Machine Learning in Particle Physics

Mike Williams
MIT

August 29, 2018

Largely inspired by a Nature review article written in collaboration with A Radovic, D Rousseau, M Kagan, D Bonacorsi, A Himmel, A Aurisano, K Terao, & T Wongjirad.

Who am I?

- I am an Associate Professor at MIT and I work on the LHCb experiment. Prior to joining the faculty at MIT in 2012, I was a postdoc at Imperial College London 2008-2012, a grad student at Carnegie Mellon, …
- I am NOT a dedicated ML specialist. I am a physicist. I write a lot of phenomenology, LHCb, and technical papers.
- But I did write the primary LHCb b-physics trigger algorithm — and for the start of Run 1 this was already using an ML algorithm! (With a custom loss definition.) This algorithm, including its Run 2 update, has collected the data used in ~300 papers to date.
- I also co-led (with Phil Ilten and Yandex colleagues) the proliferation of ML usage throughout the LHCb trigger in Run 2. I have used ML in many LHCb publications, etc.
- I co-wrote the first (to my knowledge) HEP paper that presented a dedicated HEP-specific loss function.
- I approach ML like everything else in physics: I try and understand the concepts, develop/use intuition, etc. The main goal of these lectures is to focus on the core concepts of ML and its usage in HEP. Alex has written an excellent tutorial to help you start to learn the technicalities.

The Short-Short Version

We use ML almost everywhere—to enable great science!

Out of an abundance of laziness, I've reused a lot of slides from previous talks. The end result is LHCb-heavy on examples. For the most part, all experiments are using ML in similar ways.

Overview

According to wikipedia: Machine learning is a field of computer science that gives computers the ability to *learn* (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed.

It is, in fact, a very broad field of study—and one whose use in particle physics is rapidly becoming ubiquitous.

Particle physics experiments produce some of the largest data sets in the world: O(PB/s) → zero suppression → O(10 TB/s) → event filtering → O(GB/s) permanent storage (multiply these by 10 for the HL-LHC era).

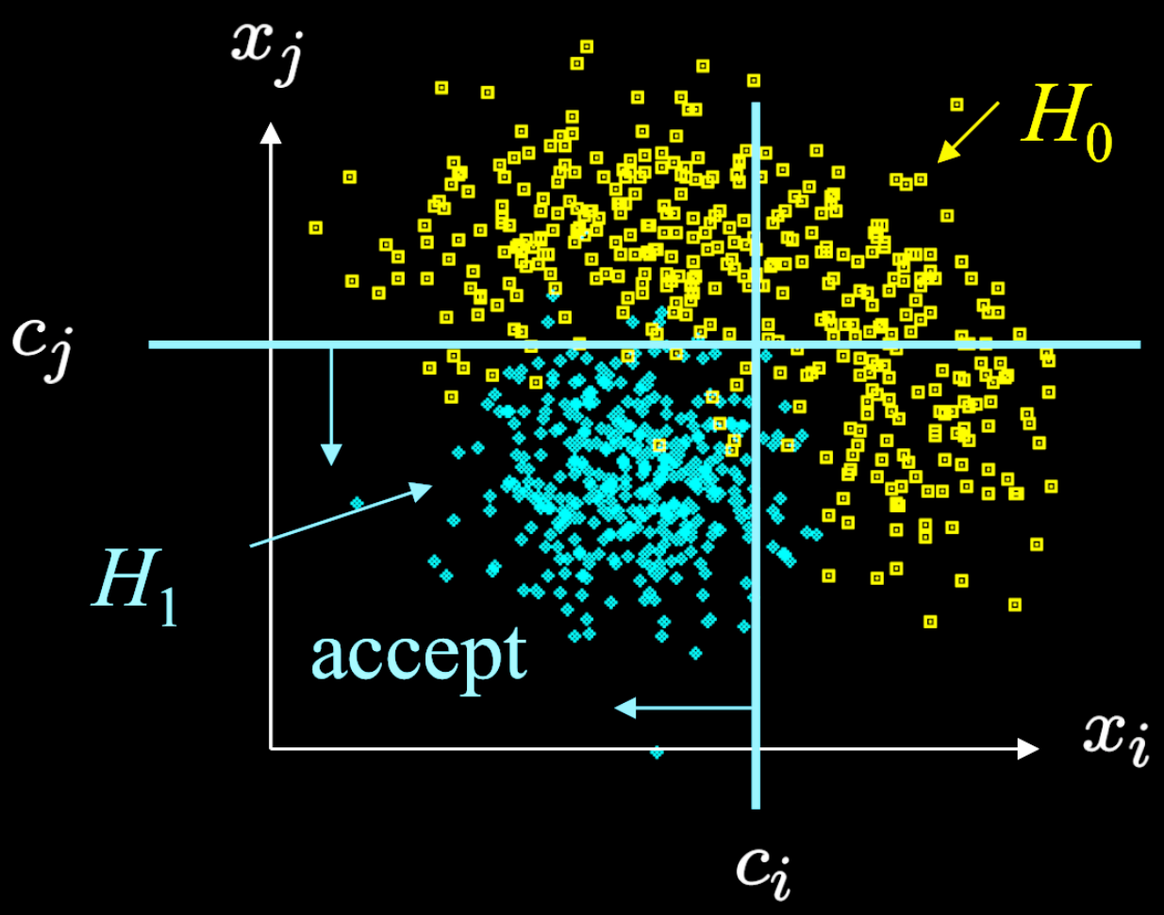
Let's start with a common task in particle physics: classification. We classify candidates (or events/reactions) as signal or background, we classify particles according to their types (particle ID), etc.

Classification task are now routinely done using ML. How? Why?

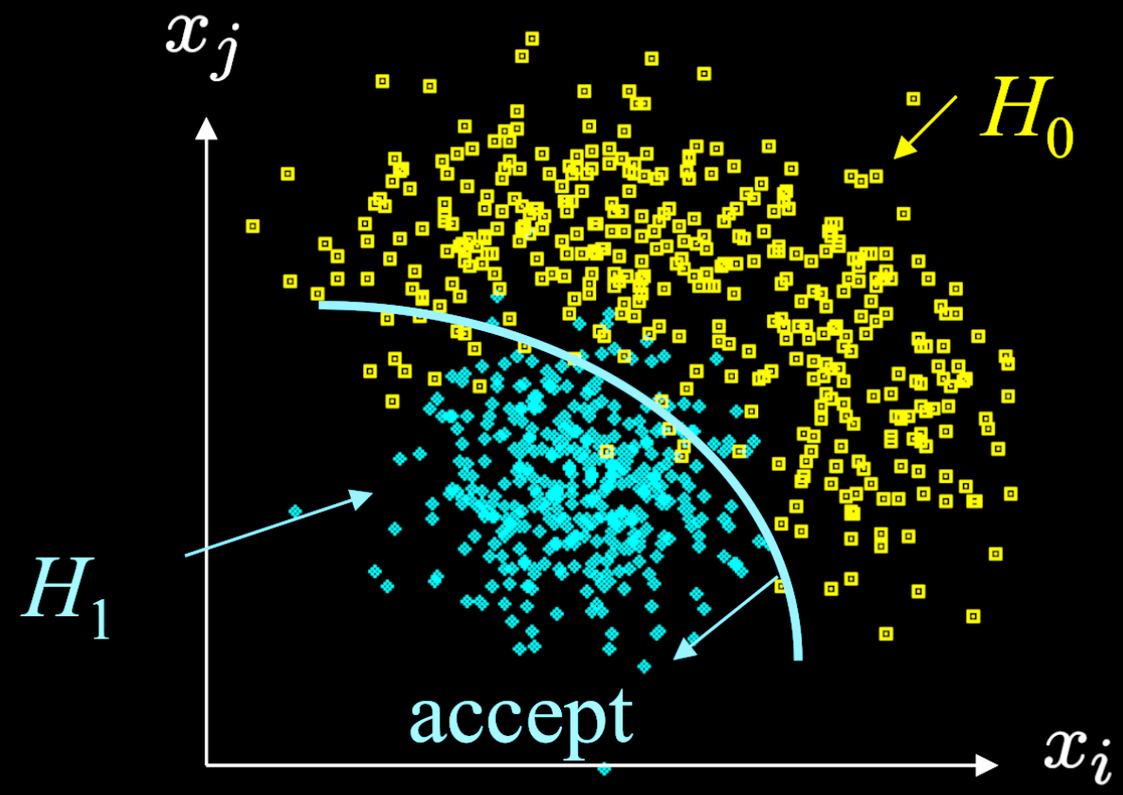
Classification

Consider two types of objects, (H_0) background and (H_1) signal, what is the optimal way to select a region to study the signal?

rectangular cuts



n-D manifold (some function of x_n)



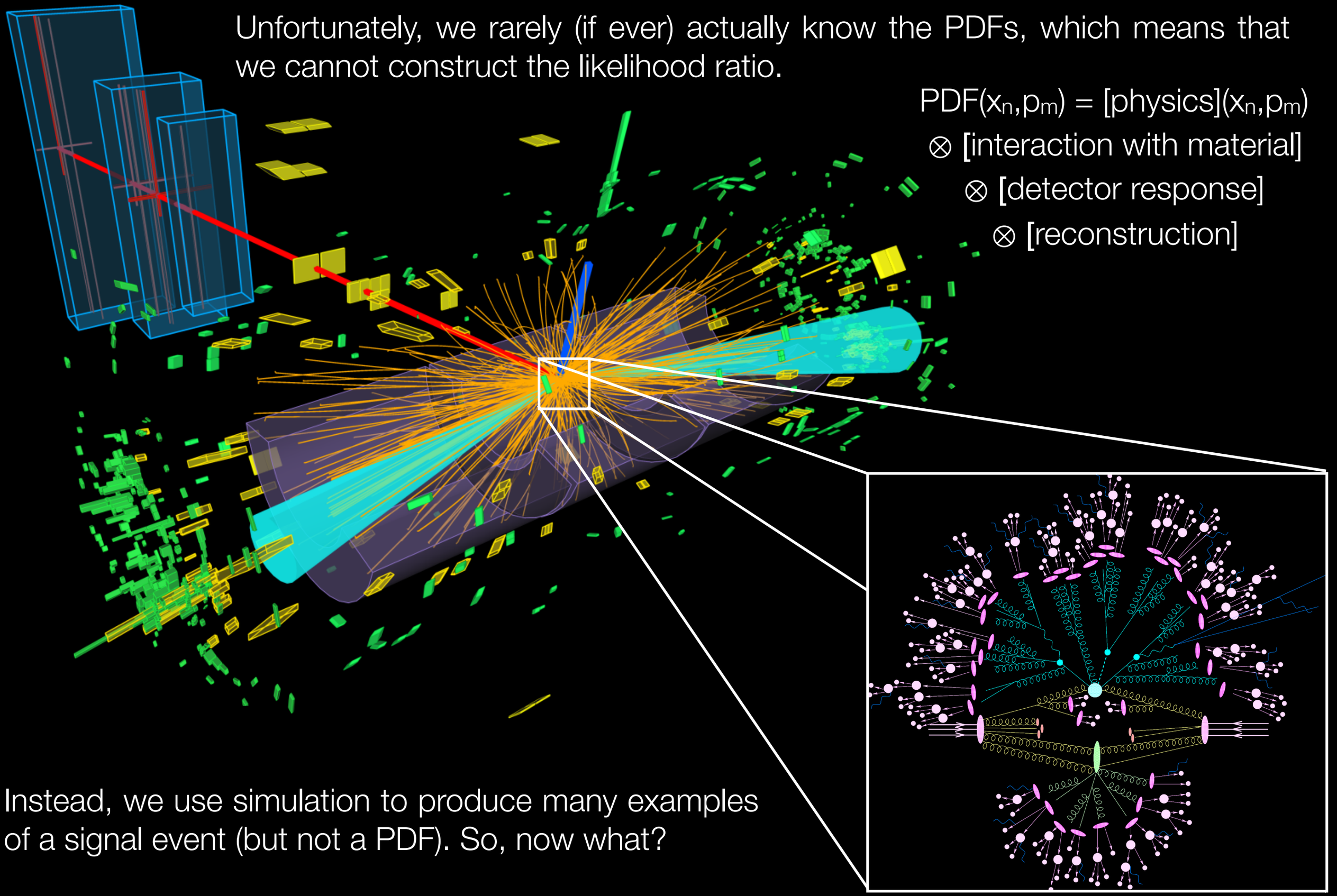
[images: G Cowan]

Problem solved in 1933: Neyman & Pearson proved that optimal boundaries are given by likelihood-ratio isobars; i.e. keep the regions with $\Lambda(x_n) >$ some constant.

The Problem

Unfortunately, we rarely (if ever) actually know the PDFs, which means that we cannot construct the likelihood ratio.

$$\text{PDF}(x_n,p_m) = [\text{physics}](x_n,p_m) \otimes [\text{interaction with material}] \otimes [\text{detector response}] \otimes [\text{reconstruction}]$$

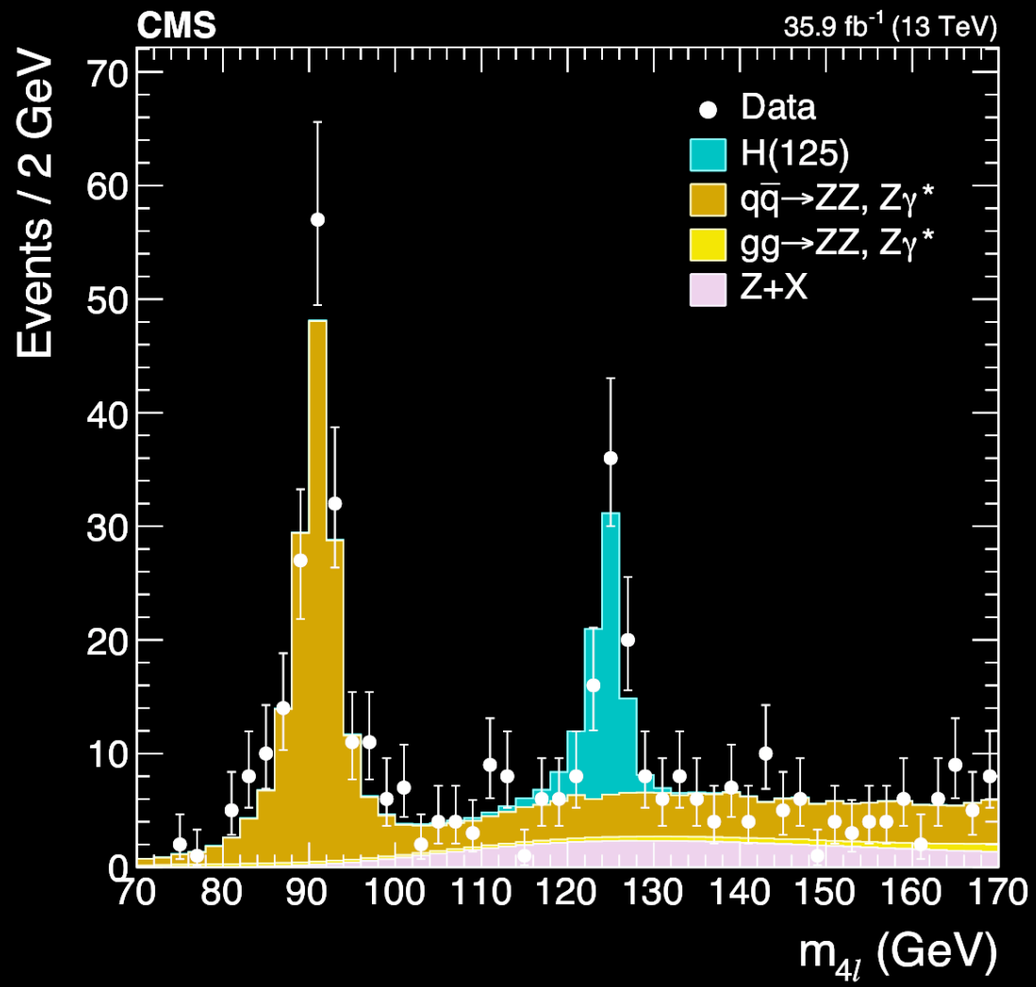


Instead, we use simulation to produce many examples of a signal event (but not a PDF). So, now what?

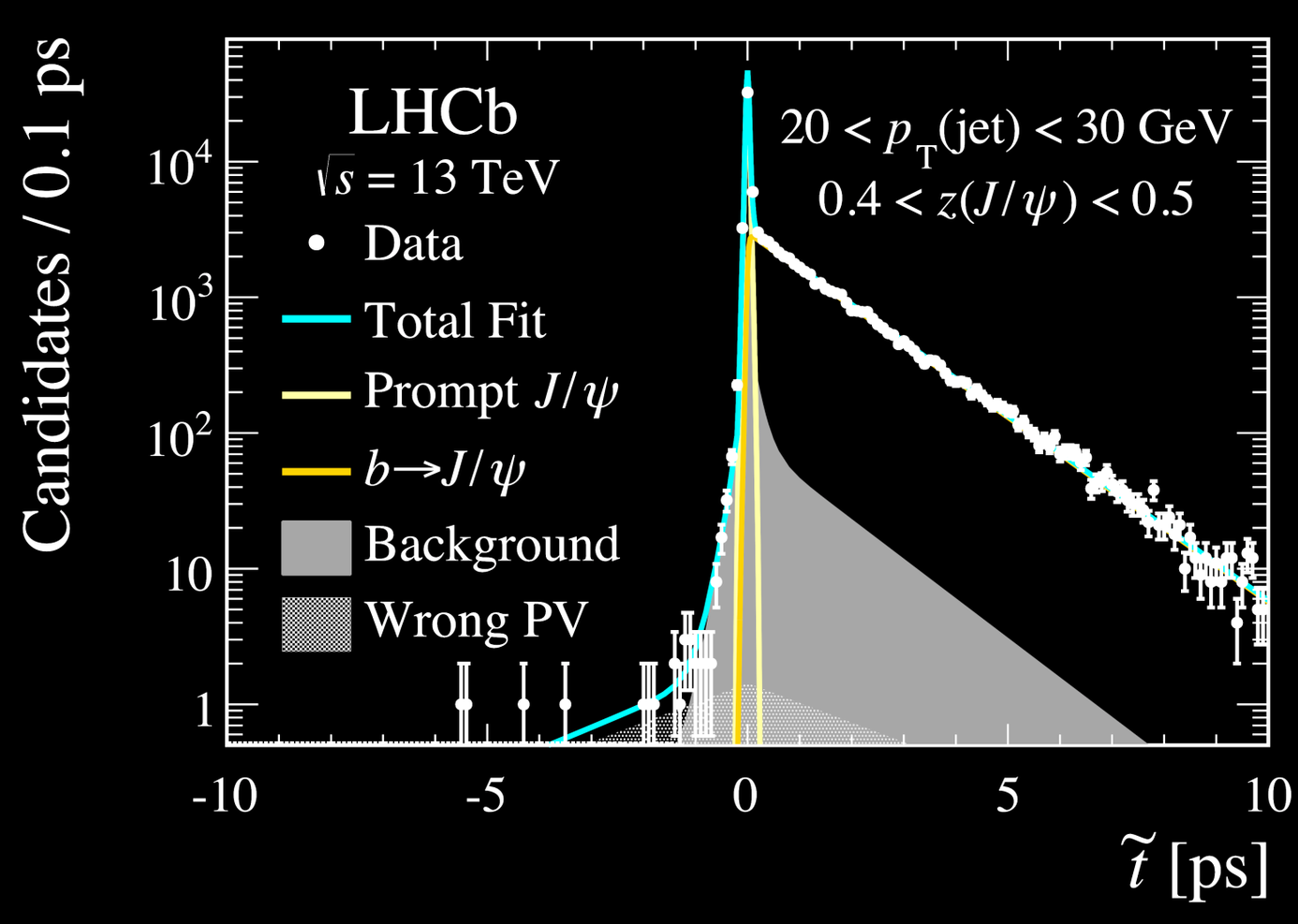
One Solution

In one dimension, we can use these examples to build a PDF, e.g. using a histogram, a kernel, or assuming some parametric function and fitting it to the examples (perhaps we leave some *nuisance* parameters free later).

[1706.09936]



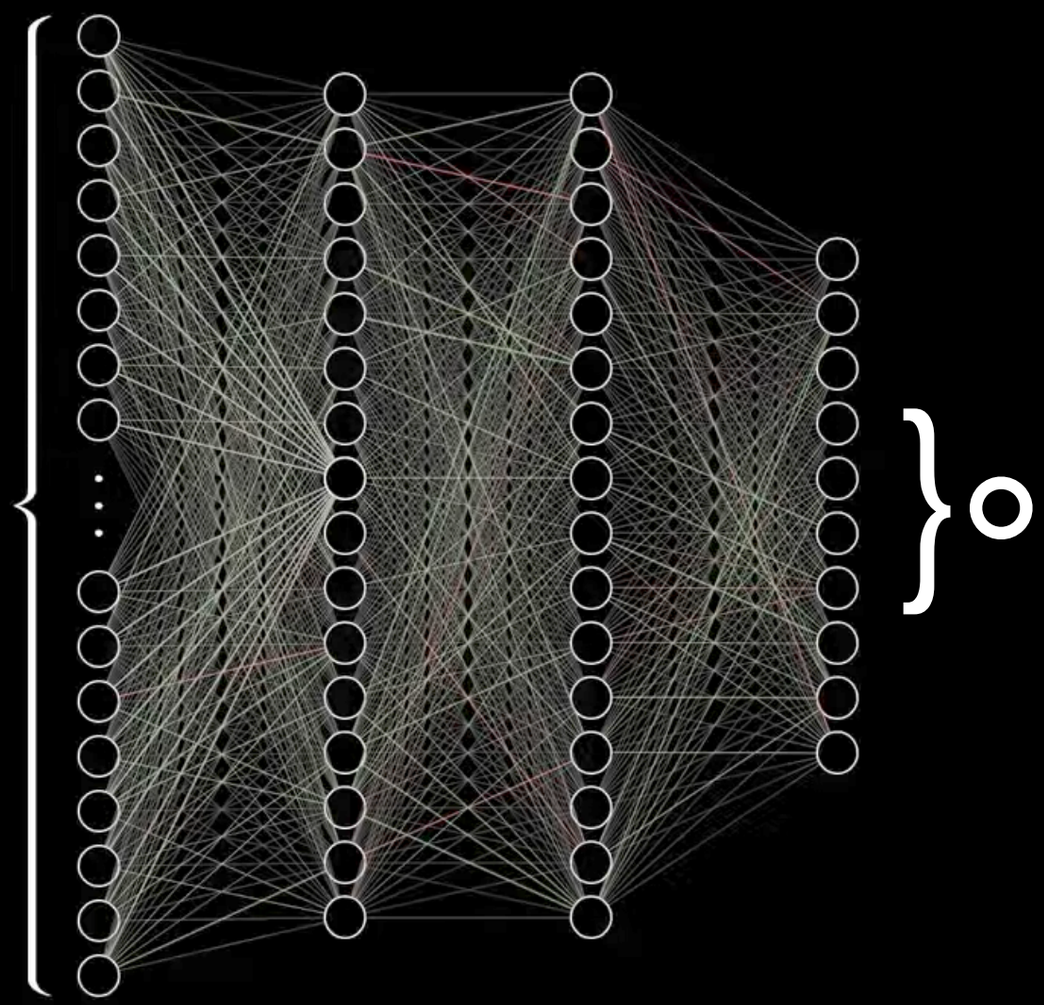
[1701.05116]

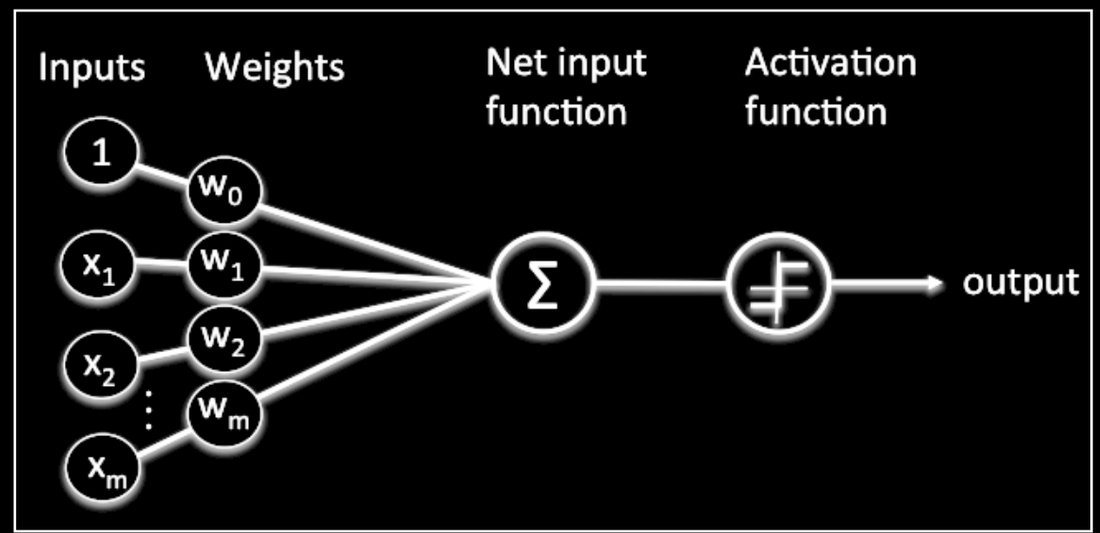


Physicists are good at feature engineering, many analyses boil 10M sensors down to a single feature (e.g. the mass). However, if we need a PDF in more than ~2 dimensions, the *curse of dimensionality* requires exponentially larger samples. Is there a better way?

A Better Solution

Using examples of signal and background, we want to learn a function, $F(x_n)$, that provides the same power as the likelihood ratio would — but without requiring that we know the PDFs (clearly, any monotonic function of the form $F(\Lambda(x_n))$ will do this).



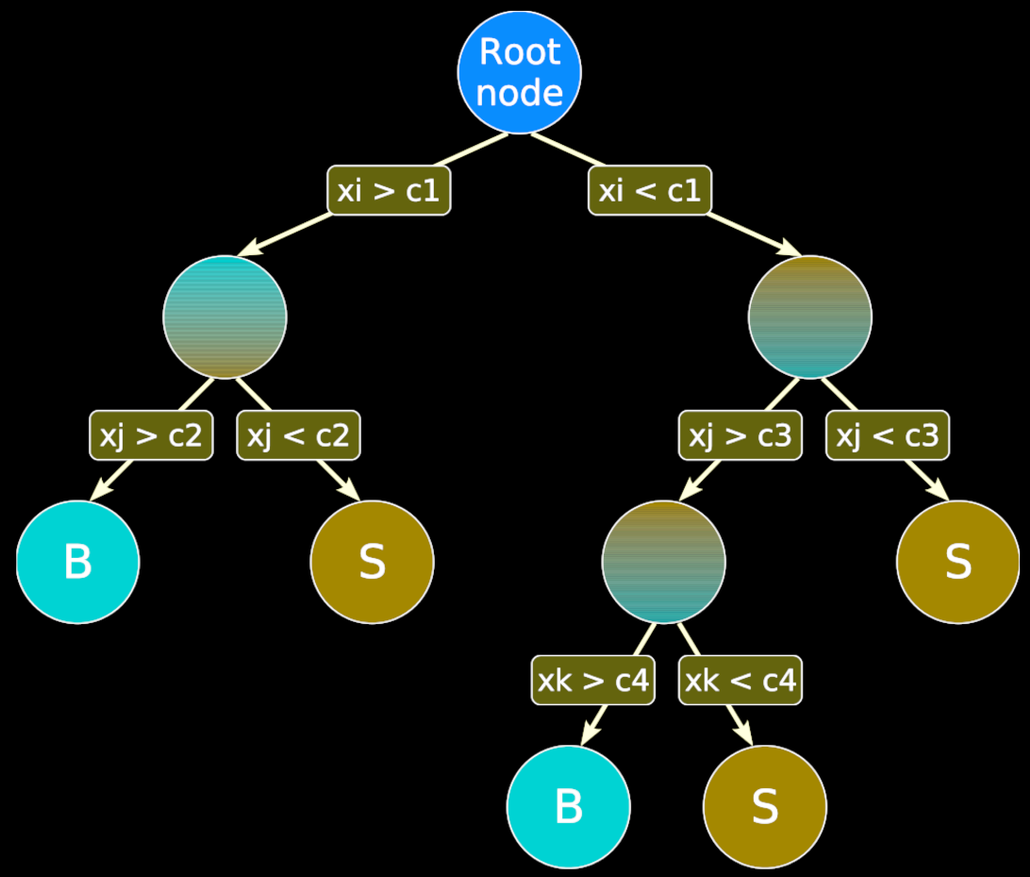


N.b., this is just a very complicated function of the form $F(x_n;w_m)$! The deeper the network, the more complex structures we can (easily) generate.

Now, just choose a quantity to optimize, and use gradient descent to learn the weights, w_m. Ideally the result is $F(\Lambda(x_n))$, though we don't know F so can't obtain Λ itself.

Other ML Algorithms

There are many ML algorithms on the market other than neural networks, though in HEP the only other one commonly used (a lot in the past, less so now) is the BDT.

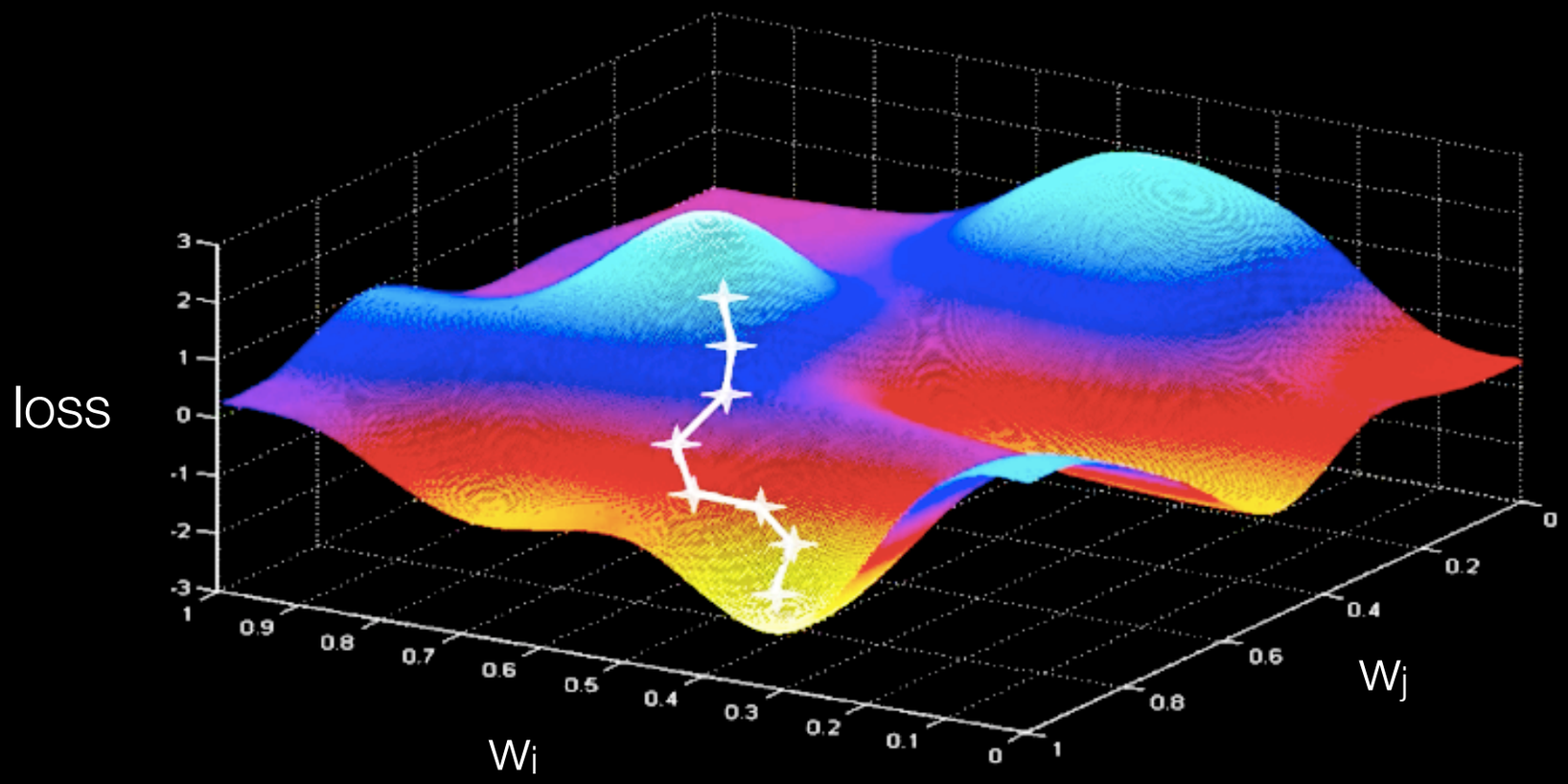


Hopefully it's clear that a single DT is a weak classifier; however, by boosting—training a series of DTs where the weights for misclassified events are increased as the series is trained, a classic example is AdaBoost—or bagging—a large number of bootstrap-copy data samples are produced, a separate DT is trained on each, and the final BDT response is the majority vote of the DTs—the final BDT can become powerful.

Training & Loss

Typically in HEP, we use supervised learning. This means that the training data are labeled, e.g. signal and background samples are provided where each event is labeled by its category. Almost always this labeled data is simulated for signal. A mixture of simulation and experimental data is used for backgrounds (problem dependent).

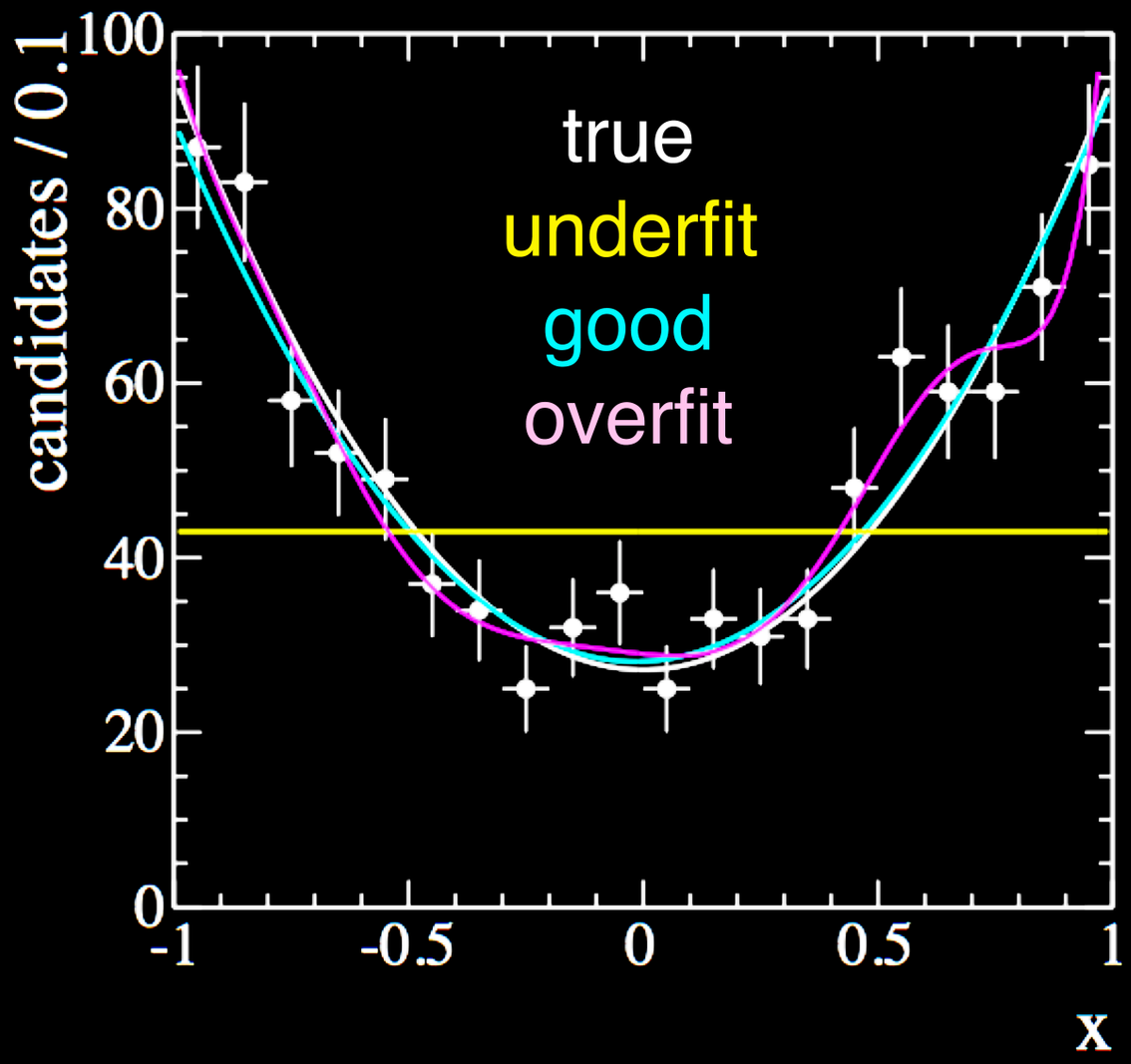
loss = F(prediction errors) (though not always)



Once you define a loss function, the goal of the training is to minimize the loss — which is commonly done using gradient descent (as is done in fits you've run).

Overtraining

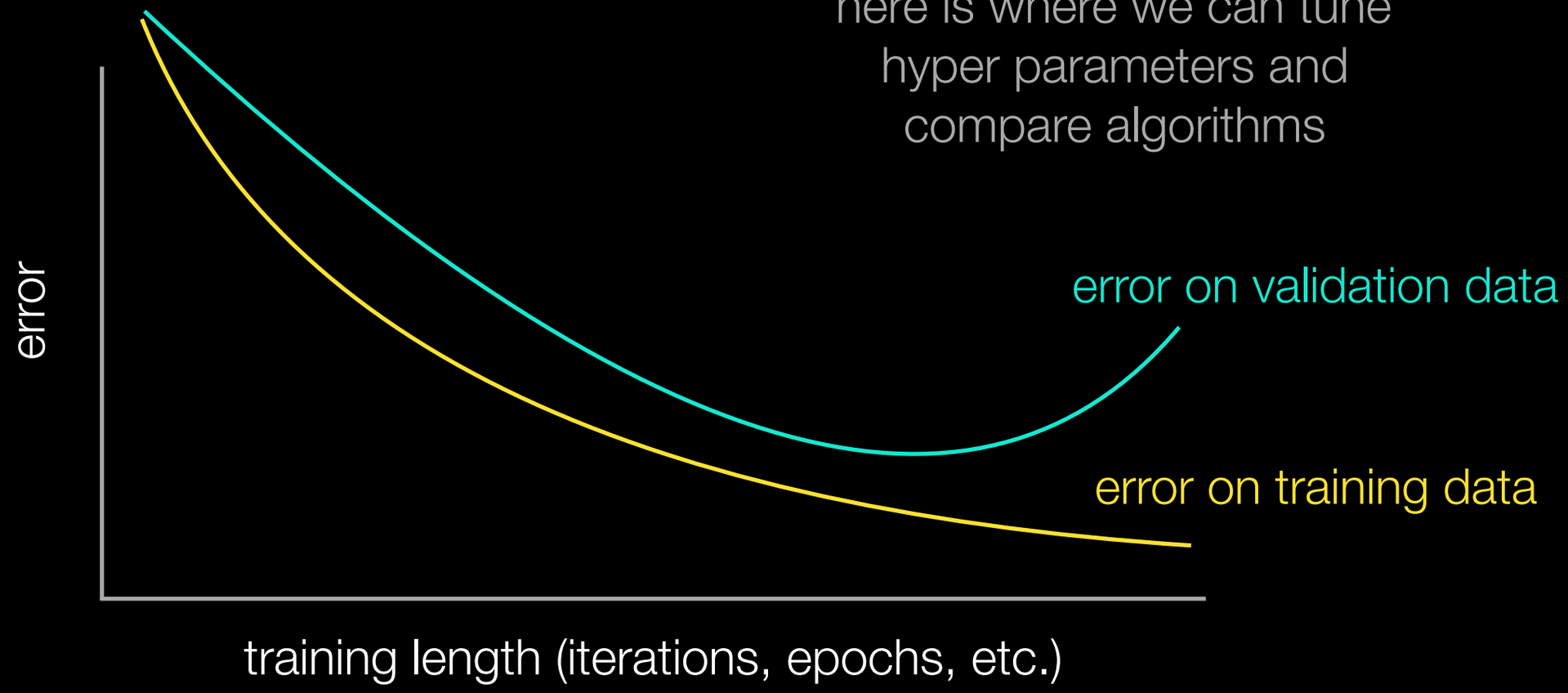
Even in a much more simple example—fitting 1-D data to a polynomial—if we give the fit function too much complexity it will definitely overfit the data. This means that we expect the model to describe unseen data worse than a more simple function.



A neural network is an extremely complex function. How do we prevent overtraining given that unconstrained minimization of so many parameters could easily overtrain?

Train, Validate, Test

The most common counter-attack to overtraining is to split the data into training, validation, and testing samples. (If the sample is not large enough, one can k-fold it.) Checking the error on validation data using cross validation is one way to spot obvious overtraining.

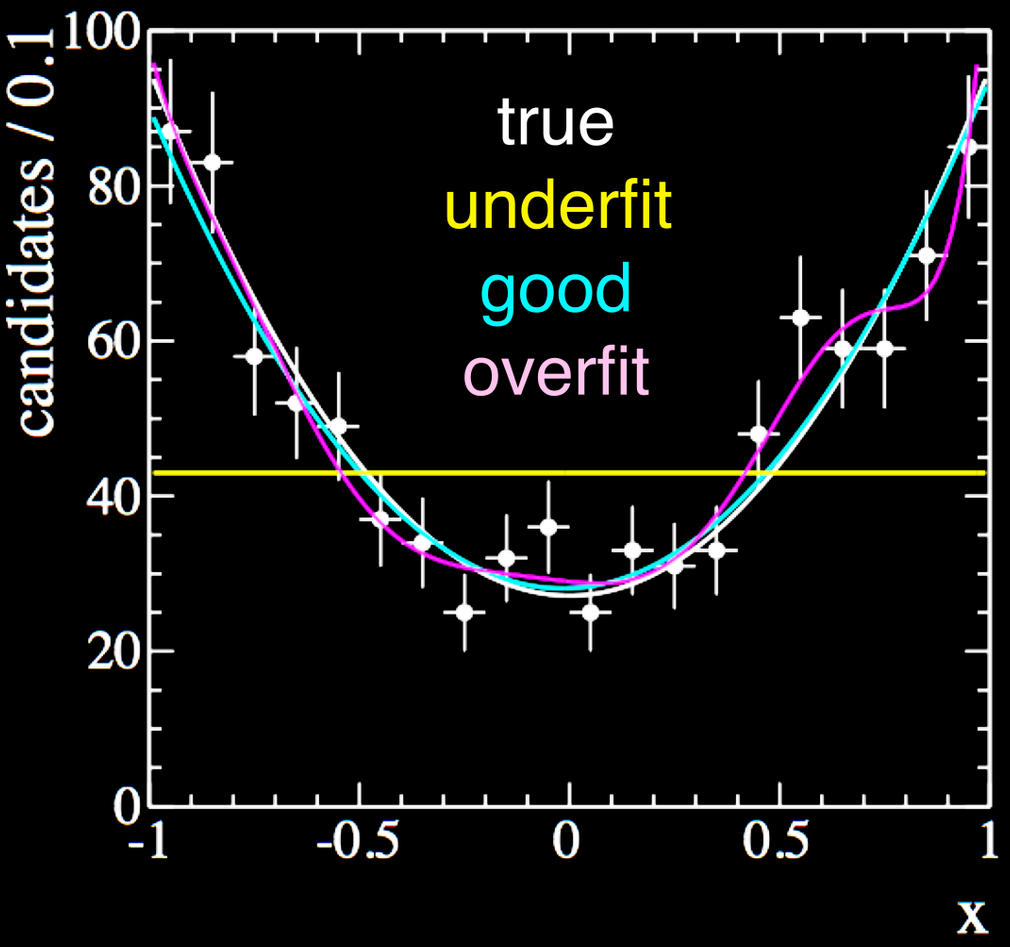


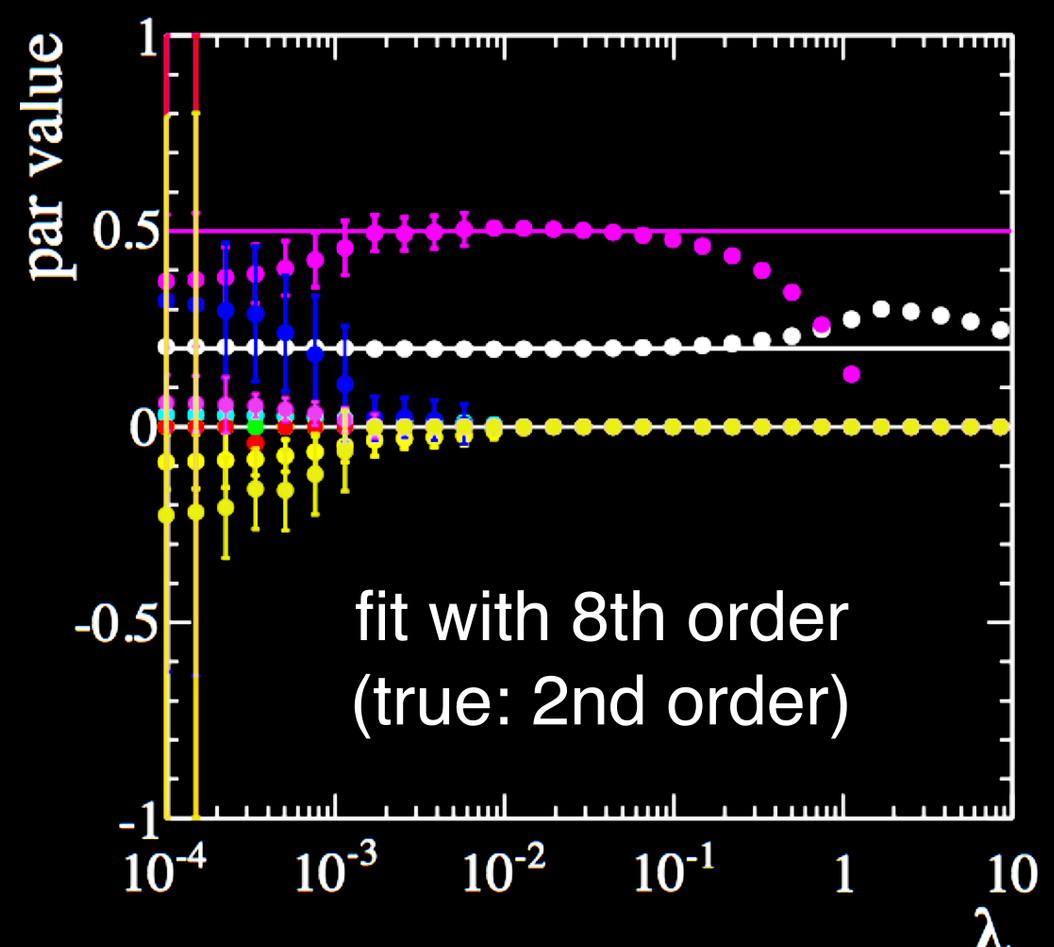
Ultimately, we determine the performance in a data-driven way using calibration data (independent of the training and validation). This gives an unbiased measurement of the performance. (This is sufficient for usage, e.g., even if the algorithm is sub-optimal, provided an unbiased performance determination exists, it can be used in an analysis.)

Regularization

Outside of ML, you may be aware of regularization methods, e.g., the LASSO or Ridge, or information criteria such as AIC or BIC. These methods can also be used to remove unnecessary complexity from the ML model.

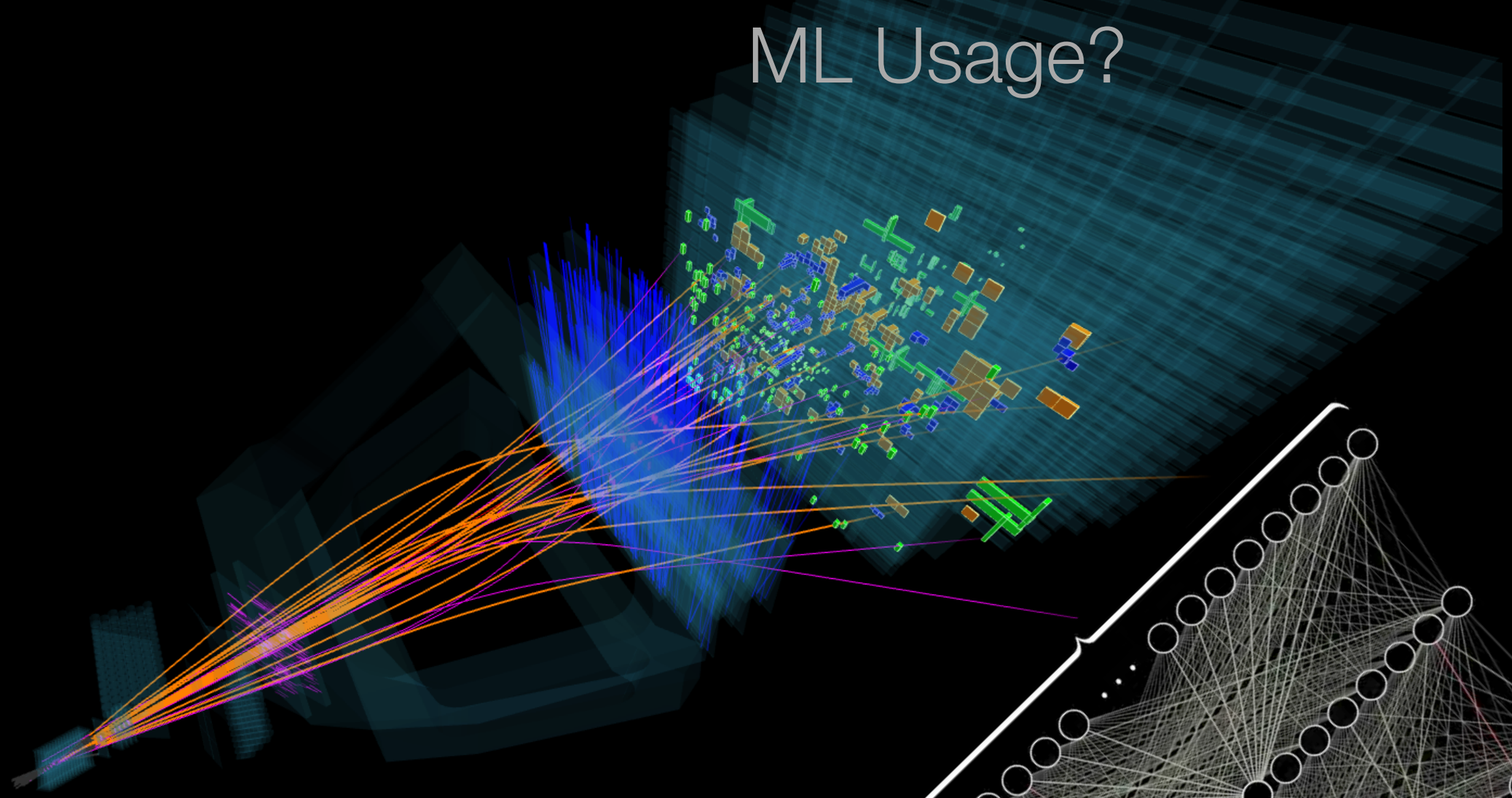
$$\chi^2 \rightarrow \chi^2 + \lambda \sum |\alpha_i|$$

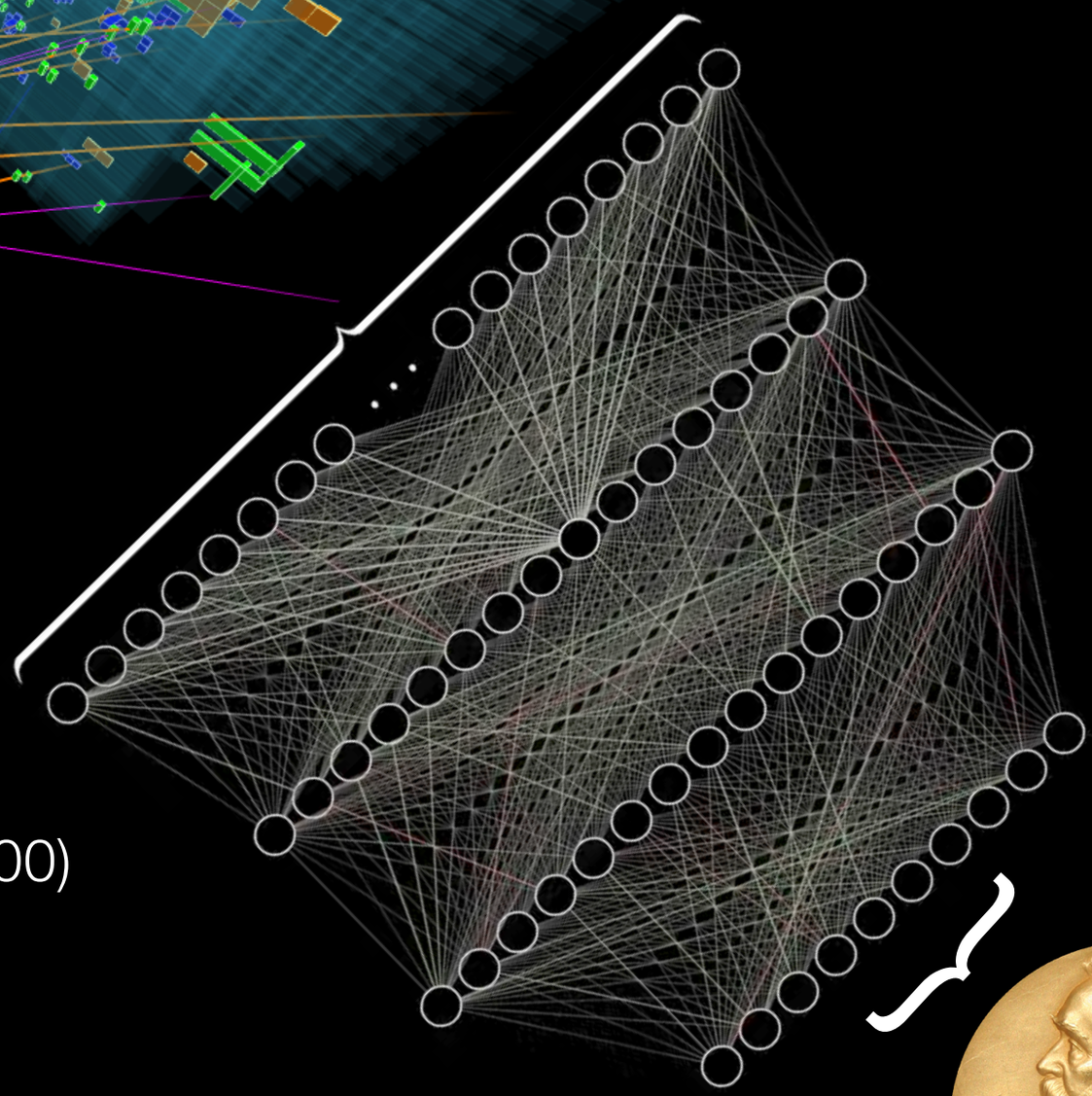




Another simple form of regularization commonly used is dropout, where nodes are ignored at random during the training.

ML Usage?





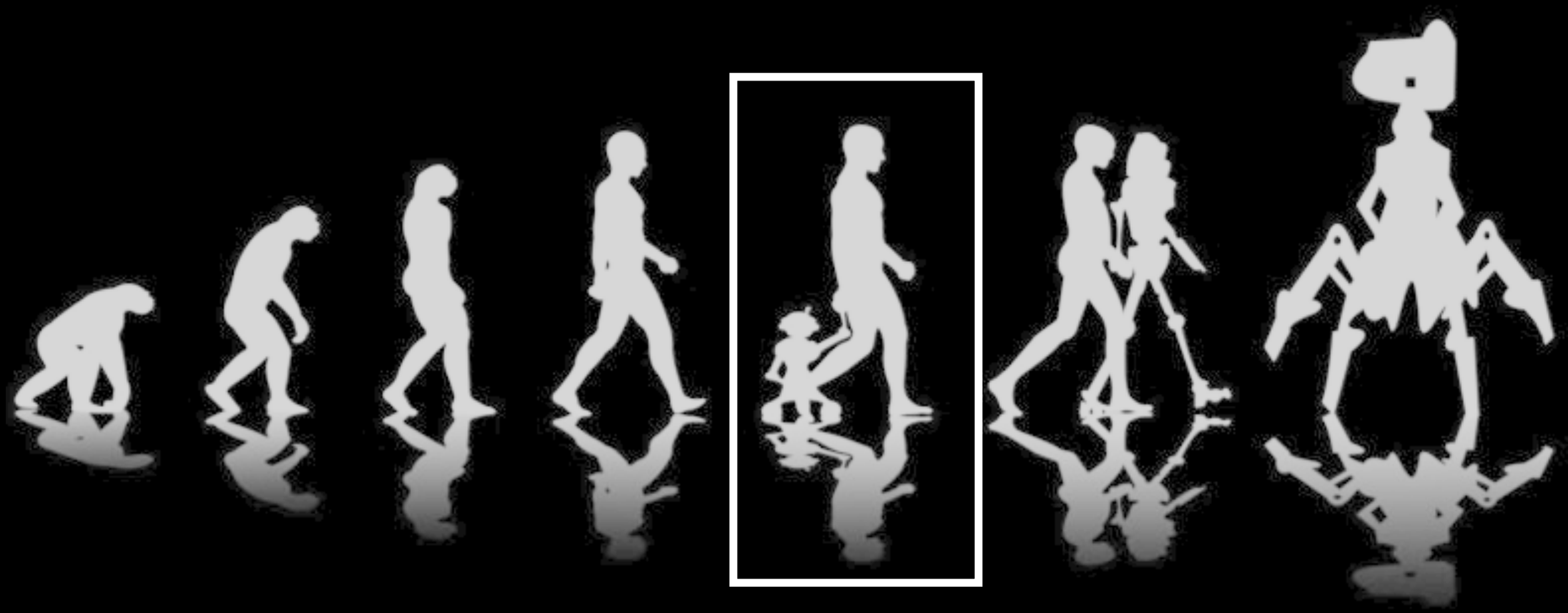
What should the input features be?

- O(10-100M) hits?
- O(100) features for each of O(100) reconstructed particles?
- O(10) high-level physics features?

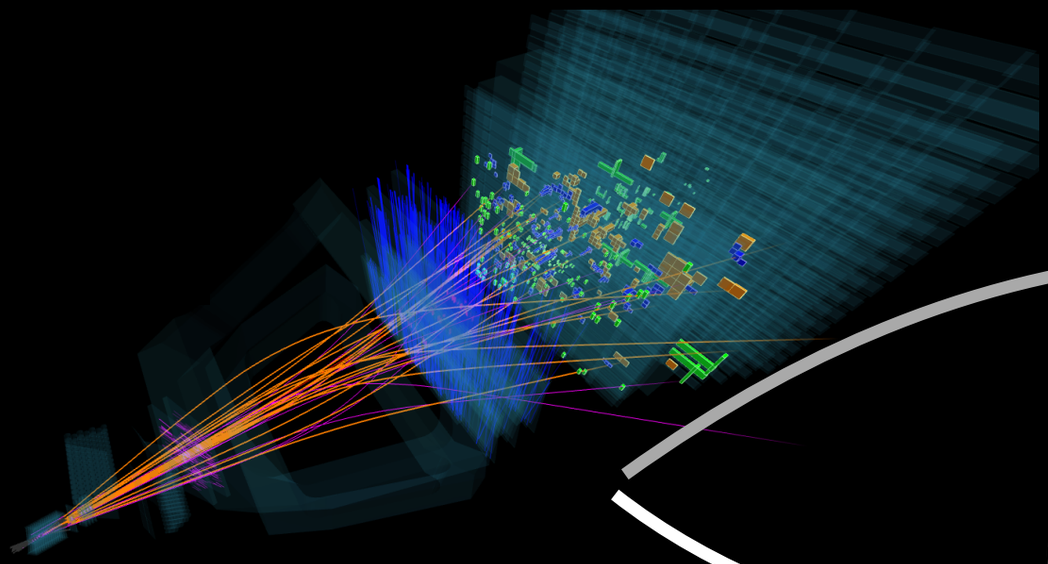
How do we ensure it's doing something sensible?
What about systematics? […]

ML as a Tool for Discovery

Let's start by using physicist-designed high-level features input to shallow algorithms, i.e. simply replace our usual cuts with better functions but largely using the same features. Where have we been using ML?



Big Data & Real-Time Analysis



5 TB/s (Run 3)
post zero suppression

Heavy use of machine learning:
V.Gligorov, MW, JINST 8 (2012) P02013
T.Likhomanenko et al [1510.00572]

partial event reconstruction on 50k CPU cores (ML-based filtering)

1 TB/s
post zero suppression

simple feature-building/selection on FPGAs

50 GB/s

6 GB/s
30 GB/s

data buffered on O(10 PB) disk

full event reconstruction (ML-based filtering)

0.7 GB/s
3 GB/s

permanent storage

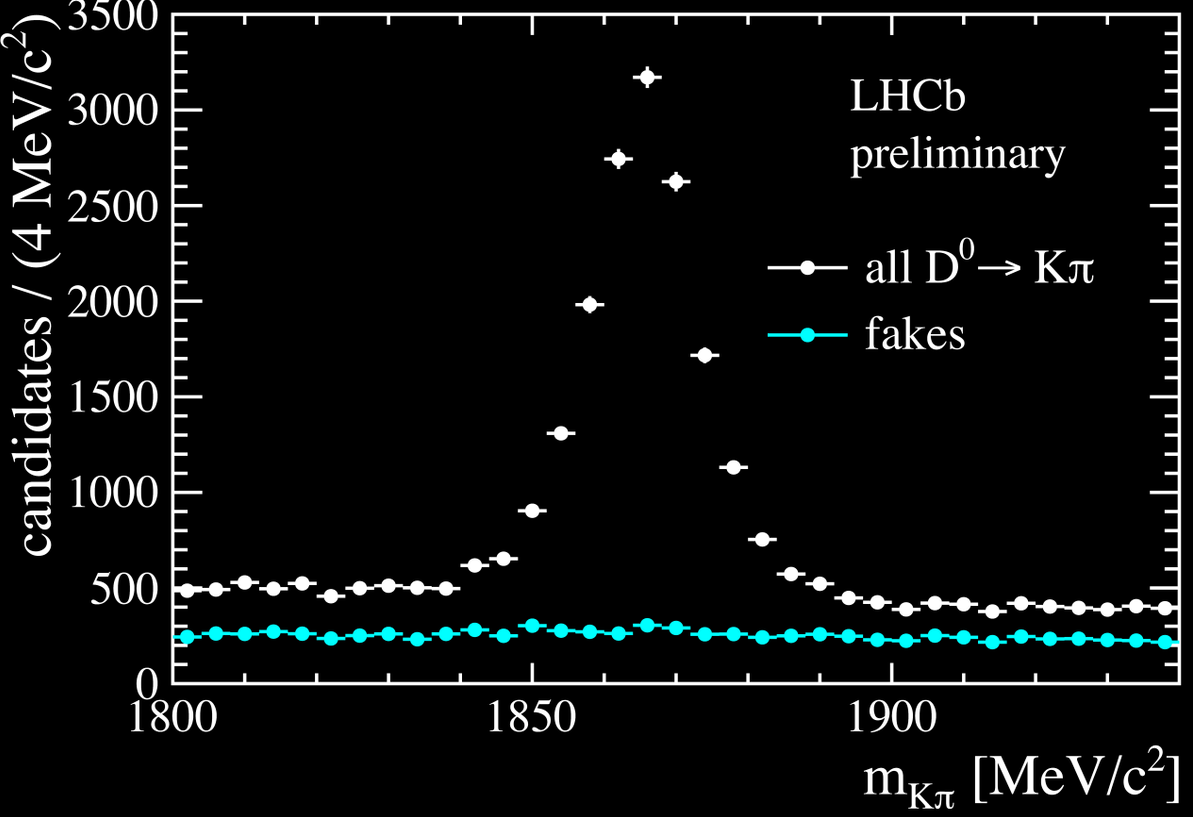
real-time alignment & calibration

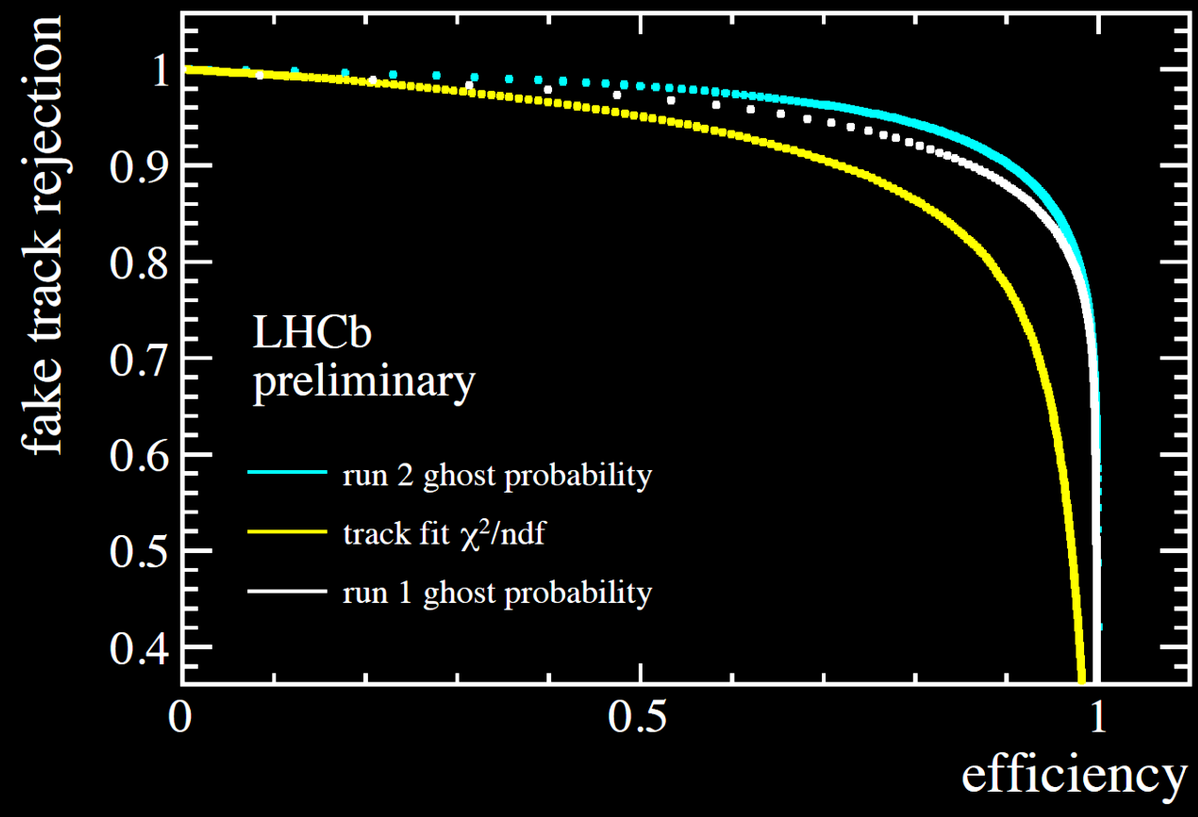
Additionally, ML is now being used to gain actionable insights from computing metadata and used to increase the efficiency of our computing resource usage.

Fake Tracks

Fake-track-killing neural network based on 21 features, most important are hit multiplicities and track-segment chi2 values from tracking subsystems. Significantly reduces the rate of events selected in the first software-trigger stage.

LHCb-PUB-2017-011

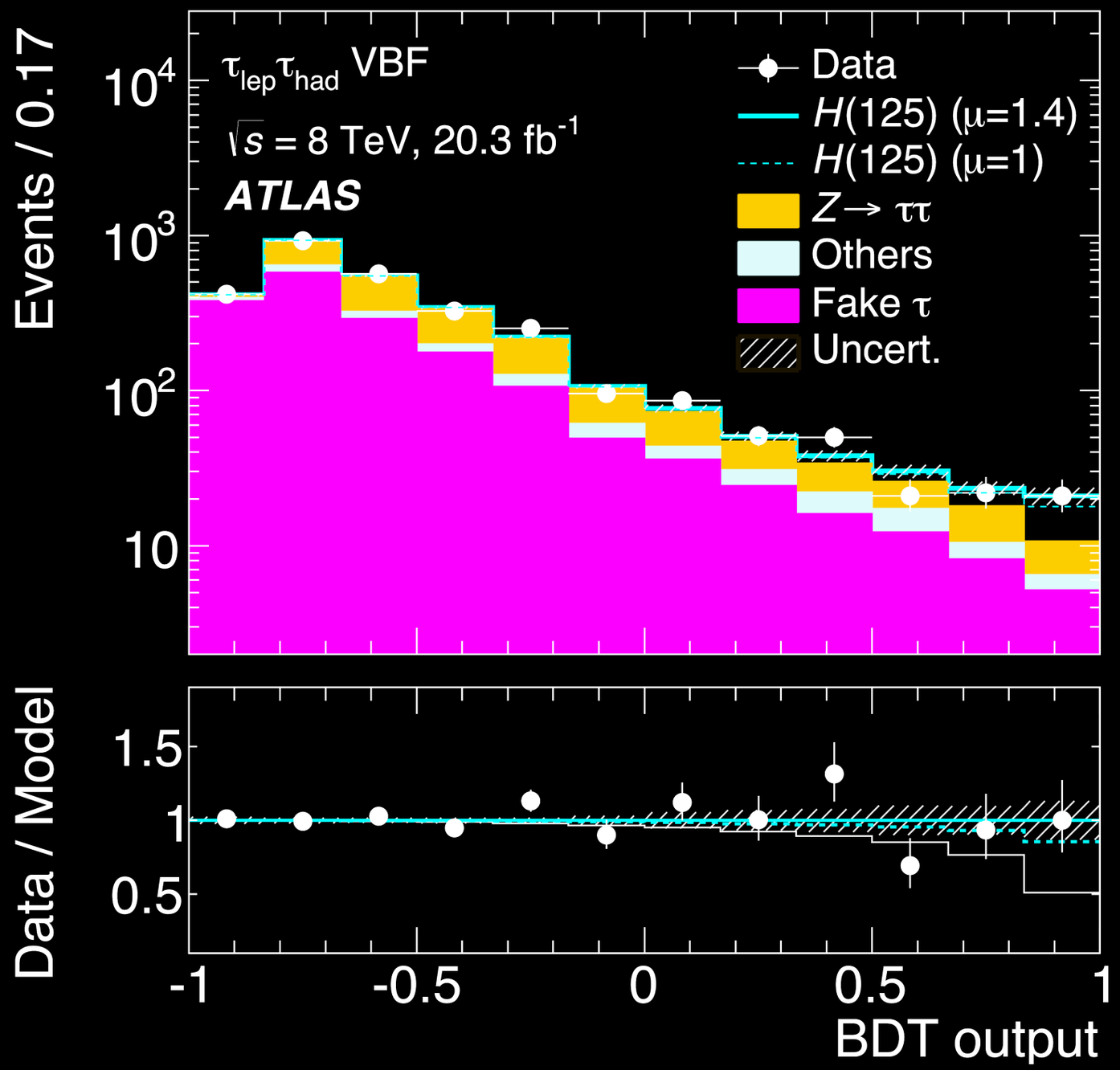




Performance evaluated using standard candle signals with and without applying a criterion on the fake-tracking-filling NN.

Higgs Discovery & Properties

ML played a key role in the discovery of the Higgs boson, especially in the diphoton analysis by CMS where ML (used to improve the resolution and to select/categorize events) increased the sensitivity by roughly the equivalent of collecting ~50% more data.



Measuring Higgs Properties

Example ML usage for the Higgs [1501.04943]

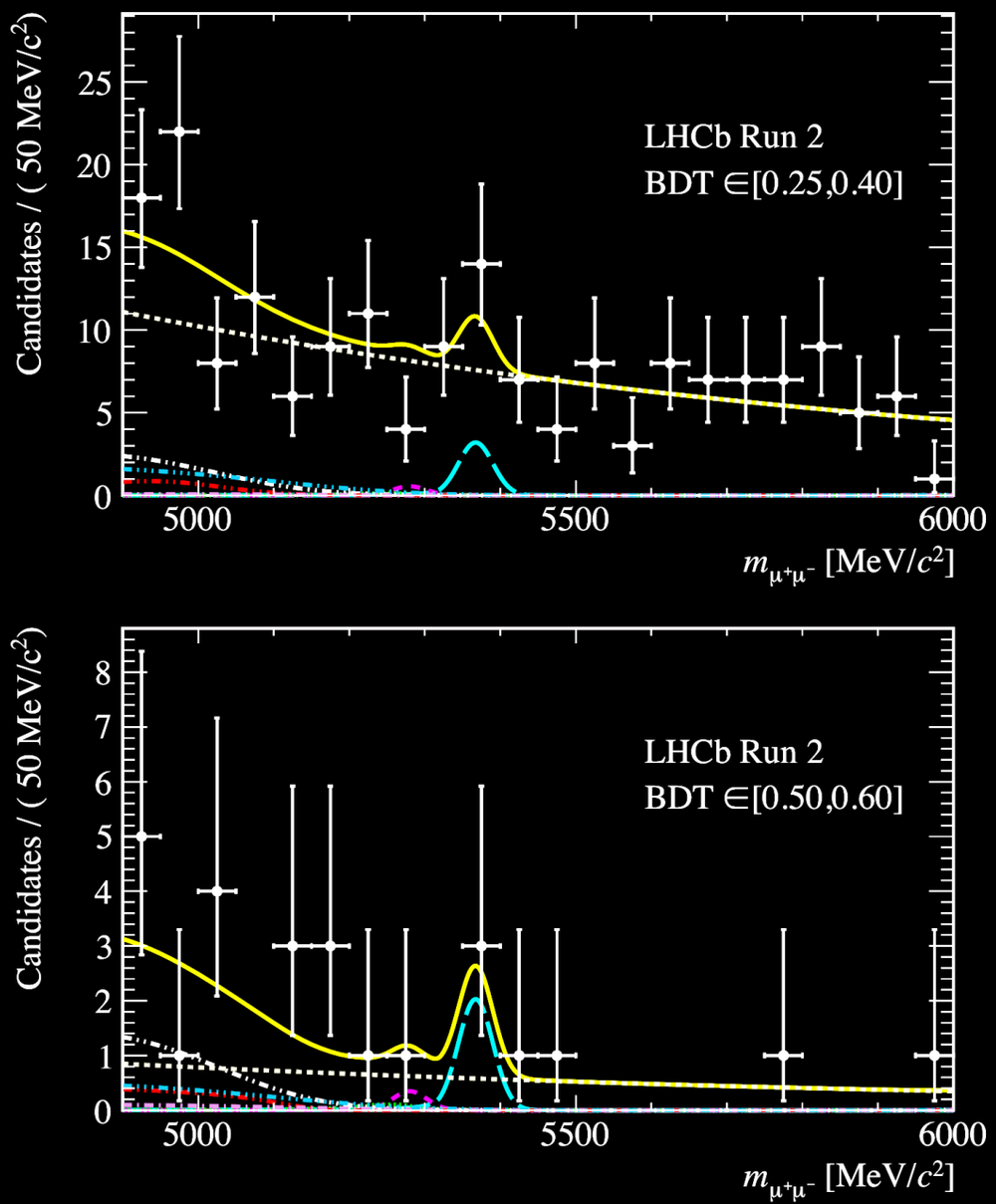
The study of tau leptons is complicated by the fact that they decay before they can be detected and by the loss of the subsequently produced neutrinos.

The ATLAS data sample was divided into 6 distinct kinematic regions, and in each a BDT was trained using 12 weakly discriminating features (improved sensitivity by ~40% vs a non-ML approach.)

High-Precision Tests of the SM

Only about 1 in every 300 billion pp collisions produces the decay Bs→μμ within the LHCb detector. LHCb uses BDT to suppress the overwhelming backgrounds to make the first single-experiment observation of this decay.

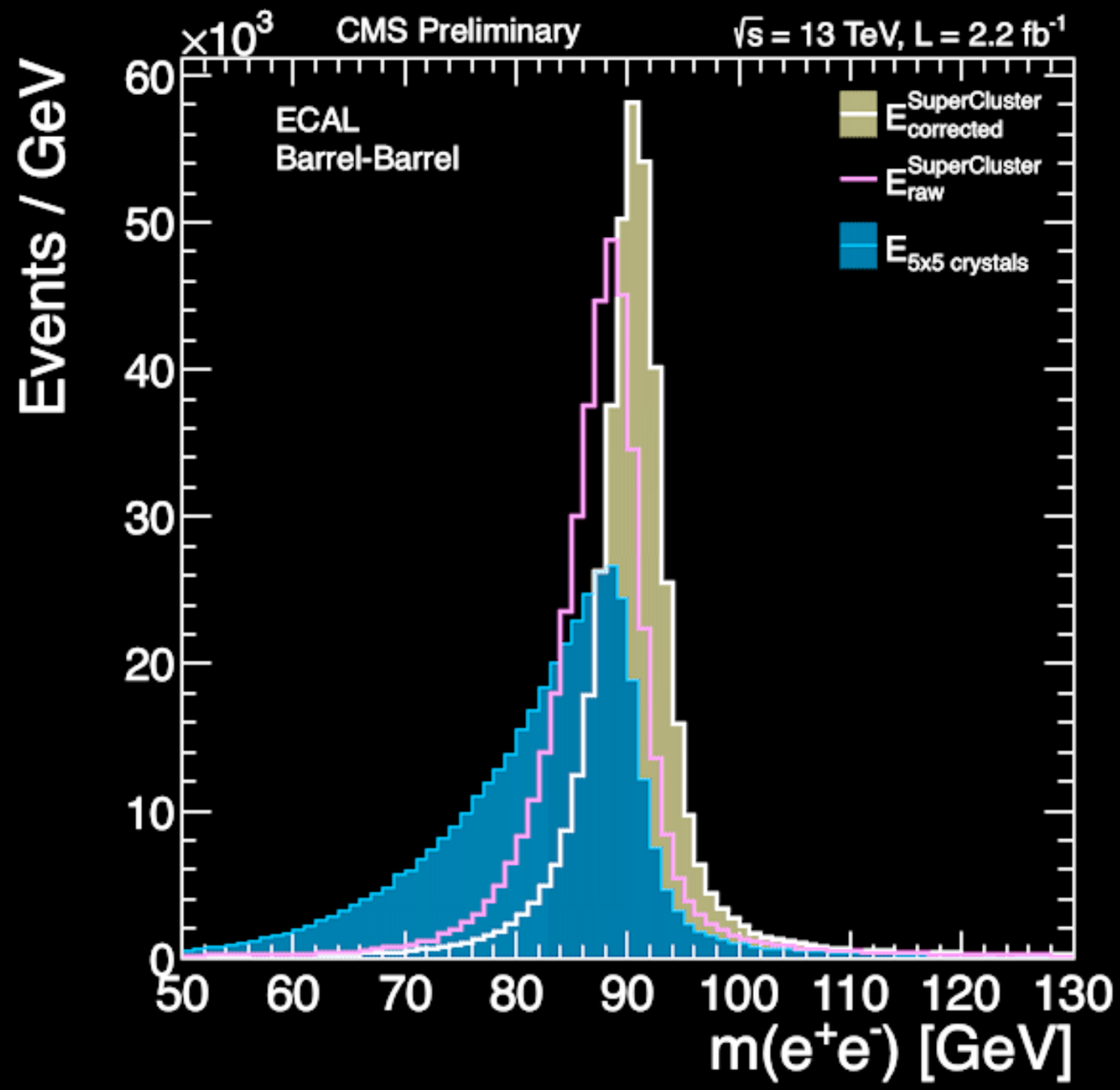
[1703.05747]



The observed decay rate agrees with the SM prediction within the archived precision of 25%. To obtain the same sensitivity without the use of ML would've required about 4 x more data.

Energy Resolution

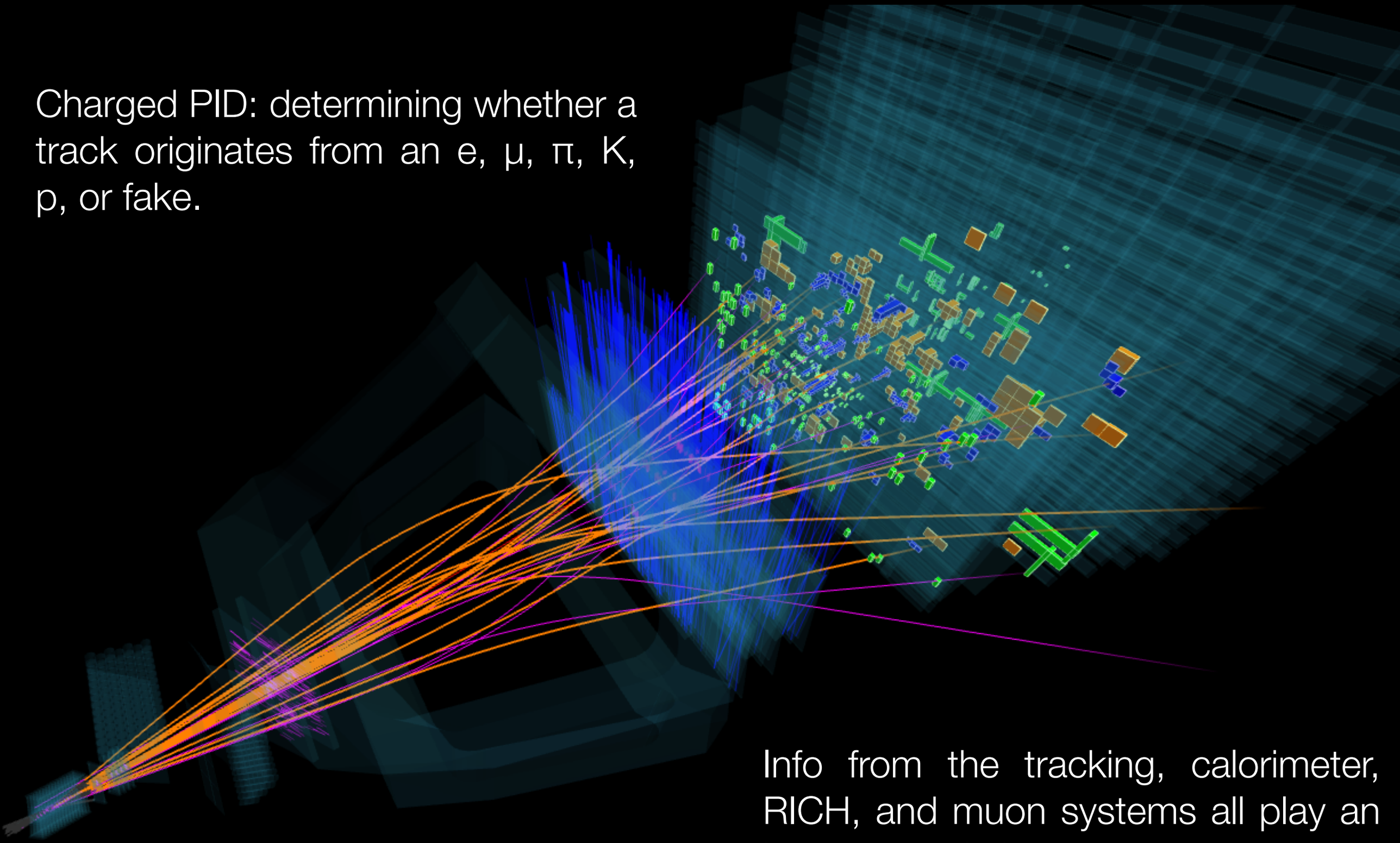
Using ML to improve the determination of particle properties is now commonplace in all LHC experiments, e.g., BDTs are used to improve the resolution of the CMS calorimeter.



Energy deposited is recorded by many sensors, which are clustered to recover the original particle energy. BDTs are trained to learn corrections using all information available in the various calorimeter sensors—which results in sizable improvement in resolution.

Case Study: Particle Identification @ LHCb

Charged PID: determining whether a track originates from an e, μ, π, K, p, or fake.

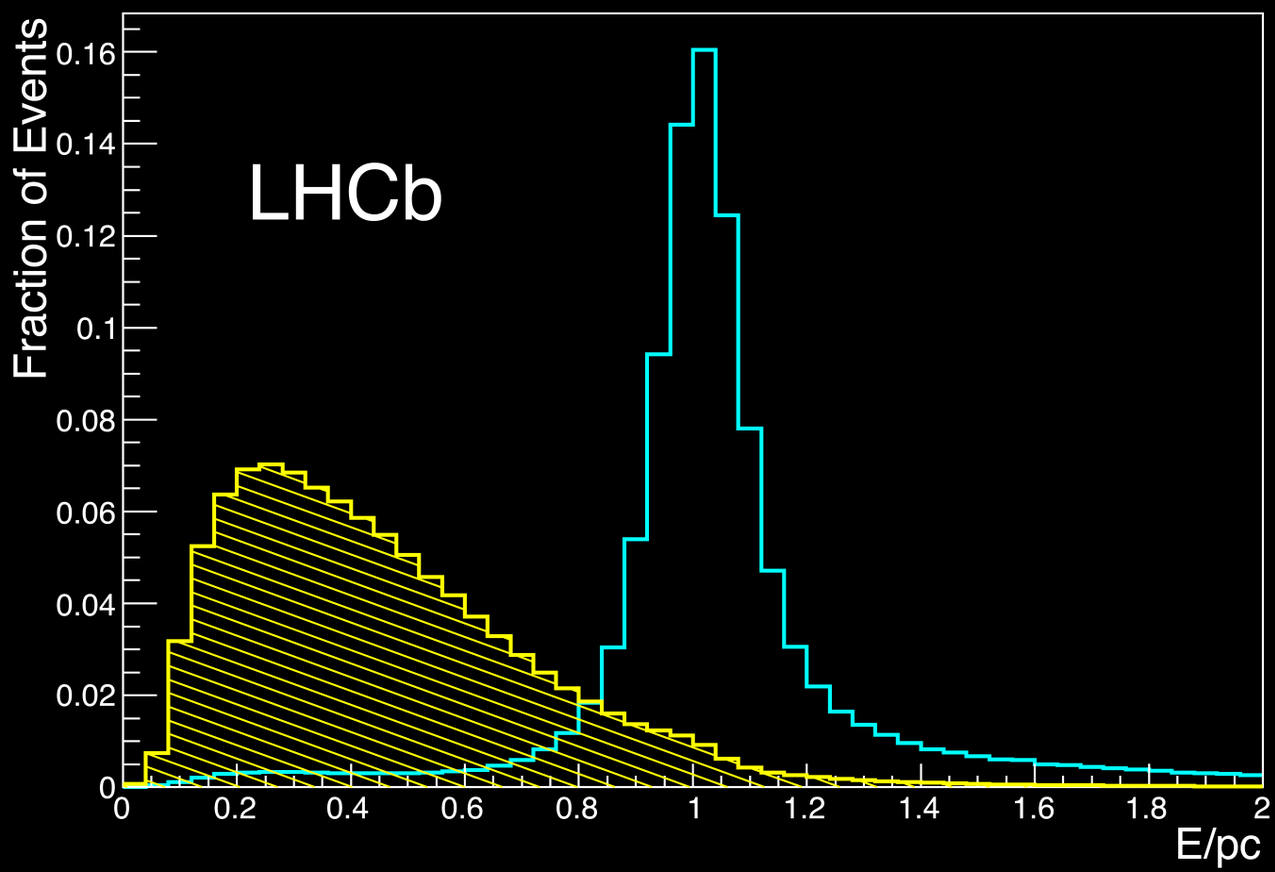


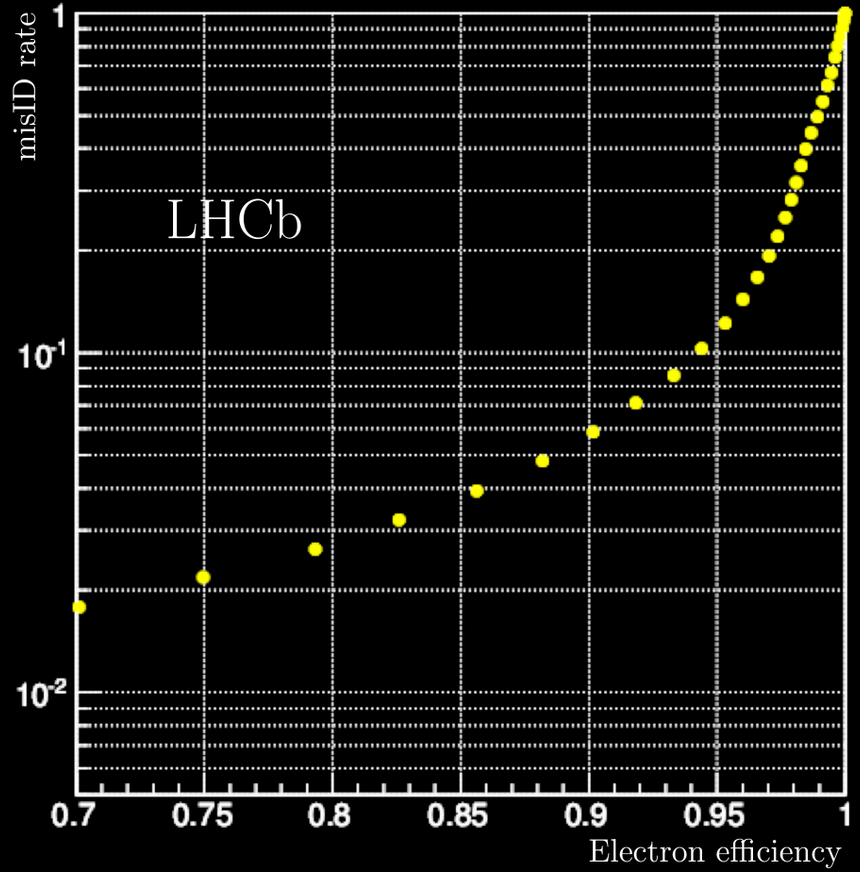
Info from the tracking, calorimeter, RICH, and muon systems all play an important role here.

Calorimeters

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

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

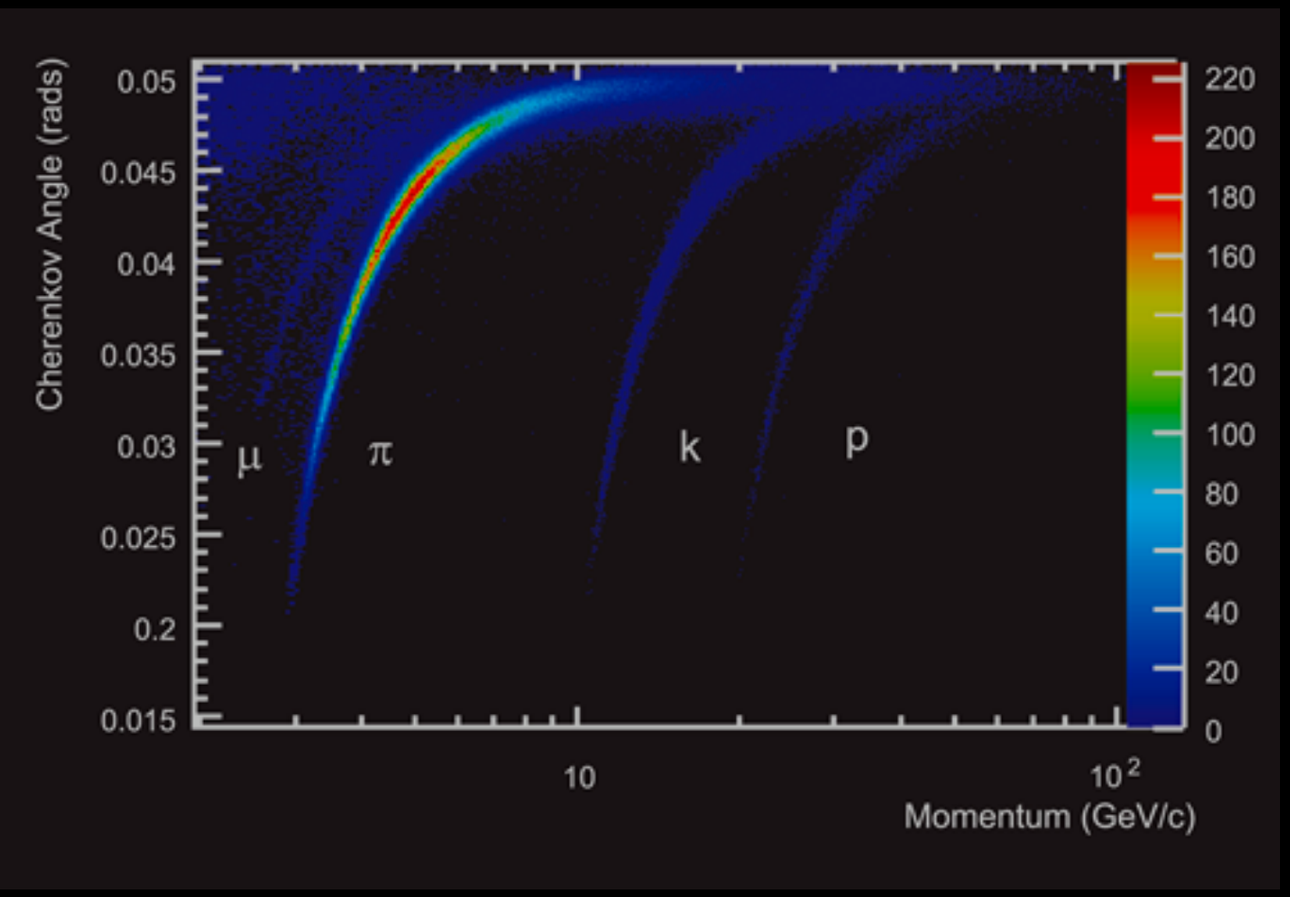


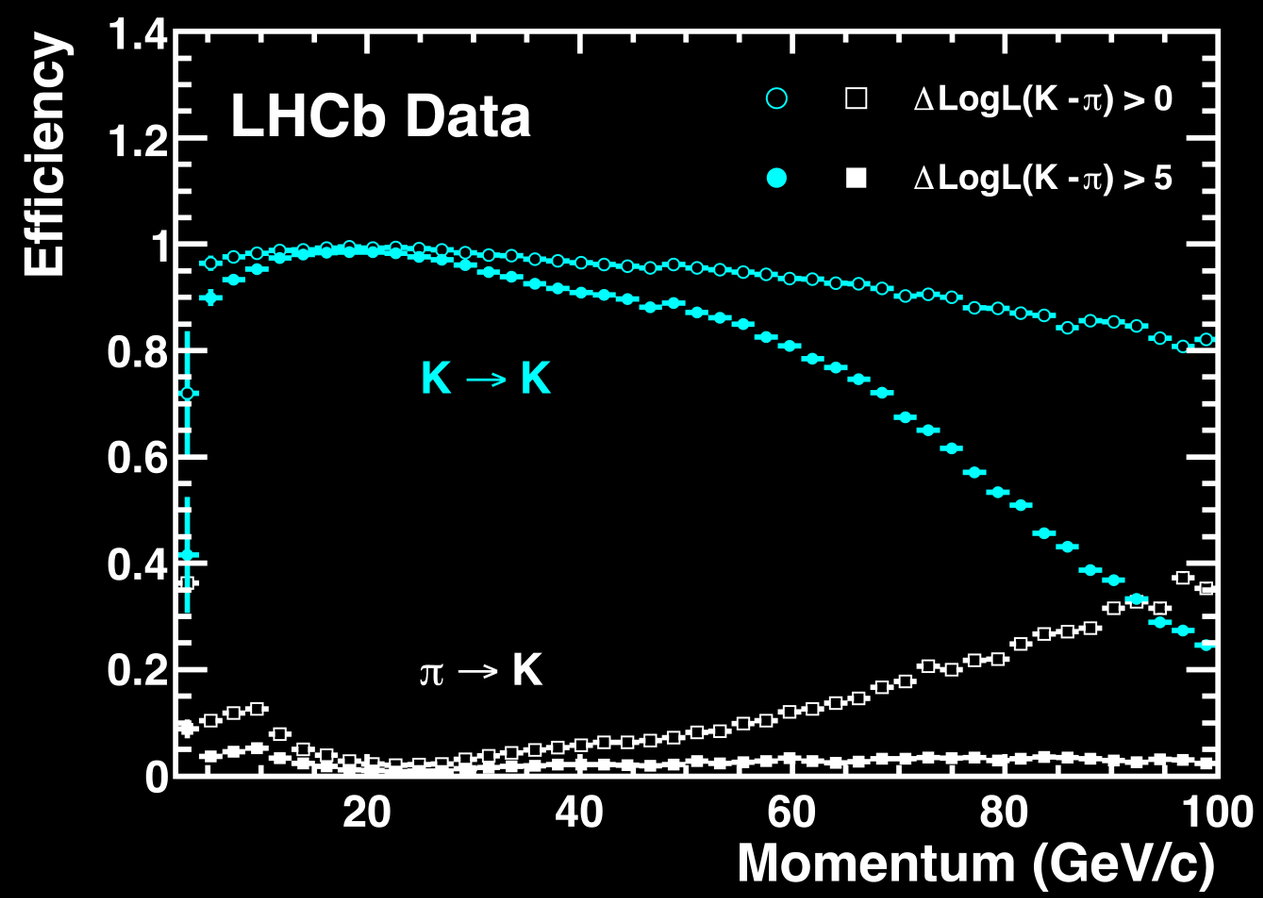


Using electrons from photon conversions and hadrons from D^0 decays, e and h PDFs are constructed from data vs track 3 momentum.

RICH

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

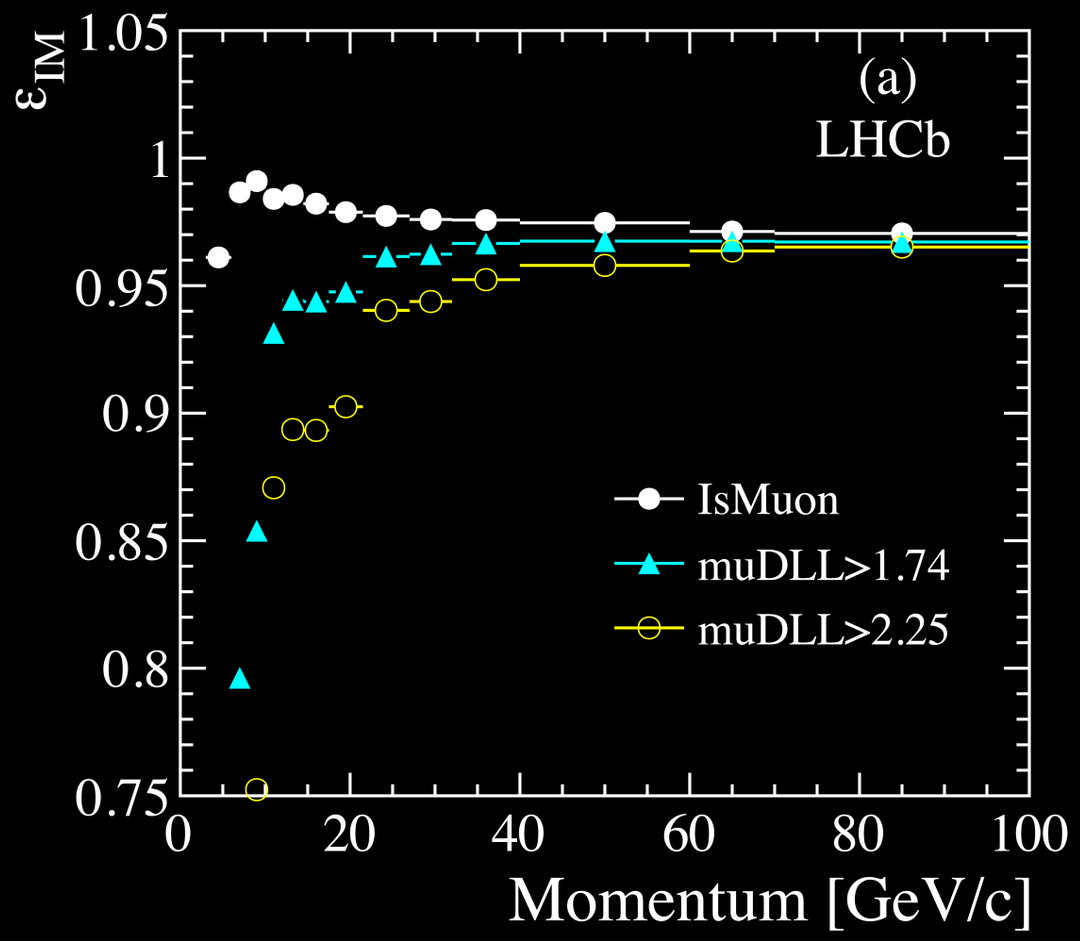


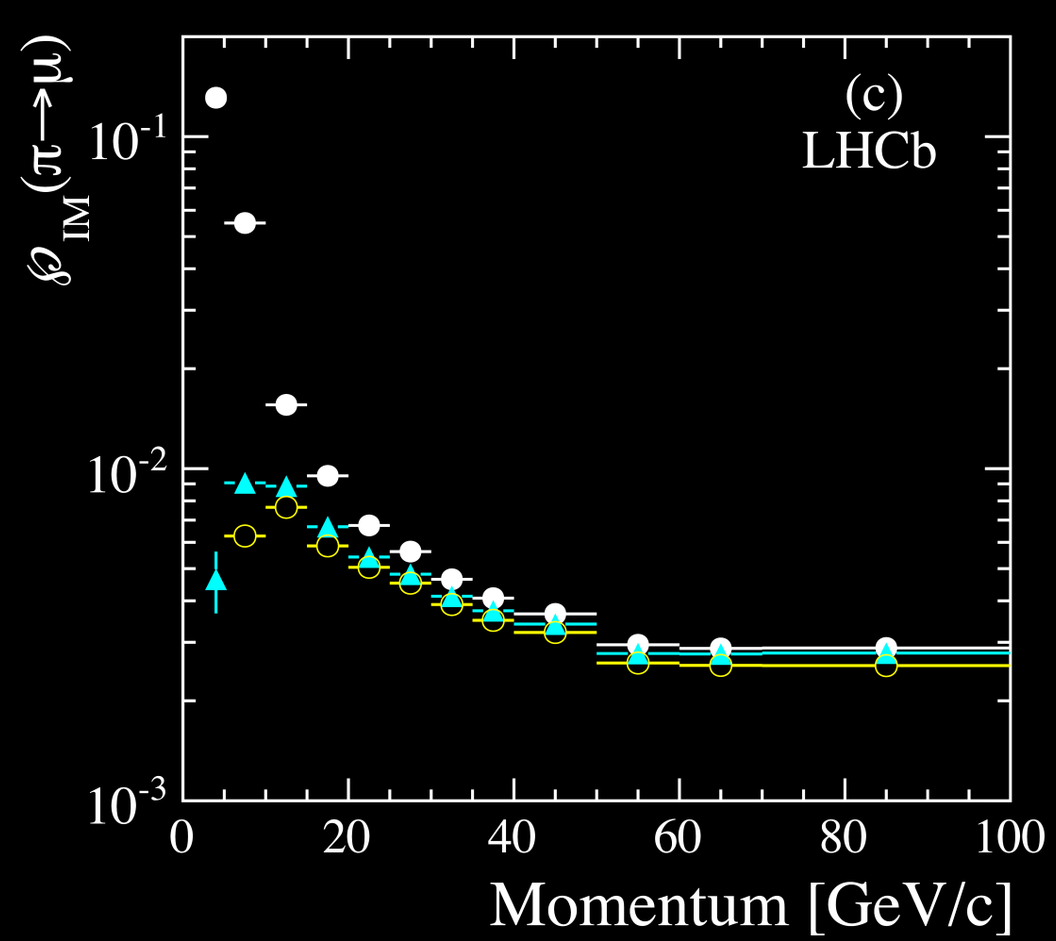


Calculate the likelihood of each RICH ring pattern observed under various PID hypotheses, then use "DLL" to arbitrate (calibrate/validate using $K_S \rightarrow \pi\pi$, $\Lambda \rightarrow p\pi$, and $D^0 \rightarrow K\pi$ data samples).

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.

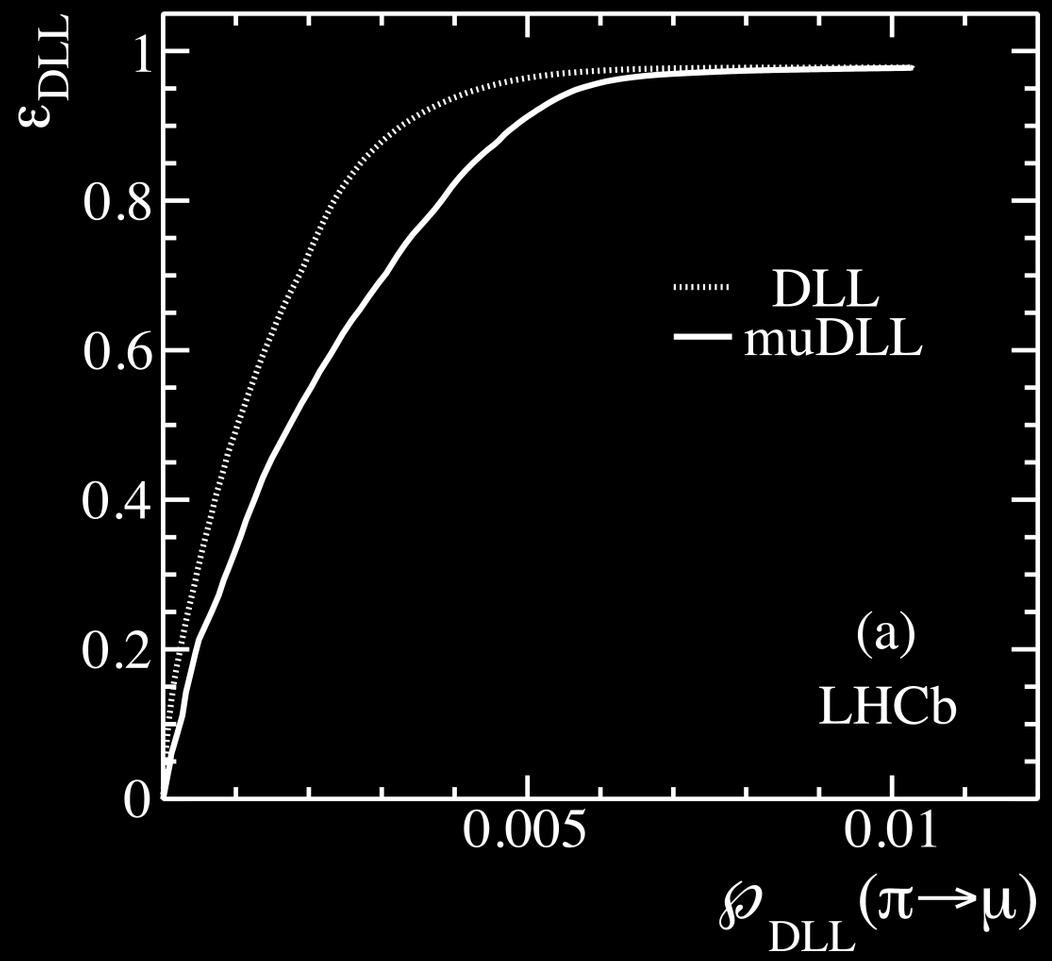


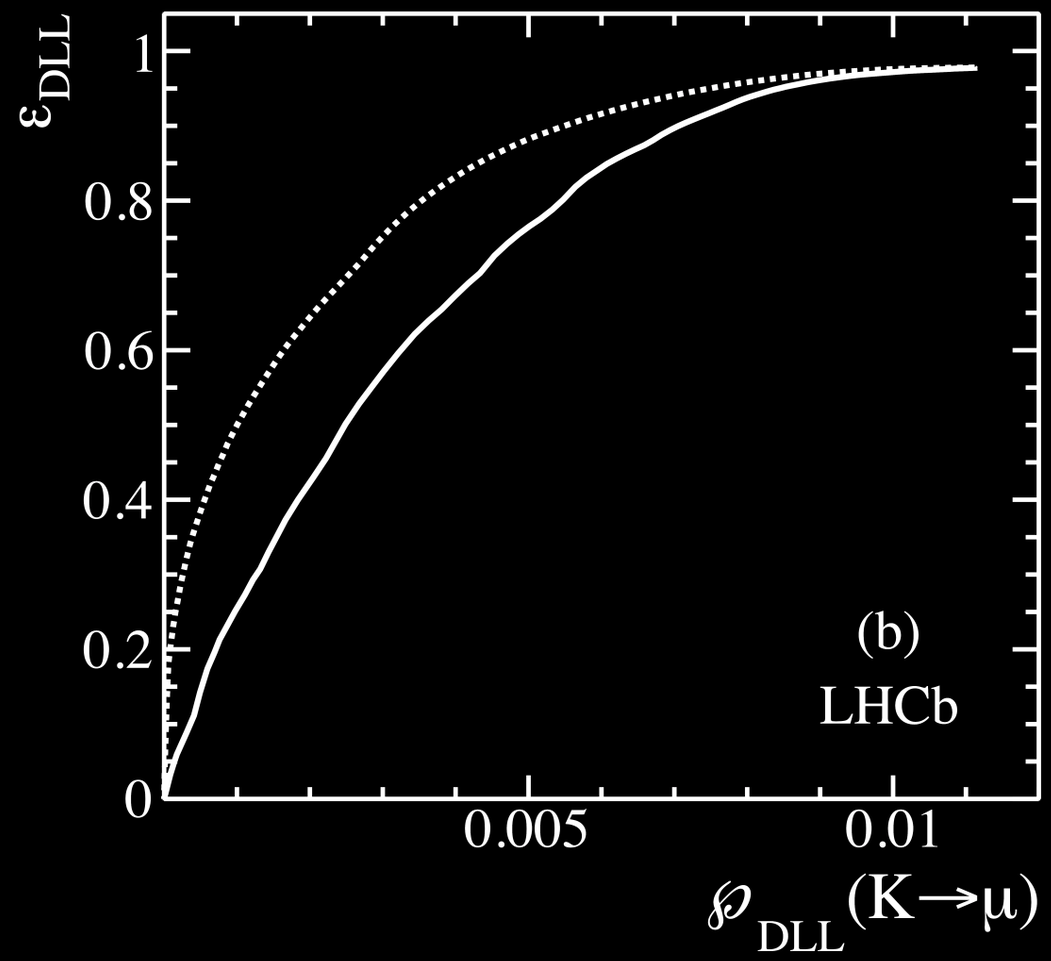


MisID from π, K→μ in flight, shared hits with a real muon, punch through, etc.

Combined DLLs

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

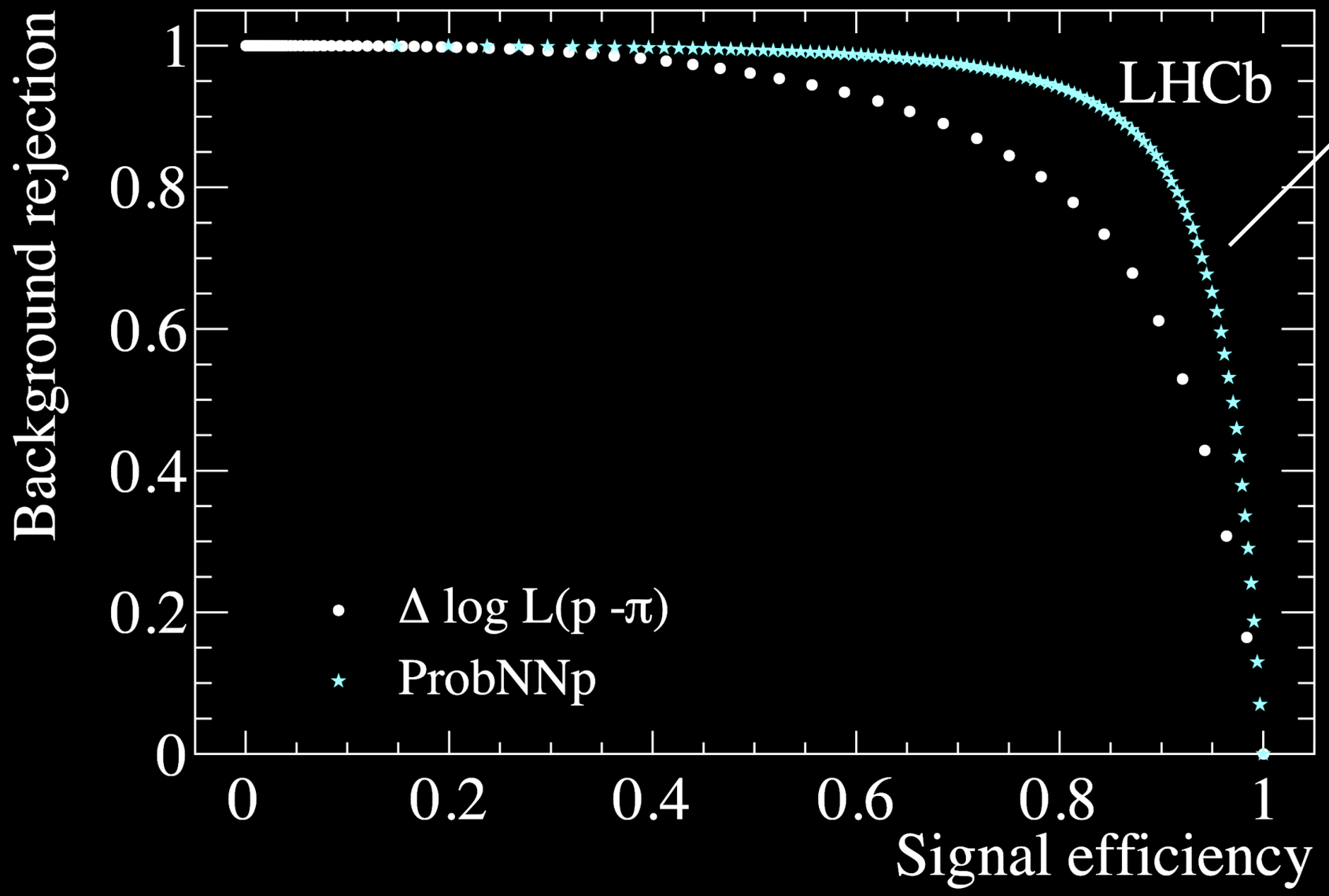




Consider the common case of K→μ decay in flight. If it was still a kaon when it passed through the RICH, then the RICH likelihood will show this. E.g., CombDLL reduces the B→hh misID rate by a factor of 6 for a loss of only 3% of B_s→μμ signal.

Particle ID

Use ML instead to identify particle types: LHCb uses NNs trained on 32 features from all subsystems, each of which is trained to identify a specific particle type.



better

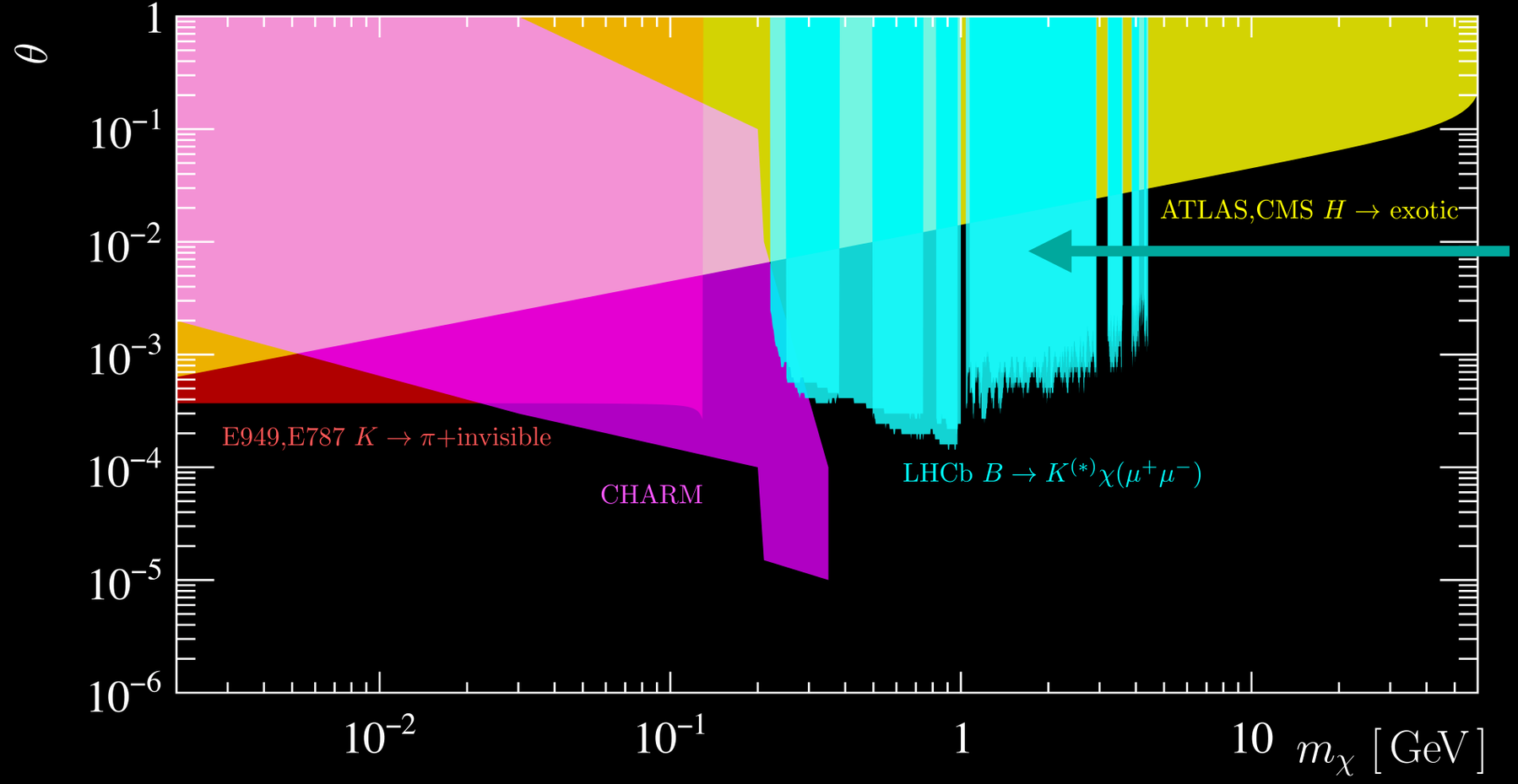
Calibration samples from *standard candles* are used to characterize the performance of the NNs (only assumption is factorization).

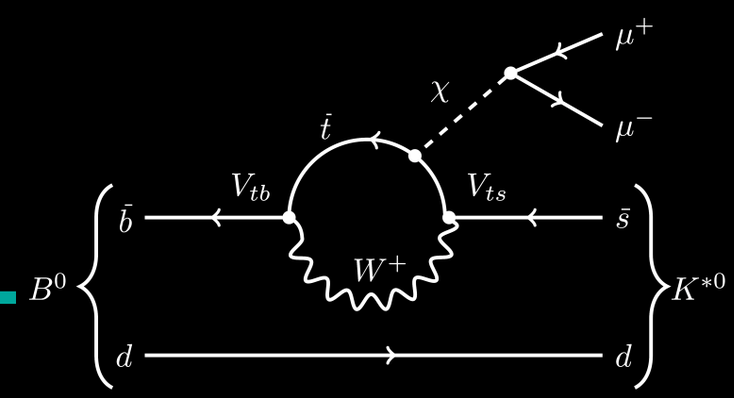
Typically get ~3x less misID background per particle. Currently exploring more advanced algorithms, which can reduce the BKGDs by another ~50%.

Physics-Aware Loss

What about cases where we don't just care about classification error on the training samples? For example, what if we're looking for a new particle whose mass and lifetime are unknown? We cannot generate a MC sample for every mass and lifetime value, so how do we do the search? (We don't want to learn the m and τ of the MC samples we train on.)

LHCb-PAPER-2015-036, LHCb-PAPER-2016-052





LHCb Higgs-portal search covers a factor of 20 in mass, and 4 orders of magnitude in lifetime.

We want to decorrelate the ML response from the features that we don't know. (This is a common problem. Even in 1-D, you want to avoid sculpting fake peaks at the mass values used to generate your training samples. Plenty of SM use cases as well.)

Physics-Aware Loss

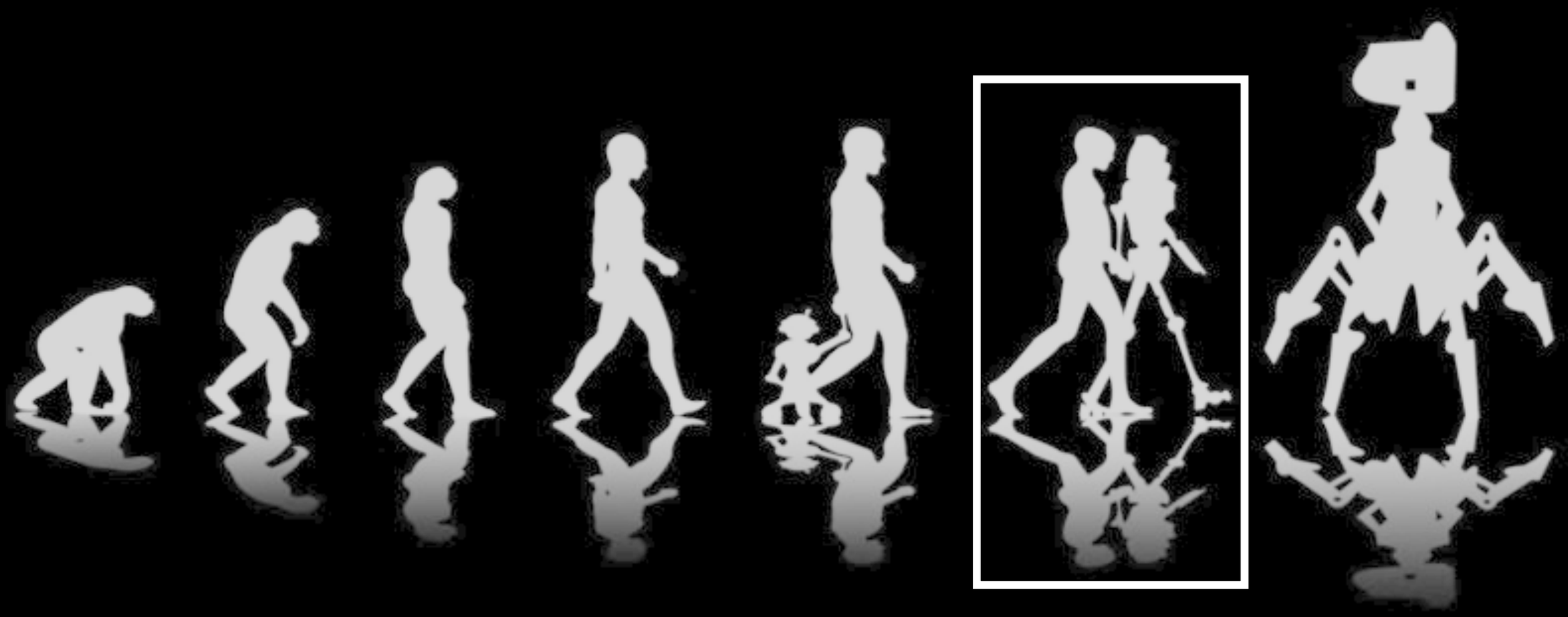
In Rogozhnikova, Bukva, Gligorov, Ustyuzhanin, MW [1410.4140], we redefined the loss function as loss = loss(classificaiton errors) + β loss(flatness), where the latter term is a differentiable function that drives the learning to a result that is independent of some user-chosen features.

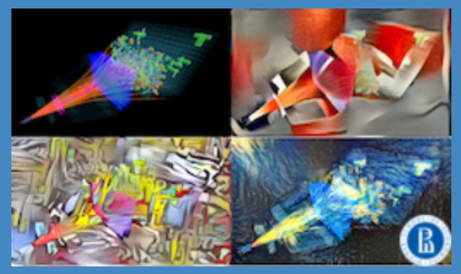
- Generate MC samples at 15 points that roughly span the mass-lifetime plane that we want to explore.
- Use 10 of these for training/validation, and hold back 5 for testing.
- Redefine the loss function to understand that we do not know the true mass and lifetime value of the hidden-sector boson following [1410.4140].
- Find that the uniform BDT performs nearly identically on the 5 samples not used in training as it does on the 10 in the training—and on these 5 samples it does much better than any traditional algorithm tried (essentially, they all learn the masses of the training samples).

What if you don't know how to define a loss function that achieves your goal? Why not just have the machine learn it? (We'll discuss this tomorrow.)

Deep Learning

Tomorrow, we'll look at (at least partially) skipping the feature-engineering step. How well can we do using deeper networks and/or special architectures?





Fourth Machine Learning in High Energy Physics Summer School 2018

6-12 August 2018
University of Oxford
Europe/London timezone

Search...

Support
mlhep2018@yandex.ru

The Fourth Machine Learning summer school organised by Yandex School of Data Analysis, Laboratory of Methods for Big Data Analysis of National Research University Higher School of Economics and University of Oxford will be held in Oxford, UK from 6 to 12 August 2018.

The school will 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 learning and hyperparameter optimisation will be covered with concrete examples and hands-on tutorials. A special data-science competition will be organised within the school to allow participants to get better feeling of real-life ML applications scenarios.

Expected number of students for the school is 50-60 people. The school is aimed at PhD students and postdoctoral researchers, but also open to masters students.

Pre-requisites for participation

- Python programming experience (e.g. http://nbviewer.jupyter.org/gist/rpmuller/5920182, https://www.codecademy.com/tracks/python)
- interest and/or background in HEP
- laptop with WiFi connectivity

Upon completion of the school participants would be able to

- formulate a HEP-related problem in ML-friendly terms;
- select quality criteria for a given problem;
- understand and apply principles of widely-used classification models (e.g. boosting, bagging, BDT, neural networks, etc) to practical cases;
- optimise features and parameters of a given model in efficient way under given restrictions;
- select the best classifier implementation amongst a variety of ML libraries (scikit-learn, xgboost, deep learning libraries, etc);
- understand and apply principles of generative model design;
- define & conduct reproducible data-driven experiments.

Data Science @ LH

9-13 November 2015
CERN
Europe/Zurich timezone
There is a live webcast for this e

- Overview
- Program
- Reading materials
- Speaker List
- Registration
- Participant List
- Videoconference Rooms
- Poster
- Network Connection Request Forms
- CERN ACCESS INFO
- Orientation

Machine Learning for Jet Physics

11-13 December 2017
Lawrence Berkeley National Laboratory
US/Pacific timezone

- Overview
- Scientific Programme
- Call for Abstracts
  - View my Abstracts
  - Submit Abstract
- Timetable
- Contribution List
- Author List
- Book of Abstracts
- Registration
  - Registration Form
- Participant List

Mon 11/12 | Tue 12/12 | Wed 13/12 | All days

Experimental/Practic... | Heavy Ion

Mon 11/12

08:00

Registration
2-100, Lawrence Berkeley Nation

09:00 **Welcome and Logistics**
2-100, Lawrence Berkeley Nation

Jets and ML in Theory
2-100, Lawrence Berkeley Nation

10:00 **Jets and ML in CMS (30'+15')**

Menu

tory

EP@FNAL is the third in a series of workshops dedicated to Data Science in High Ene

2nd IML Machine Learning Workshop

9 Apr 2018, 09:00 → 12 Apr 2018, 20:00 Europe/Zurich

500-1-001 - Main Auditorium (CERN)

Lorenzo Moneta (CERN), Markus Stoye (CERN), Paul Seyfert (Universita & INFN, Milano-Bicocca (IT)), Rudiger Haake (CERN), Steven Randolph Schramm (Universite de Geneve (CH))

Description Inter-experimental Machine Learning Working Group Workshop on Machine Learning will be held between April 9 and 11, 2018. There will also be a full-day hackathon on April 12.

Videoconference Rooms IML-MachineLearning-WG — Join

Registration IML Topical Machine Learning Workshop — 230 / 500 — Register

Participants Aaron White, Abdelwadoud maameri, Abdessamie Chelli, Abdollah Mohammadi, Adam Christopher Elwood, Adriano Di Florio, afifa boubia, Aishik Ghosh, Alessandro Bertolin, Alex Wang, Alexander Melzer

CTD/WIT17

Connecting the Dots — Intelligent Trackers 2017

6th - 9th March 2017, LAL-Orsay, France

Home | Poster

- Program
- Committees
- Registration
- Participants
- contact us
- Lodging
- Travel to Orsay
- Social events

Home

With the parallel progress in pattern recognition algorithms and microelectronic technology, the design and performance of tracking detector is rooted in the solid interplay of hardware and software : sensors, readout and trigger electronics, online and offline reconstruction software. The main focus of the workshop is on pattern recognition and machine learning algorithms devoted to the reconstruction of particle tracks or jets in high energy physics experiments, and the hardware developments that enable them.

This 2017 edition is a merger of the Connecting The Dot series (see CTD2015 Berkeley, CTD2016 Vienna) with the Workshop on Intelligent Tracker series (see WIT2010 Berkeley, WIT2012 Pisa, WIT2014 Penn).

The workshop will be plenary sessions only, with a mix of invited talks and accepted contributions.

Registration are closed

P2IO | Université Paris Sud | Inria | Paris-Saclay Center for Data Science

Learni

nce, Israe

ne Learning and

nd Nails!

oring together particle physicists and machine learning
hallenges posed by high-energy physics data analysis
problems. While some of these problems are simply waiting to be matched with well-established techniques (the pairing of hammers and nails), many require or inspire the

PROGRAM

MINUTES

CONNECTING THE DOTS 2018
4TH INTERNATIONAL WORKSHOP
20-22 MARCH 2018
UNIVERSITY OF WASHINGTON, SEATTLE, USA

Connecting The Dots 2018

20-22 March 2018
University of Washington Seattle
US/Pacific timezone

Search...

- Overview
- Scientific Programme
- Timetable
- Seattle tracking hackathon
  - TrackML hackathon registration
- Registration
- Participant List
- Contribution List
- Videoconference Rooms
- Accommodations
- Transportation

This is a workshop on track reconstruction and other problems in pattern recognition in sparsely sampled data. The workshop is intended to be inclusive across other disciplines wherever similar problems arise. The main focus will be on pattern recognition and machine learning problems that arise e.g. in the reconstruction of particle tracks or jets in high energy physics experiments.

This 2018 edition is the 4th of the Connecting The Dot series (see CTD2015 Berkeley, CTD2016 Vienna, WIT/CTD2017 LAL-Orsay).

The workshop is plenary sessions only, with a mix of invited talks and accepted contributions. There will also be a Poster session.

Wifi is available on site, eduroam credentials, from your institution or CERN, are recommended (but not mandatory).

Follow us on twitter @ctdwit , the official hashtag is #ctd2018 .

Weizmann Institute of Science

Scientific Committee

Kyle Cranmer

slide stolen from K Cranmer

Summary

- For the case where PDFs are known, there is no need for ML — but it's rare to truly be in that situation.
- The use of ML has become ubiquitous in HEP. Many common classification and regression tasks already performed by ML-based algorithms, including in the real-time (trigger) event-classification systems.
- Physics-aware loss functions are a powerful way to use ML in situations where out-of-the-box algorithms fail.
- Deep learning is starting to make an impact, first with HEP problems that are closely related to those commonly solved using DL—but we're now moving towards a *producer* phase (rather than just consumer) in HEP. I'll focus on this tomorrow.
- If you're interesting in learning more about the details and/or software tools, consider attending ML-HEP next year.