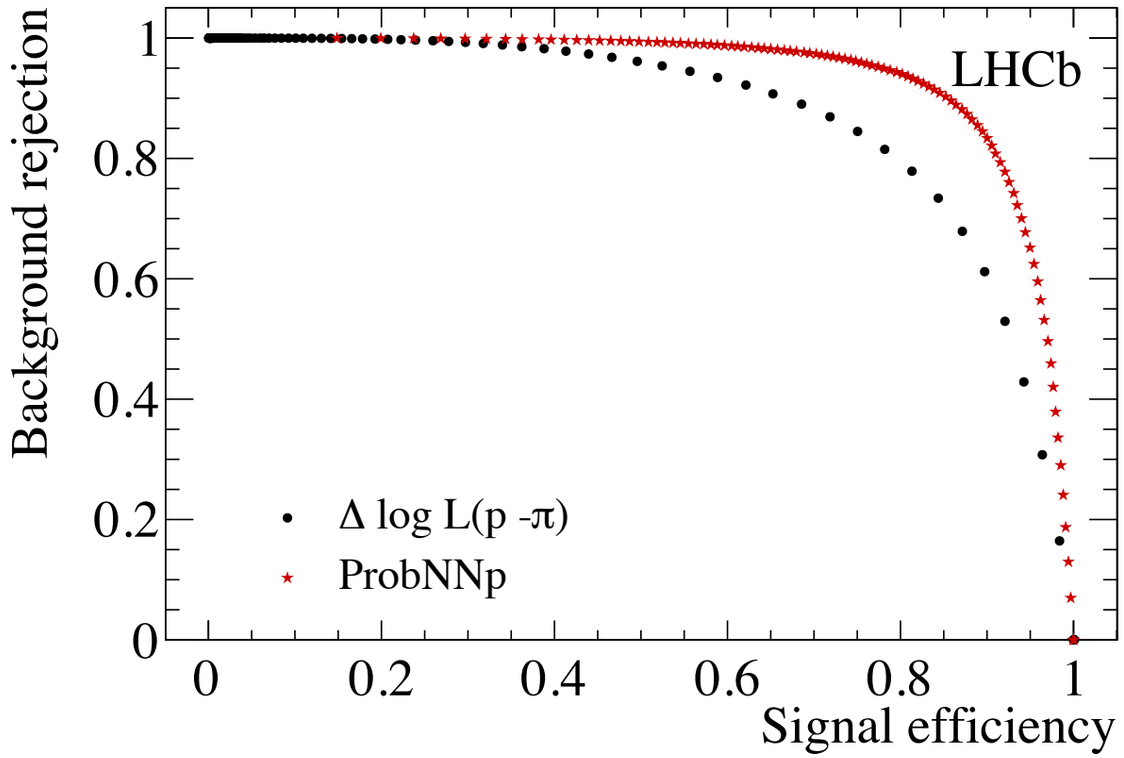


Training a NN to also consider missing hits, track-segment matching x^2, detector occupancy, etc, we are able to greatly reduce the fake-track rate.

PID NNs

Single-hidden-layer MLP NN trained on 32 features from all subsystems. Each is trained to identify a specific type of particle (or fake track).





Typically get a factor of 2-3x less pion contamination in a muon sample than using the CombDLL approach -- ~5-10x less in a dimuon sample!

See: Ilten, Soreq, Thaler, MW, Xue, Proposed inclusive dark photon search at LHCb, PRL 116 (2016) 251803.

Currently exploring state-of-the-art: XGBoost ~ Deep NN ~ 50% less BKGD than basic BDT or ANN, which again give 2-3x less BKGD than DLLs.

Details

•We train our charged-particle PID NN's on MC then measure their performance using data control samples. In principle, data samples could also be used in the training, but then one would need to deal with BKGD in those samples (and wait for data to be taken to do the training).

•However ML algorithms are trained, it's vital that they be validated using a data-driven approach!

•There is an "art" to how to weight the various BKGD track types in the training. E.g., fake tracks are a major concern when studying electrons, but much less so for other track types. One could consider using a bespoke NN training for each measurement; however, a much simpler approach is to use the charged-PID NN's as input features themselves in a channel-specific ML-selection algorithm (stacking).

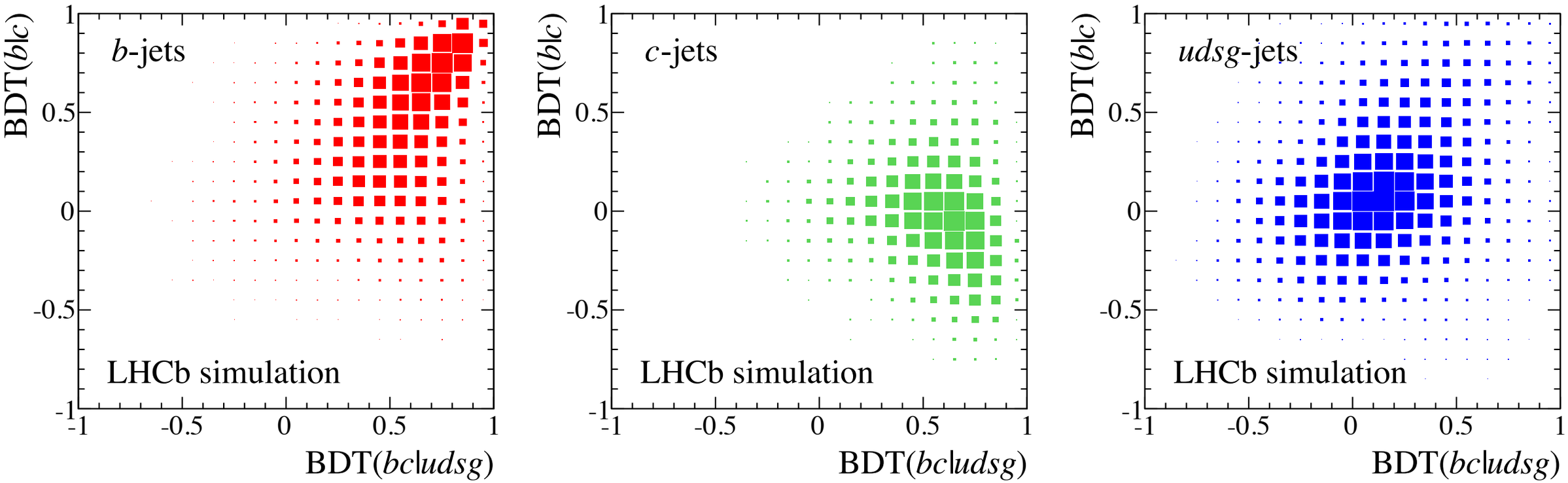
Heavy-Flavor Jet ID (Tagging)

ML Jet Tagging

JINST 10 (2015) P06013
LHCb-PAPER-2015-016

Put 10 features into two BDTs: one for b,c vs light, and another for b vs c. No feature can fully separate types, but their correlations (largely) can.

LHCb simulation: each distribution normalized to one; 70%, 25%, 1% of b, c, light jets are tagged.



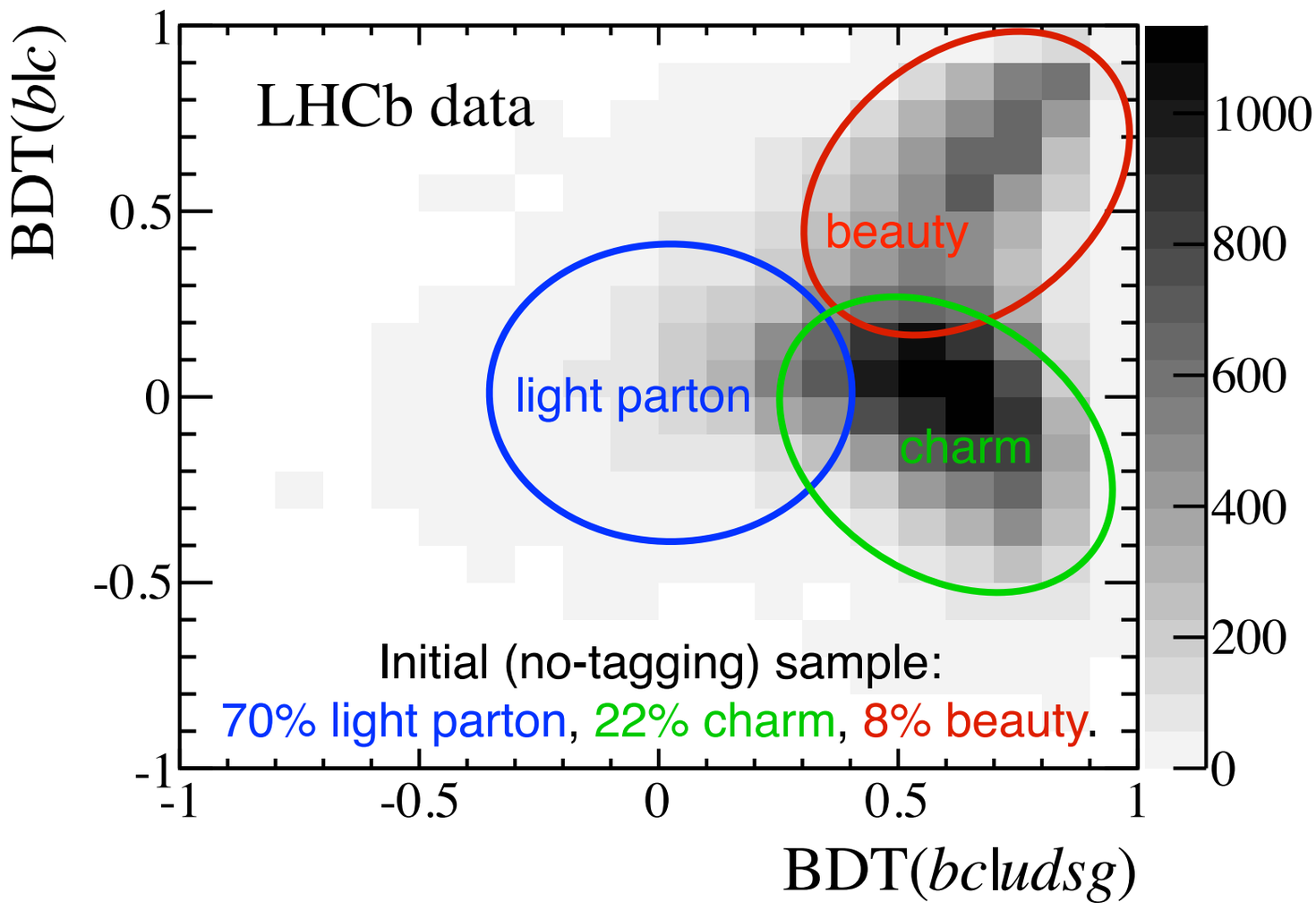
Could cut on BDT responses to obtain high-purity b-jet or c-jet samples. Alternatively, fit 2-D BDT distribution to extract the b-jet and c-jet yields.

Looked at doing a single 3-class algorithm but that doesn't seem to help here (shown to work better in other applications).

2-D BDT Fits

JINST 10 (2015) P06013
LHCb-PAPER-2015-016

2-D BDT plane (nearly) optimally utilizes 10-D info to ID b, c, and light jets.



Performance validated & calibrated using large heavy-flavor-enriched jet data samples (2-D data validation much easier than 10-D!).

Details

•We train our jet-tagging BDTs on MC then calibrate and measure their performance using data control samples.

•To reiterate: it's vital that ML algorithms be validated in a data-driven manner! (What this means varies for different applications of course.)

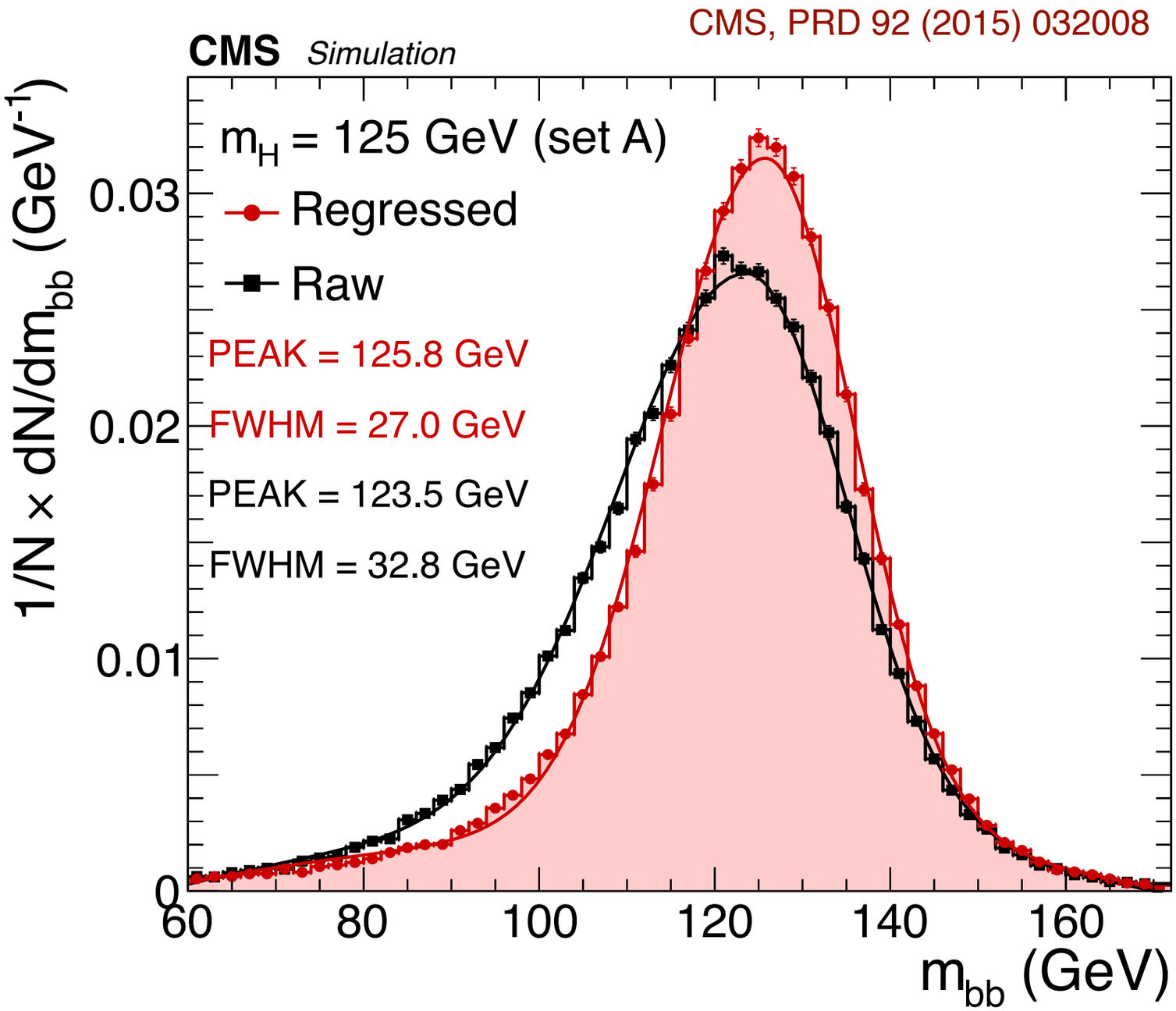
•Dimensional reduction achieved by such algorithms makes it possible to maximize performance without complicating validation.

•This work was done using Adaboost, currently studying state-of-the-art algorithms (re-examining the 3-class issue).

N.b., dimensional reduction can be exploited to build GOF tests for large dimensionality (Weisser, MW, to appear soon).

Regression

ML is also now being used for regression. For example, CMS has moved to ML-based jet-energy corrections (e.g., for Higgs to bb).



Advertisement

ML schools for Ph.D. students in Lund 2016, St. Petersburg 2015.

Machine Learning High Energy Physics Summer School 2016

20-26 June 2016
Lund University
Europe/Zurich timezone

- Overview
- Scientific Programme
- My Conference
  - My Sessions
- Important dates
- Speakers
- Registration fee
- Registration
- Frequently asked questions
- Venue
- MLHEP2015 feedback
- Committees

The Machine Learning (ML) summer school is intended to cover the relatively young area of data analysis and computational research that has started to emerge in High Energy Physics (HEP). It is known by several names including "Multivariate Analysis", "Neural Networks", "Classification/Clusterization techniques". In more generic terms, these techniques belong to the field of "Machine Learning", which is an area that is based on research performed in Statistics and has received a lot of attention from the Data Science community.

There are plenty of essential problems in High energy Physics that can be solved using Machine Learning methods. These vary from online data filtering and reconstruction to offline data analysis.

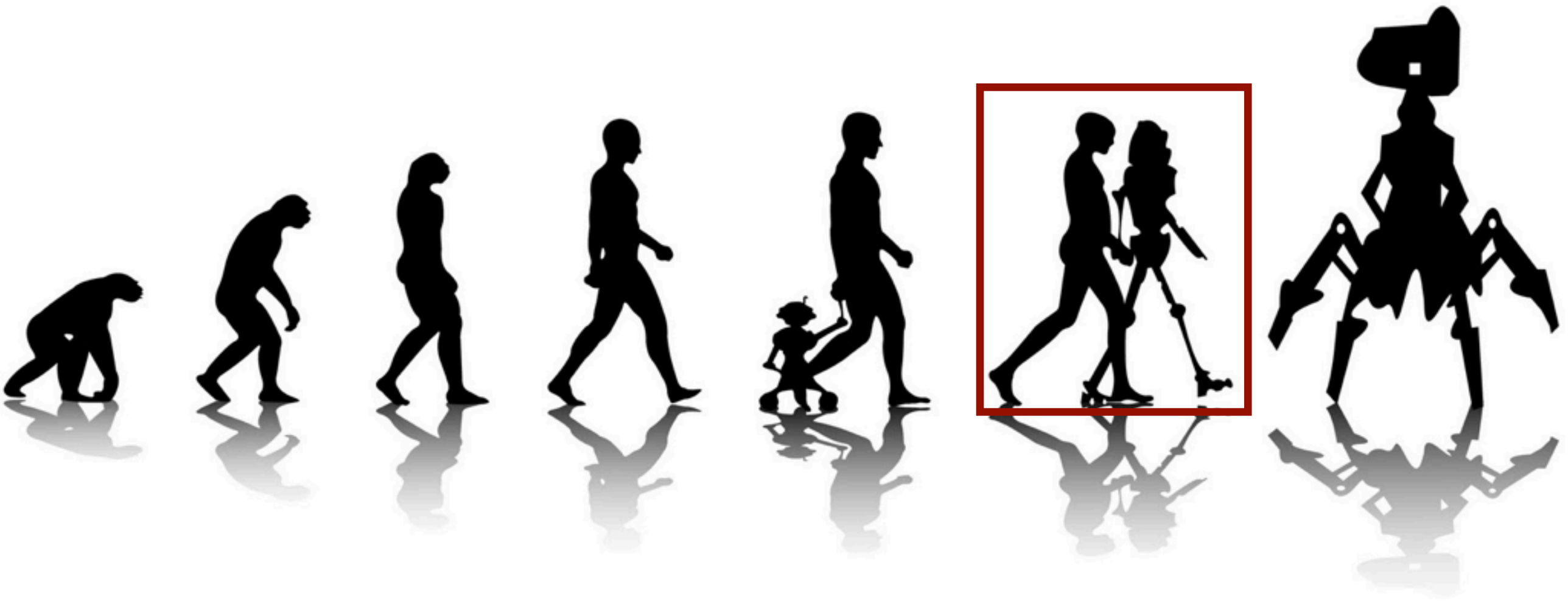
Students of the school will receive a theoretical and practical introduction to this new field and will be able to apply acquired knowledge to solve their own problems. Topics ranging from decision trees to deep larning and hyperparameter optimization will be covered with concrete examples and hands-on tutorials. A special data-science competition will be organized within the school to allow participants to get better feeling of real-life ML applications scenarios.

The MLHEP school is a satellite event to the LHCP2016 conference, so its dates and venue (Lund University) are well-aligned with the conference.

Expected number of students for the school is 40-50 people.

Great hands-on experience, don't be deterred by the HEP in the name!

The (Near) Future



Current Work & Prospects

•Custom loss functions, e.g., response is de-correlated from some set of features (Stevens, MW [1305.7248]; Rogozhnikova, Bukva, Gligorov, Ustyuzhanin, MW [1410.4140]). Already used to search for a new boson (LHCb, PRL 115 (2015) 161802), and currently being used in many papers to appear soon.

•Deep learning is actively being explored as a way to remove the human-led feature-design stage (Baldi, Sadowski, Whiteson [1402.4735]).

•Parametrized classifiers to interpolate the optimal selection when only a discrete subset of possible signals has been simulated (Baldi, Cranmer, Faucett, Sadowski, Whiteson [1601.077913]).

•Autoencoders, ML-based tracking, anomaly detection, domain adaptation, MVA re-weighting, etc.

•N.b., beware of non-general optimizations in some purpose-built algorithms!

See also pypi.python.org/pypi/hep_ml/0.2.0 for some custom HEP algorithms.

Summary

•Machine learning algorithms are now commonplace in HEP analyses. Open-source tools are now very good and getting better daily.

•ML algorithms exploit high-dimensional correlations to improve on cut-based selections (can also view them as dimensional-reduction methods). Even basic algorithms tend to give big improvements, and state-of-the-art algorithms are now easy for novices to use.

•It's vital that such algorithms can be validated/calibrated in a data-driven approach---and not just because your more senior colleagues don't like them!

•There are many great and simple data-analysis techniques out there that are almost unknown to physics (Bayesian optimization, BIC, LASSO, etc). Don't be afraid to use them!

•The potential physics applications of many other cutting-edge ideas are now being studied (the future is now!).

The Future?