Preparing for the future: opportunities for ML in ATLAS & CMS

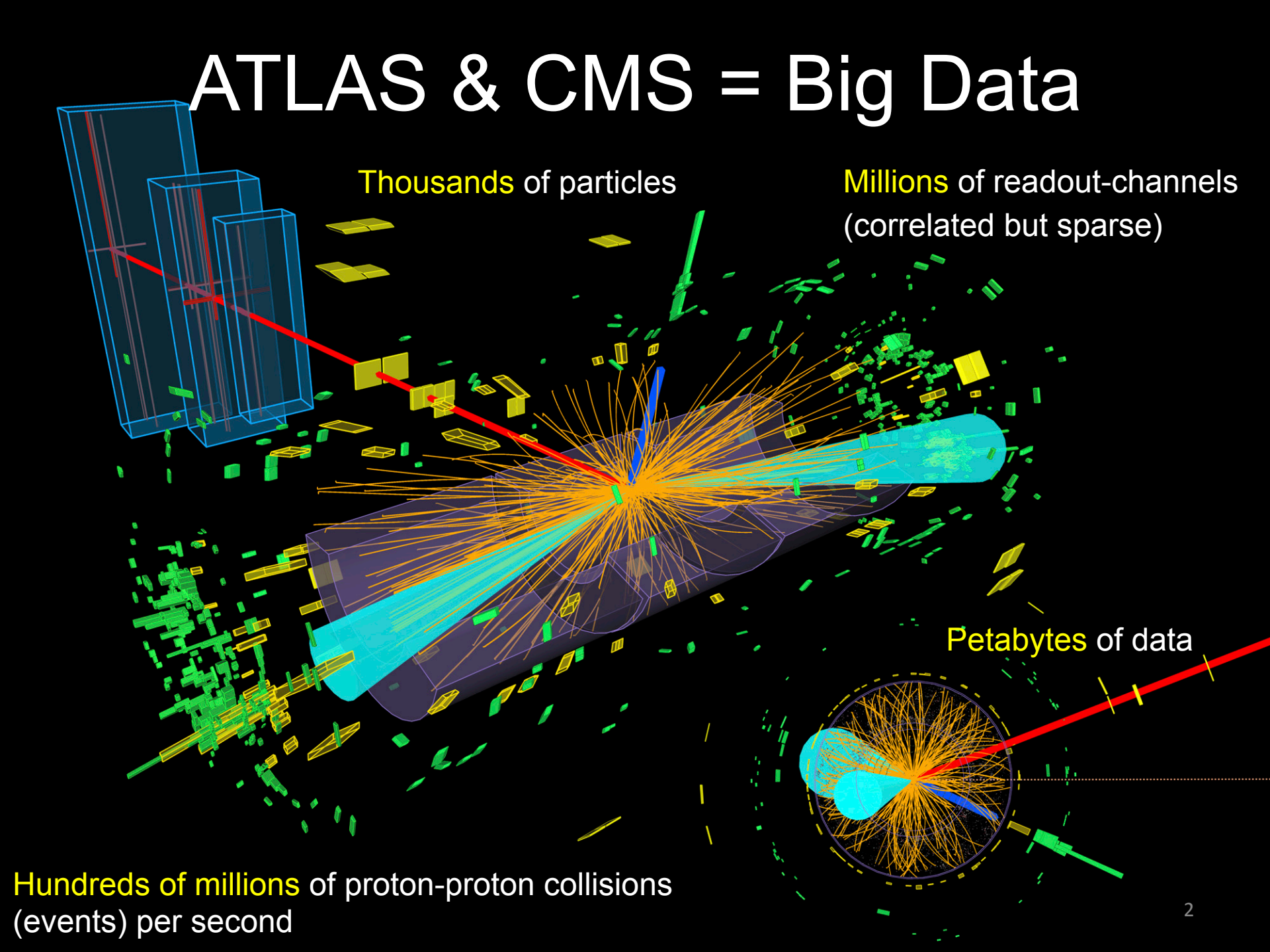
Personal view

Tobias Golling, University of Geneva

Data Science @ LHC 2015 Workshop, November 2015

UNIVERSITÉ DE GENÈVE

ATLAS & CMS = Big Data



Disclaimer

- Not meant to give a complete picture
- Rather give a few representative examples and identify a few key questions
- Target Data Science audience – introduce minimum of physics context needed
- The idea is to leave ample time for discussion

What we care about

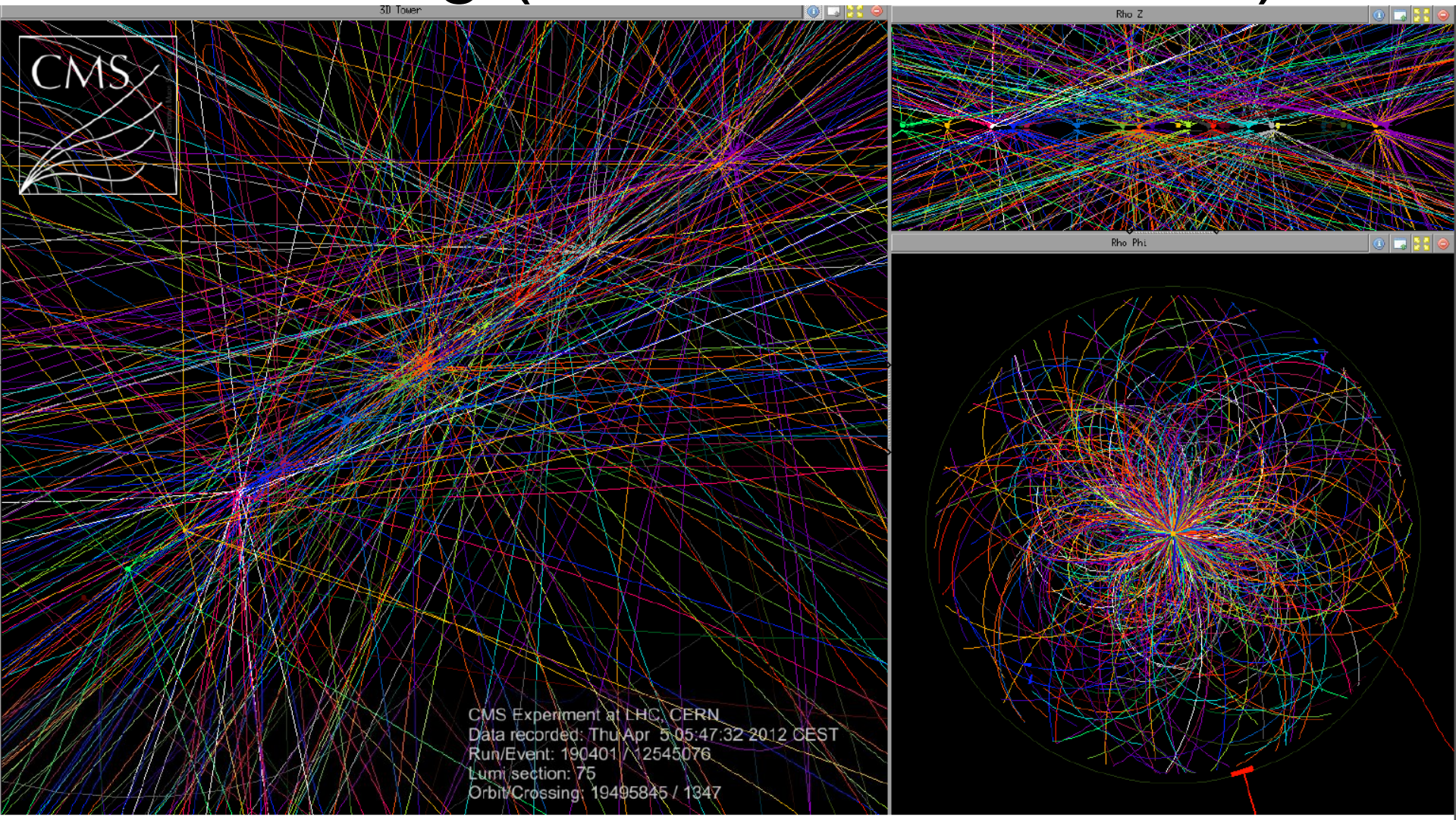
- We are physicists, not data scientists
- We want to focus on physics (mainly)
- We want to make "optimal" use of our data – given some boundary conditions
  - Performance: fully exploit information content in our data
  - Given CPU, memory, statistics (simulation) limitations
  - Minimize "re-inventing the wheel" i.e. make optimal use of person-power, exploit synergies: applicability of one ML solution to another related HEP problem (b-jets ⇒ boosted objects)
    - What tools/approach for what problem?
    - Learn how to tune, validate and troubleshoot tool
  - Complexity, tunability, robustness
  - Gain understanding: are there features we haven't used yet
  - Use state-of-the-art tools: stay connected with ML community

Typical HEP problems

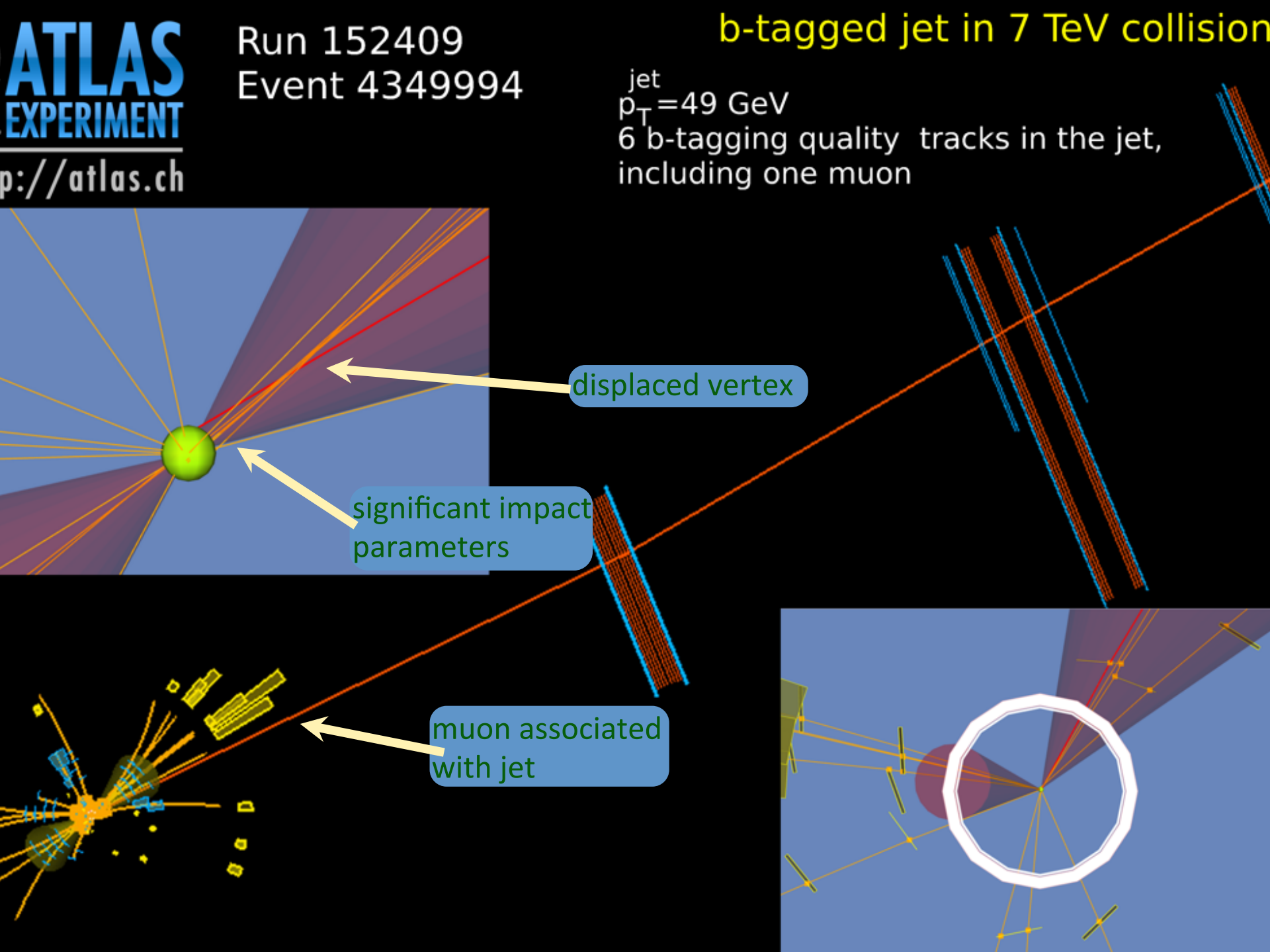
- Particle finding, reconstruction, classification
  - B-jet ID (BDT, NN, ATL-PHYS-PUB-2015-022), tau ID (BDT, arxiv:1412.7086), electron ID (LH, ATLAS-CONF-2014-032), …
- Event classification (see Mauro's talk)
- Automatic fault detection
- Regression problems (see Mauro's talk)
  - Electron and photon energy calibration using BDT regression (arxiv:1407.5063)

Particle classification: b-jet identification

Tracking (from Vincenzo's talk)

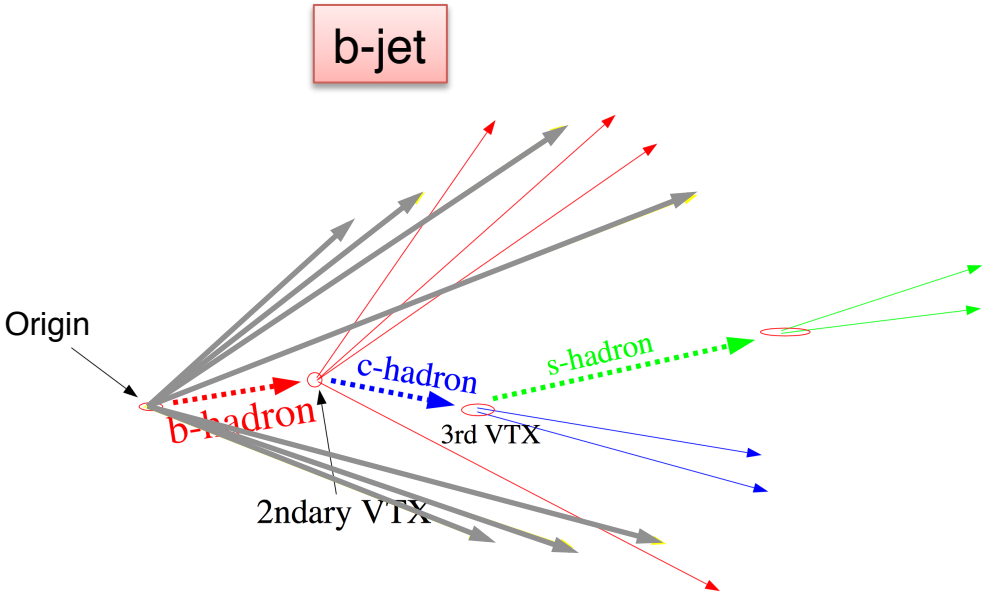


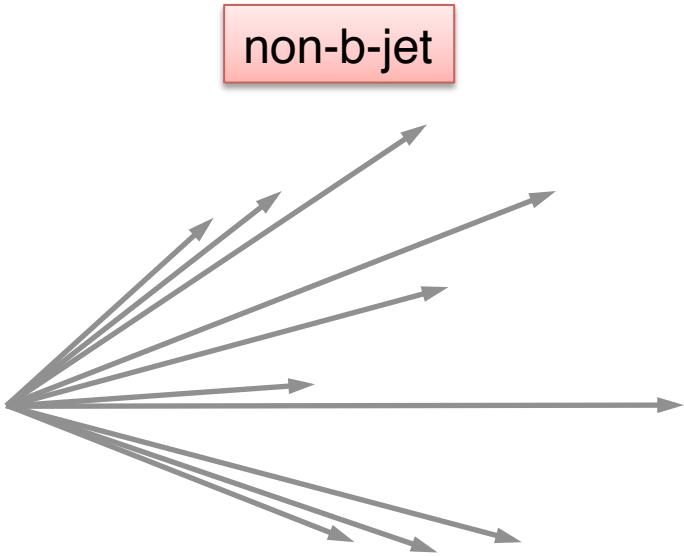
Not just identification and classification,
also precise parameter determination


ATLAS
EXPERIMENT
p://atlas.ch
Run 152409
Event 4349994
b-tagged jet in 7 TeV collision
jet
p_T=49 GeV
6 b-tagging quality tracks in the jet,
including one muon
displaced vertex
significant impact parameters
muon associated with jet

Example: b-jet Identification

- Compression of information: hits → tracks → jet-based quantities, e.g.:
  - Displaced vertex
  - b probability of jet as product of b track probabilities
- Inputs: we currently use jet-based quantities
- Output: b-jet or not using NN & BDT: mainly TMVA, also test AGILEPack (C++ framework for deep learning designed for HEP purposes by Luke de Oliveira: https://github.com/lukedeo/AGILEPack)

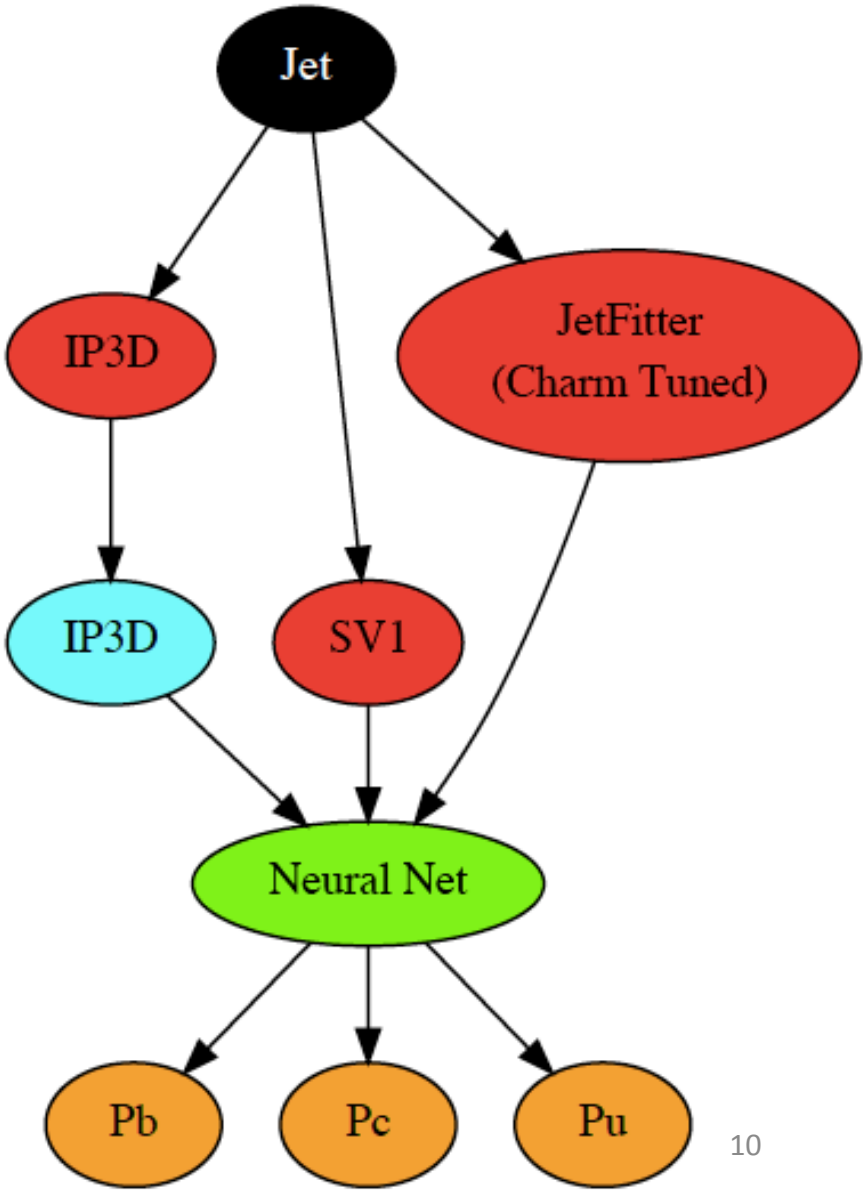




ATLAS-CONF-2011-102, ATL-PHYS-PUB-2015-022, ATLAS-CONF-2014-046

Example: charm-jet Identification

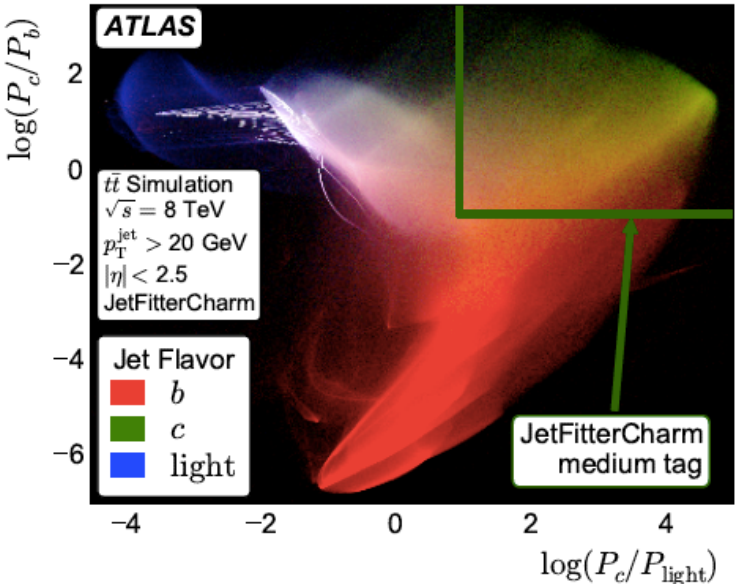
- Based on the same main idea: “soft lepton" and “lifetime"
- More difficult than b-jet identification – charm-jets are “between" light and b in many distributions
  - Shorter decay lengths for charm-jets
  - Fewer tracks than b-jets, hard to resolve displaced vertex
- JetFitterCharm (b-tagging retuned)
  - Looser track selection
  - New variables
  - Used in 2 analysis [arxiv: 1501.01325, arxiv:1407.0608]

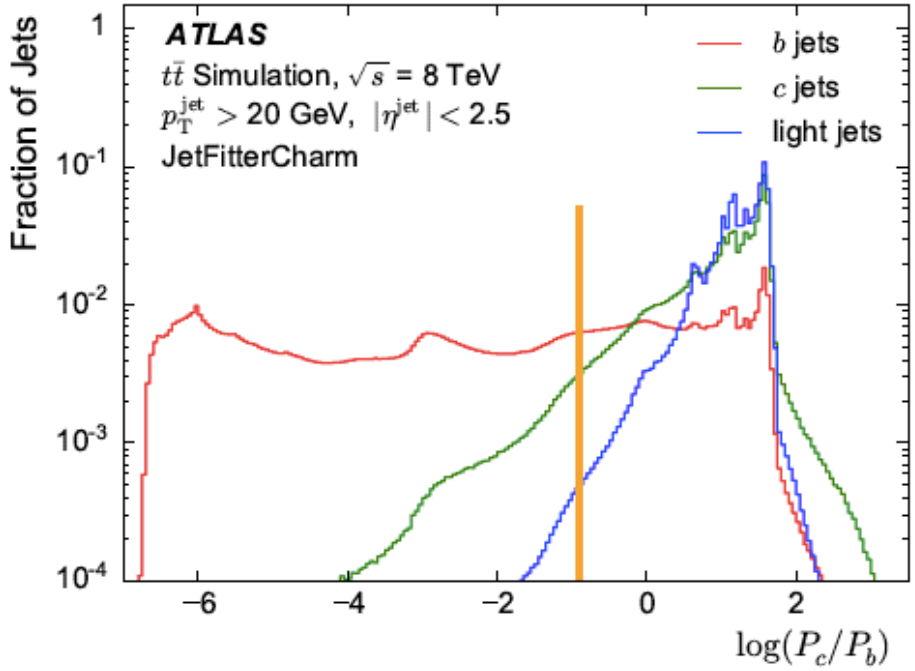


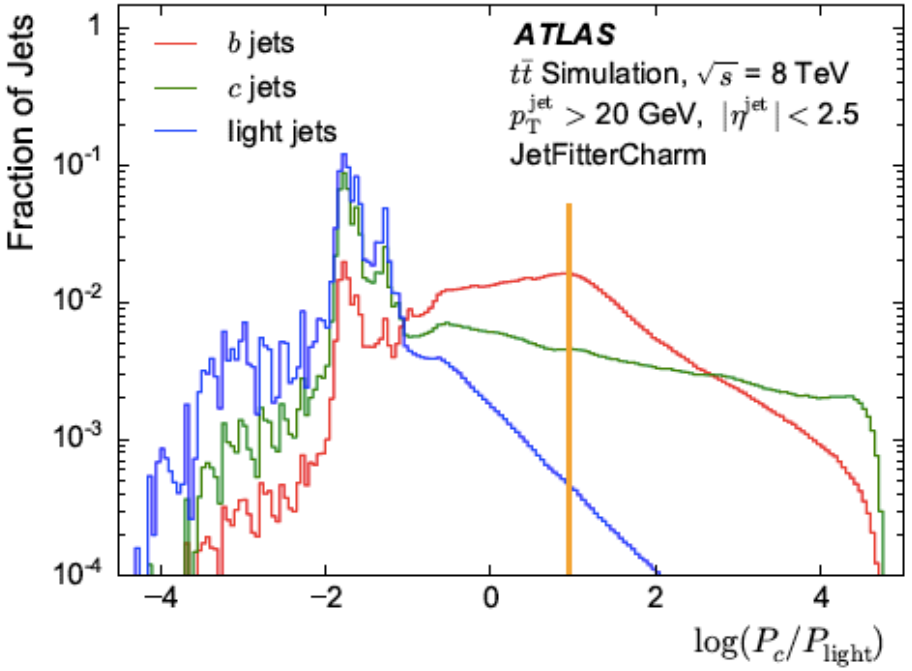
Example: charm-jet Identification

- Define 2 discriminants based on 3 NN outputs:

$$\text{anti-}b \equiv \frac{P_c}{P_b} \qquad \text{anti-light} \equiv \frac{P_c}{P_{\text{light}}}$$





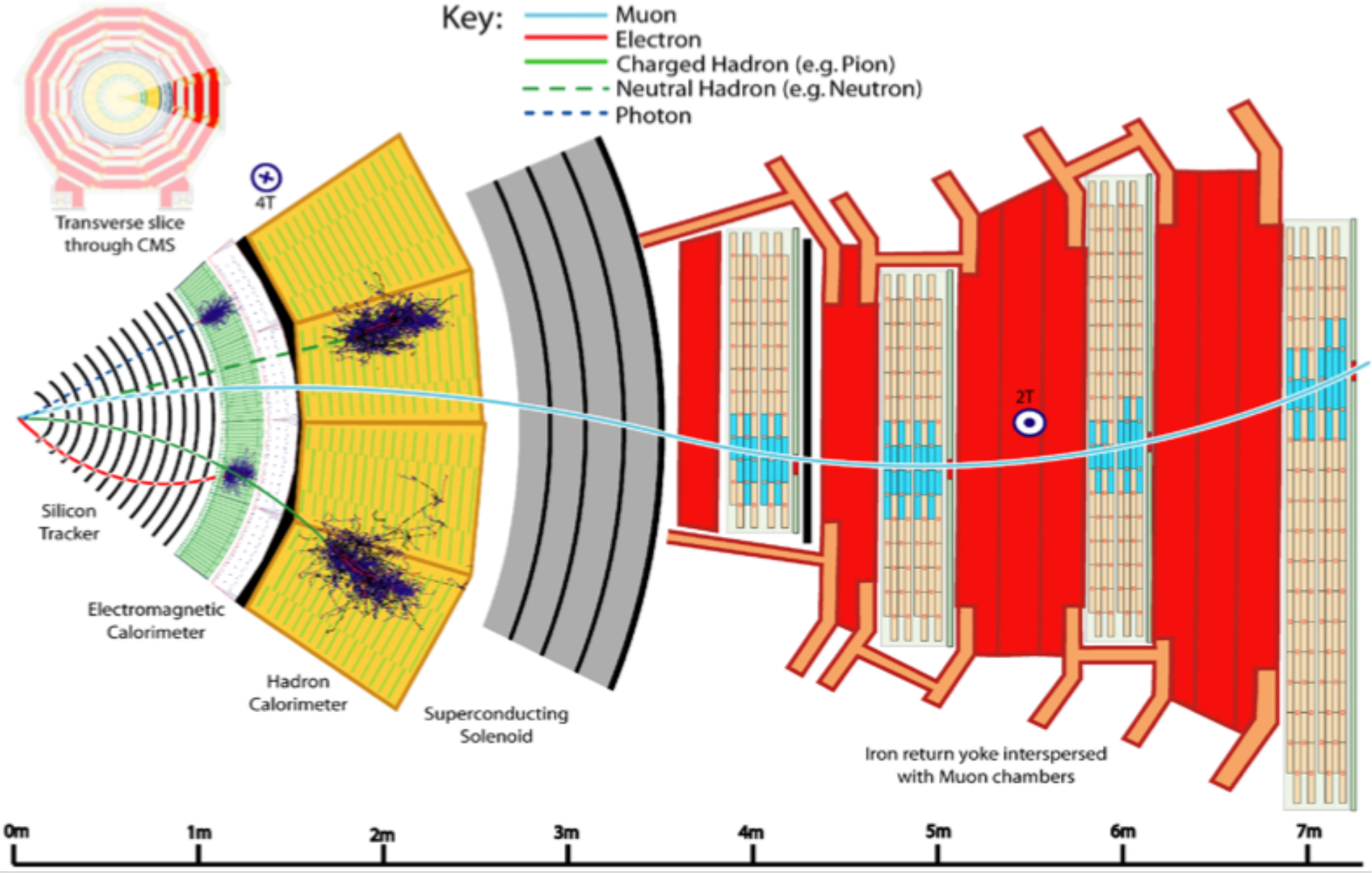


Interesting follow-up questions (Example: b-jets and charm-jets)

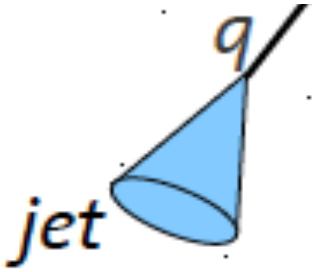
- Have we identified the best observables (minimally correlated, provide best predictive power)
- How to find / construct new observables (capture full feature space)
- Is it beneficial to go to lower-level input variables: track-based (variable length)
- How to go from a multi-class output (b, charm, light) back to individually tuned 2-class outputs (1 signal vs 2 backgrounds) given analysis needs
- Just began to apply Deep Learning

Particle classification: boosted objects

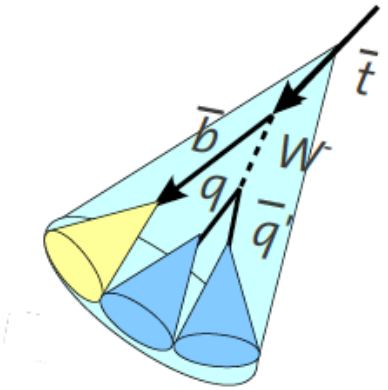
Example: boosted objects



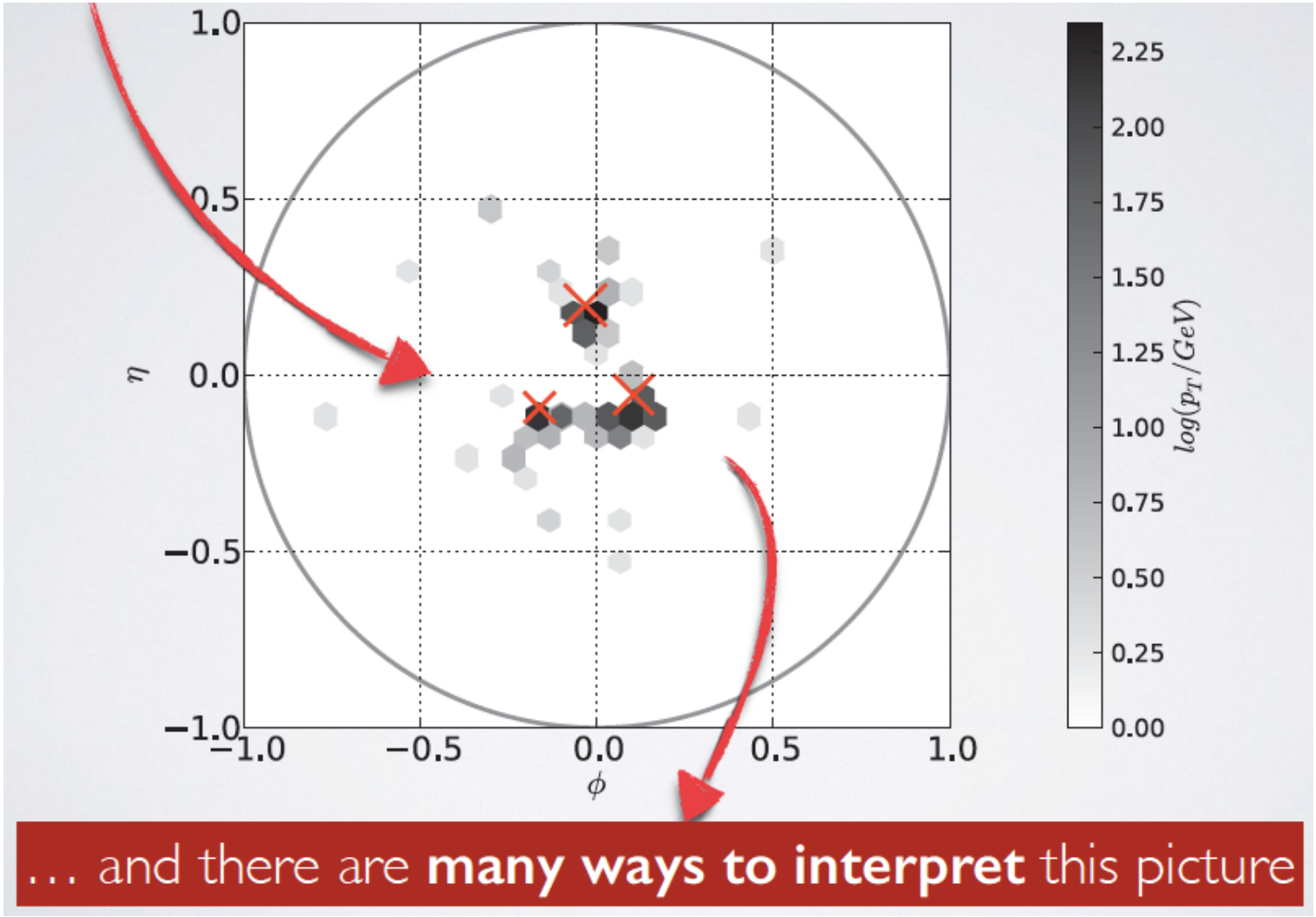
Identifying boosted objects



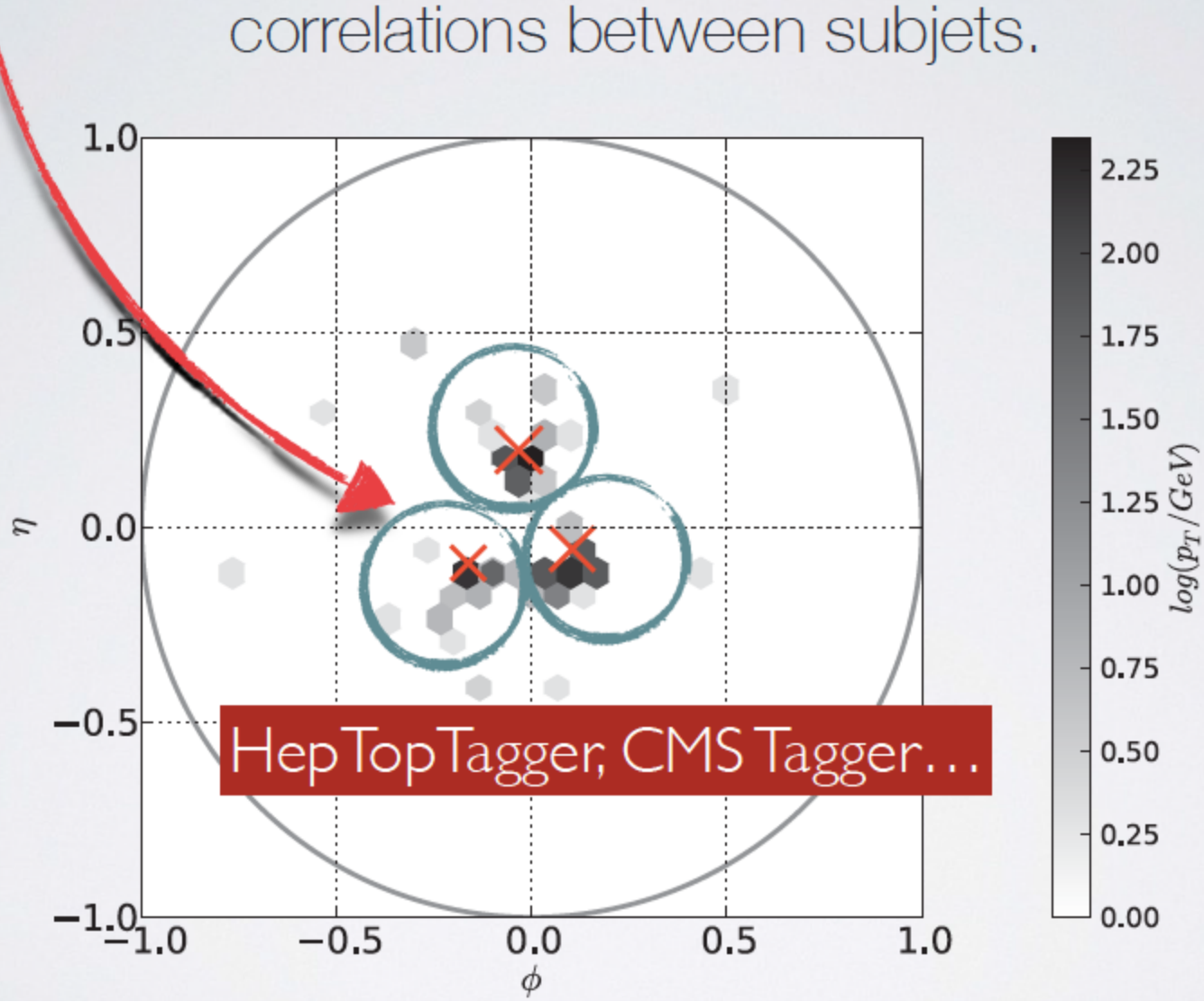
vs



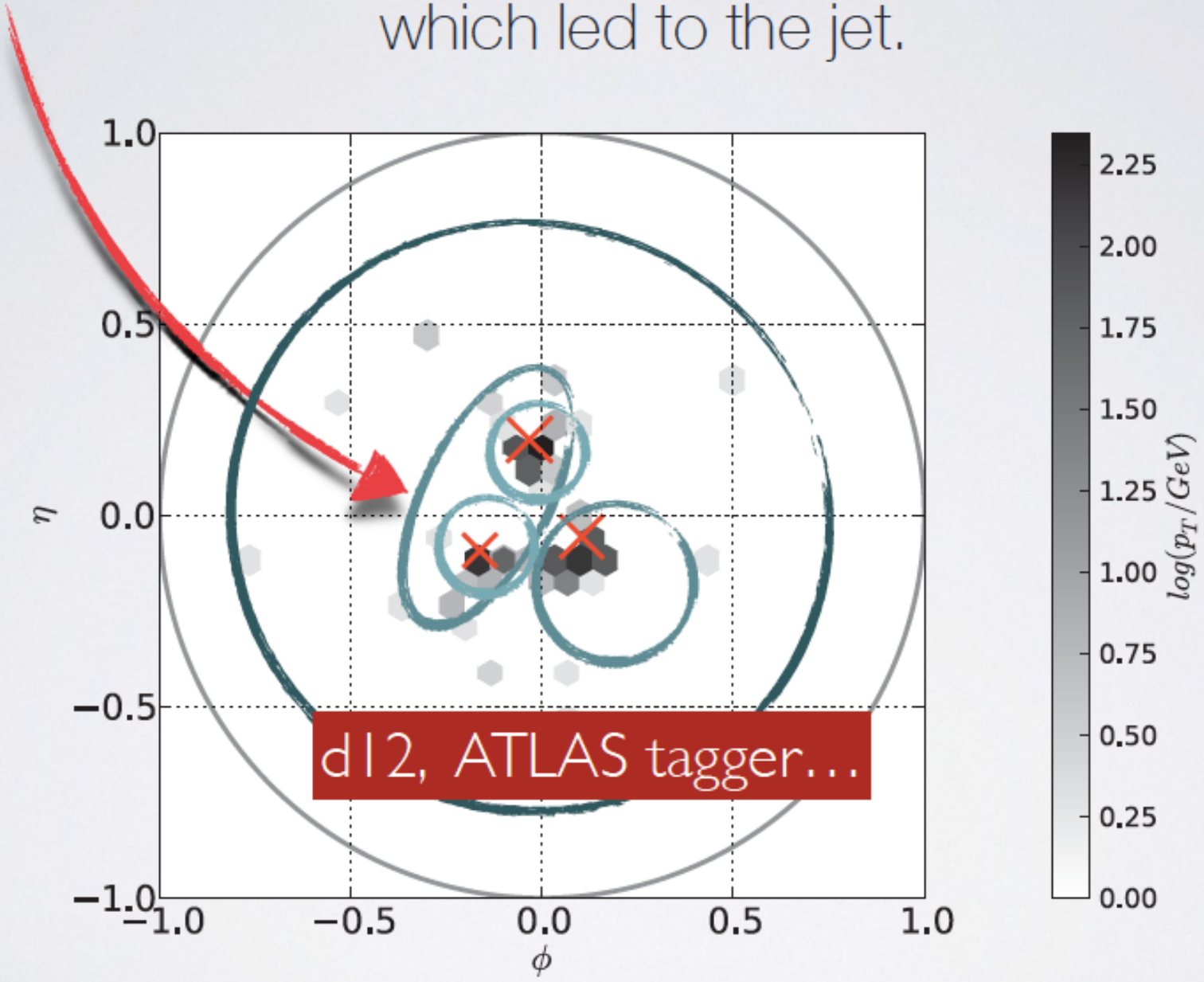
Identifying boosted objects



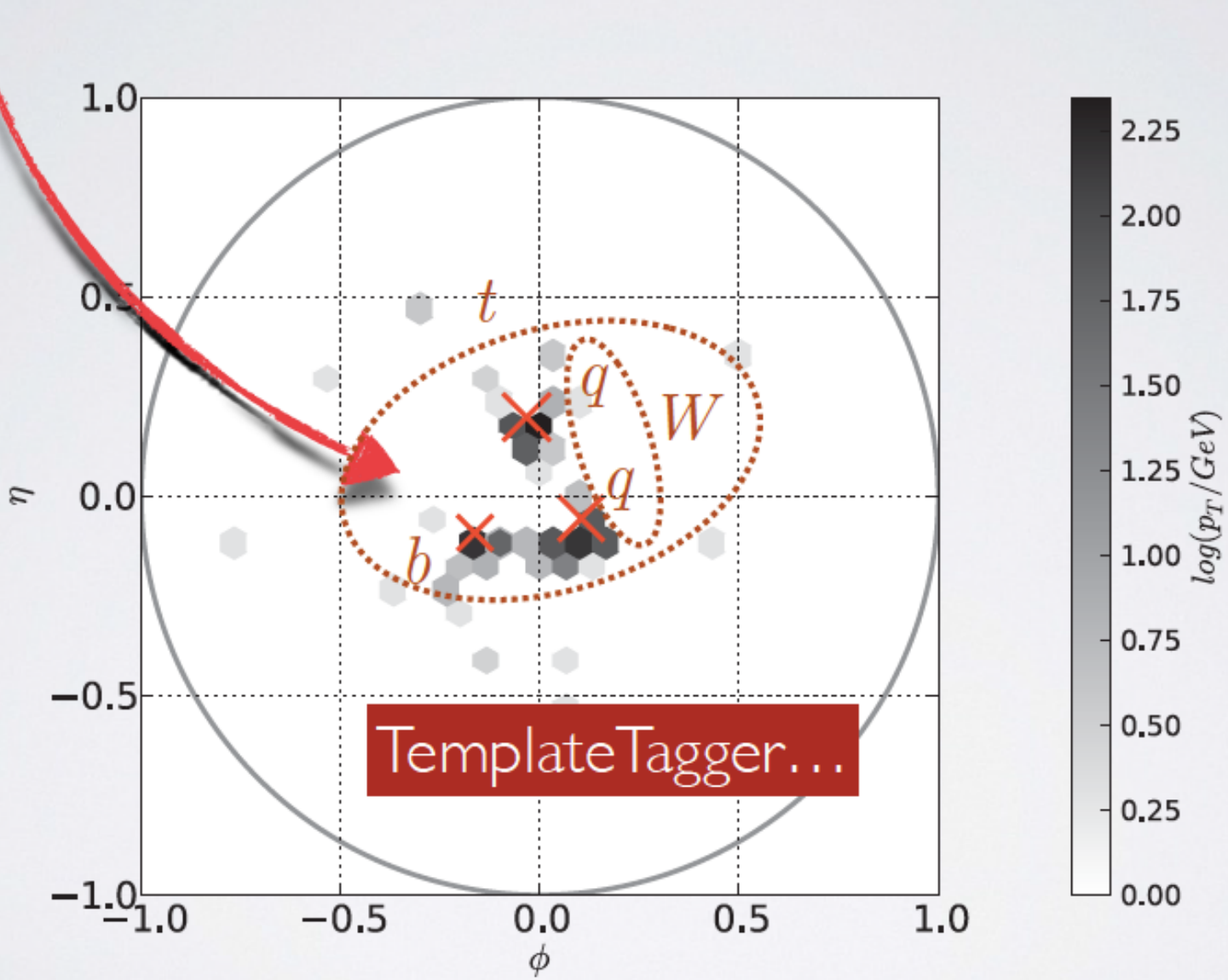
Subjets - recluster the event with a smaller cone and exploit correlations between subjets.



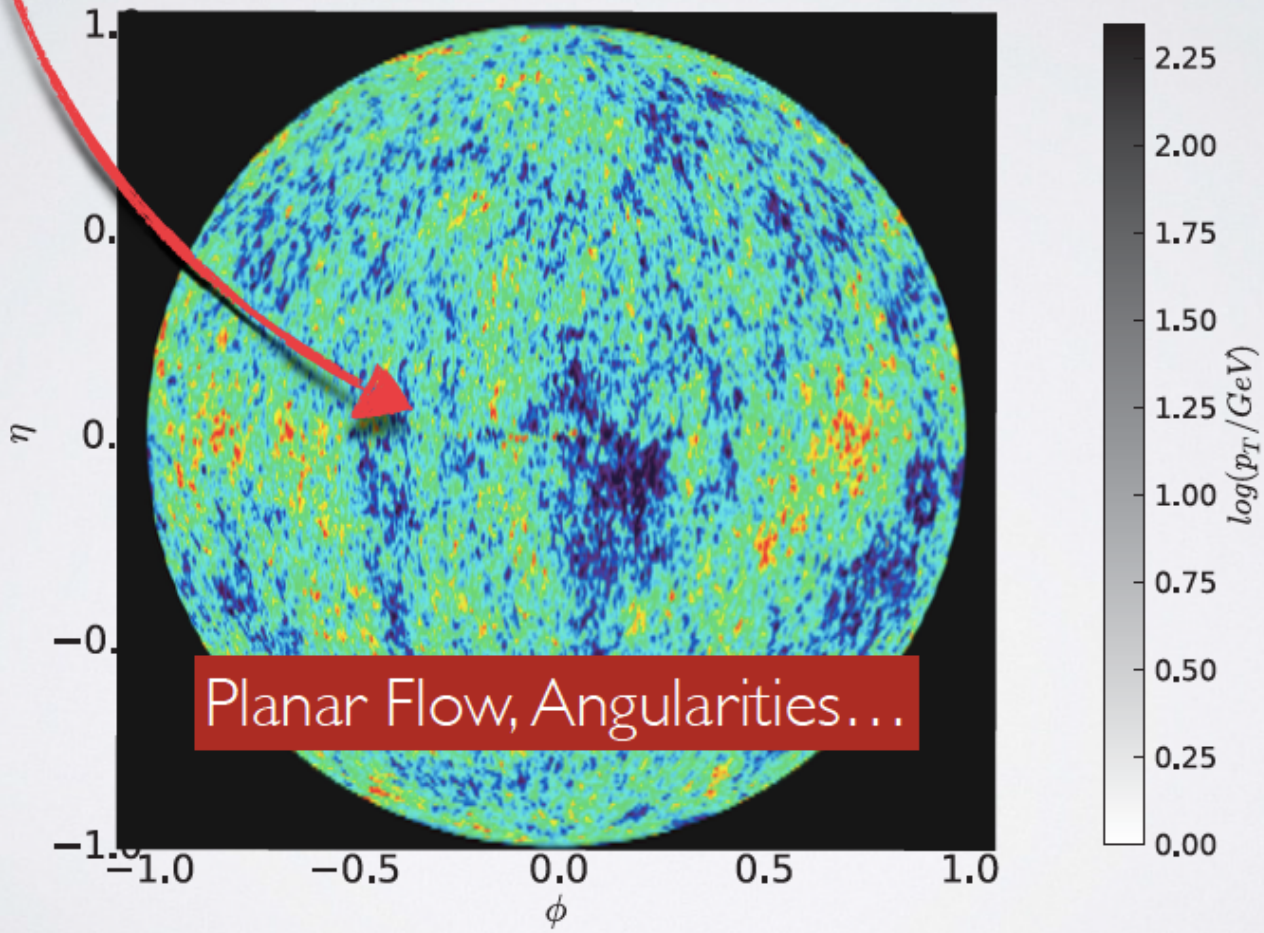
Clustering history- exploit the **differences in steps** which led to the jet.



Partons - Interpret the jet as a **partonic structure** with kinematic properties of some heavy boosted object.



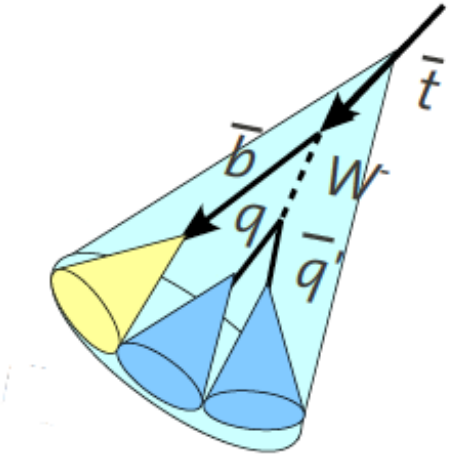
Energy distribution - the picture is essentially some distribution $f(\eta, \phi)$. Look at the moments of the distribution



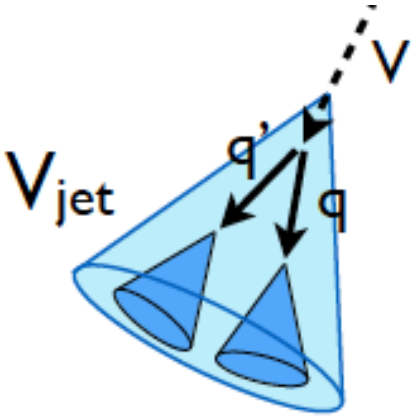
Identifying boosted objects

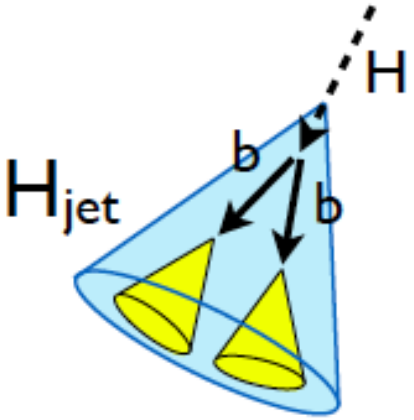
- Essentially image recognition problem
- 3D jet-image
- Calorimeter cells ~ pixels in a camera (use all available information for jet classification)
- Use computer vision classification algorithms (facial recognition)
- One difference: we have a model for the image – no model to recognize Brad Pitt's face!
- NN [arXiv:1501.05968]
- Fisher discriminant with pre-processing [arxiv:1407.5675]
  - Use subjets to align images: like eyes in a face
  - Make use of symmetries: center, rotate, translate

Multi-class problem: various boosted objects

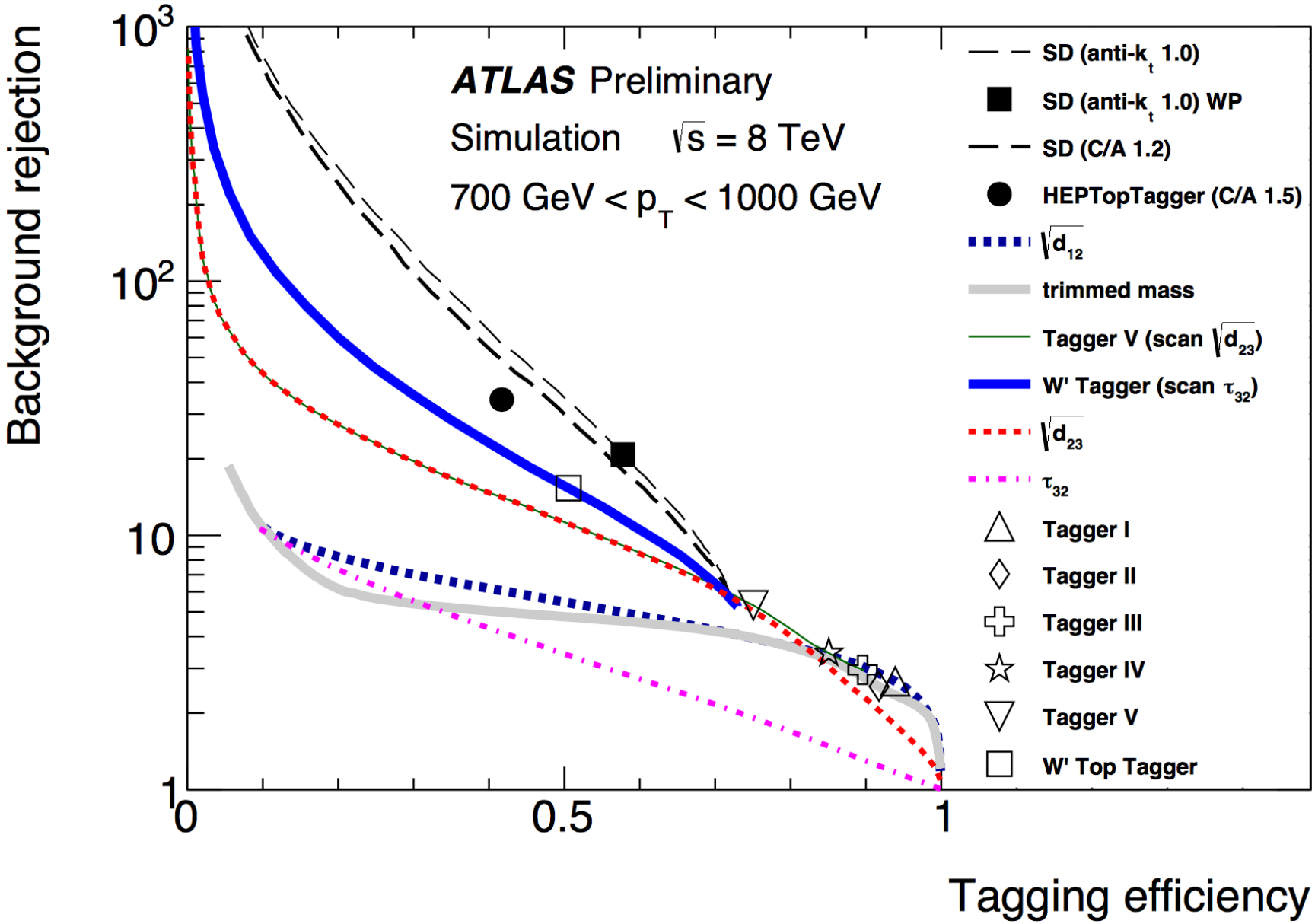


vs



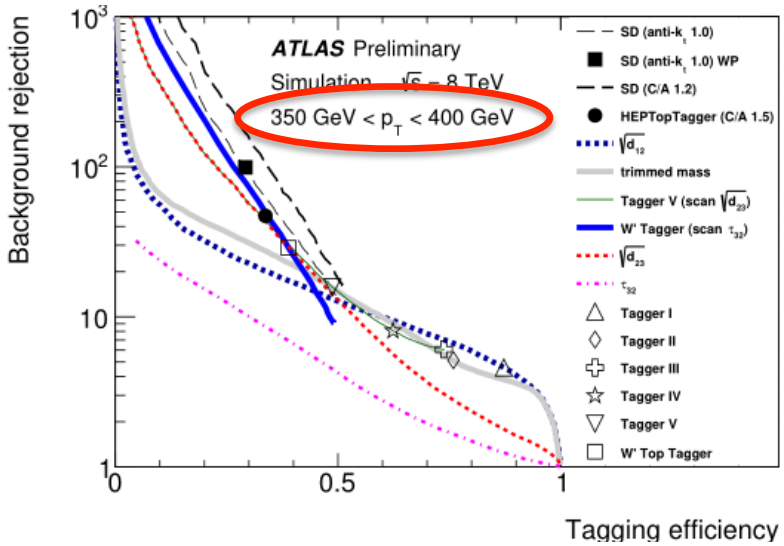


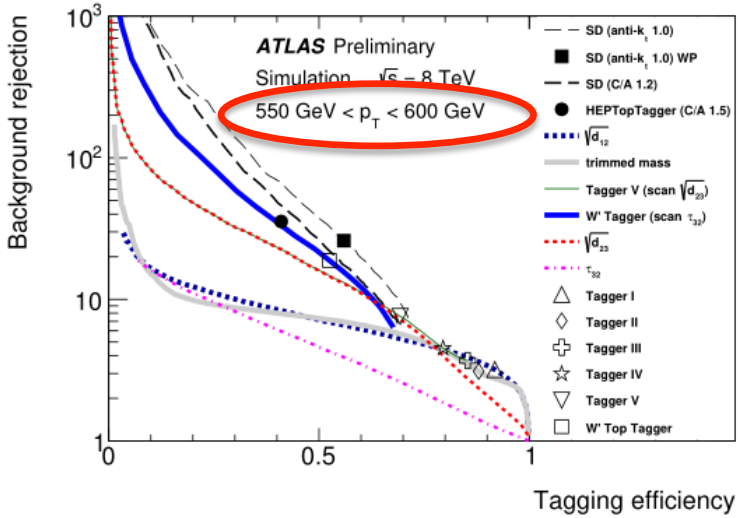
Boosted objects: figure of merit

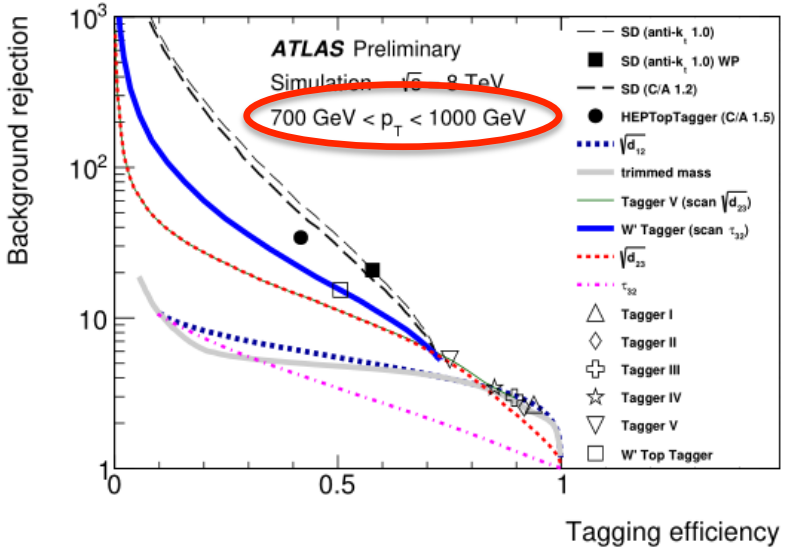


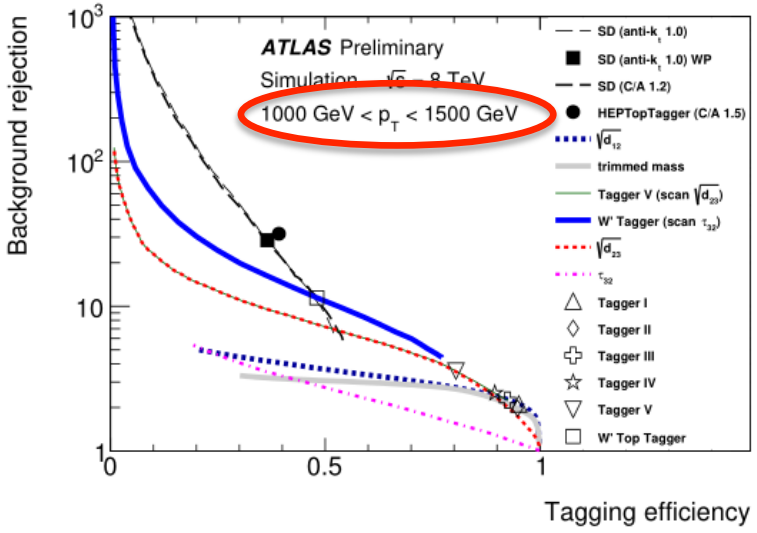
ATLAS-CONF-2015-036

Boosted objects: figure of merit vs pT









ATLAS-CONF-2015-036

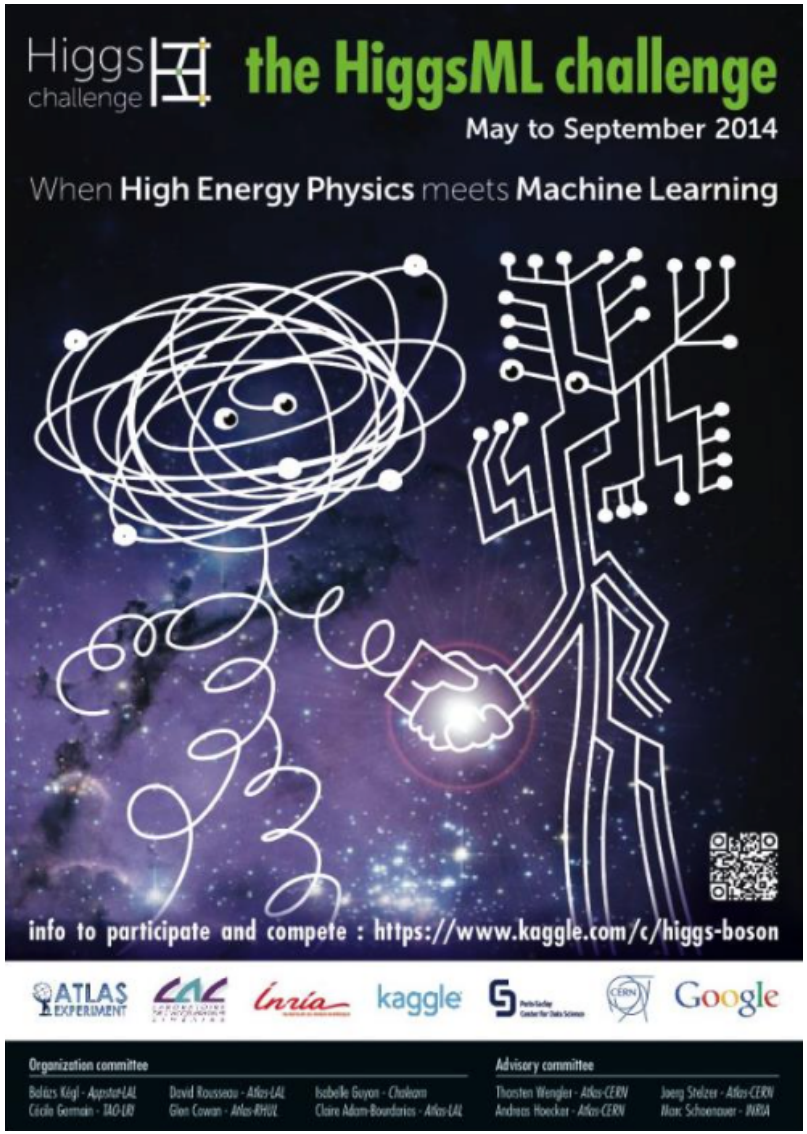
e.g. best performance for all p_T vs. best performance for high p_T only

Interesting follow-up questions (Example: boosted objects)

- What are the upper bounds on discrimination
  - For a detector with perfect resolution (informing future detector designs)
- Are there higher-level features not yet captured in our observables? [arxiv:1407.5675]
  - Using huge amounts of statistics: finely binning in all known features and see if there are further features

HEP Machine Learning Challenges

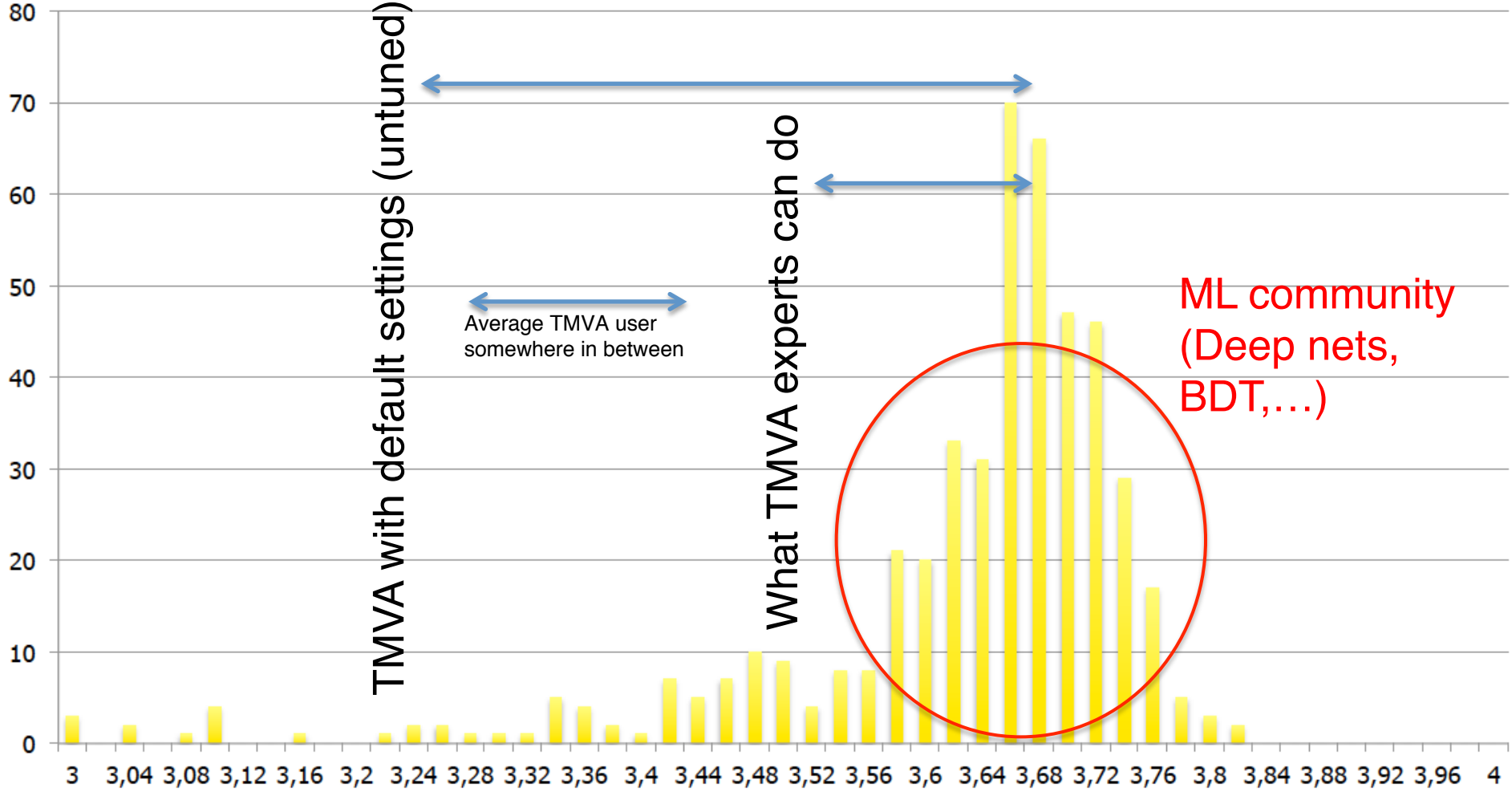
Higgs ML Challenge



- Big success !
- 1785 teams (1942 people) have participated
- 6517 people have downloaded the data
- Most popular challenge on the Kaggle platform (until spring 2015)
- 35772 solutions uploaded
- 136 forum topics with 1100 posts

- Similar challenge by LHCb

Solidifying Case for ML for HEP



20-40% more data needed to get the same improvement

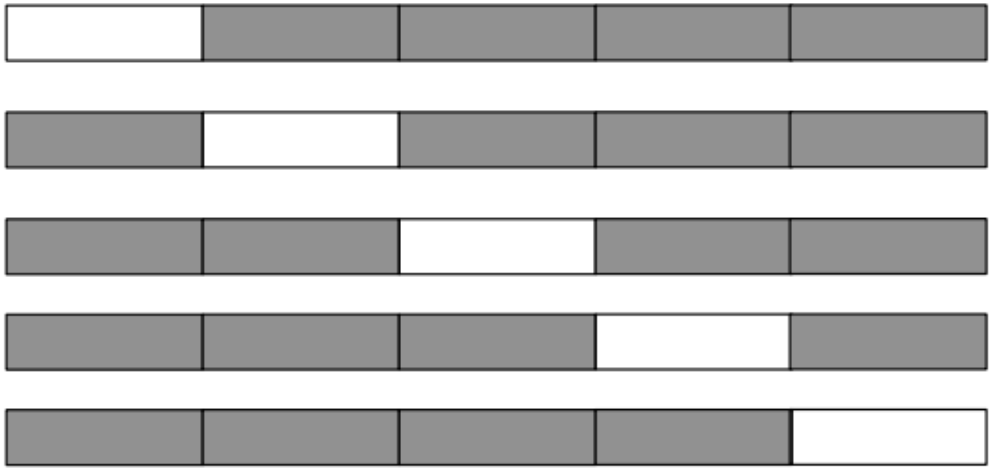
Interesting follow-up questions (Example: Higgs ML challenge)

- Learned a lot on interaction with ML community
  - Problem not too easy / not too complicated
  - Definition of figure of merit
  - Long-term planning: follow-up after challenge
  - Goal: just want a solution or reuse code
- K-fold cross validation to measure performance independently of training

K-fold cross validation

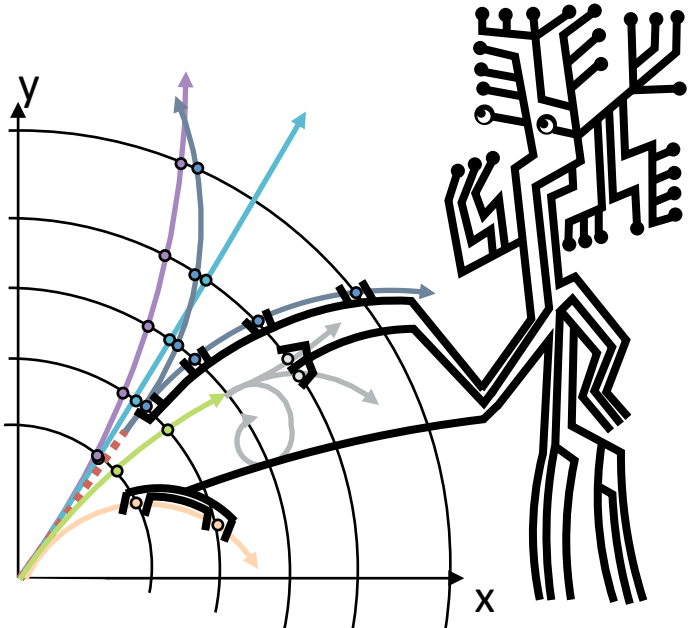
1) Reserve a test sample from the data $\Omega \rightarrow \Omega' \subset \Omega$ (if one wants to validate generalisation beyond the *k*-fold cross validation step).
2) Randomly split the remaining data into k sub samples:
$$\Omega' \rightarrow \Omega_i, i = 1, 2, \ldots k.$$
3) Cycle through training *k* times, each time leaving one sub-sample out.

e.g. 5-fold cross validation: train 5 times dropping out one sub-sample at a time:



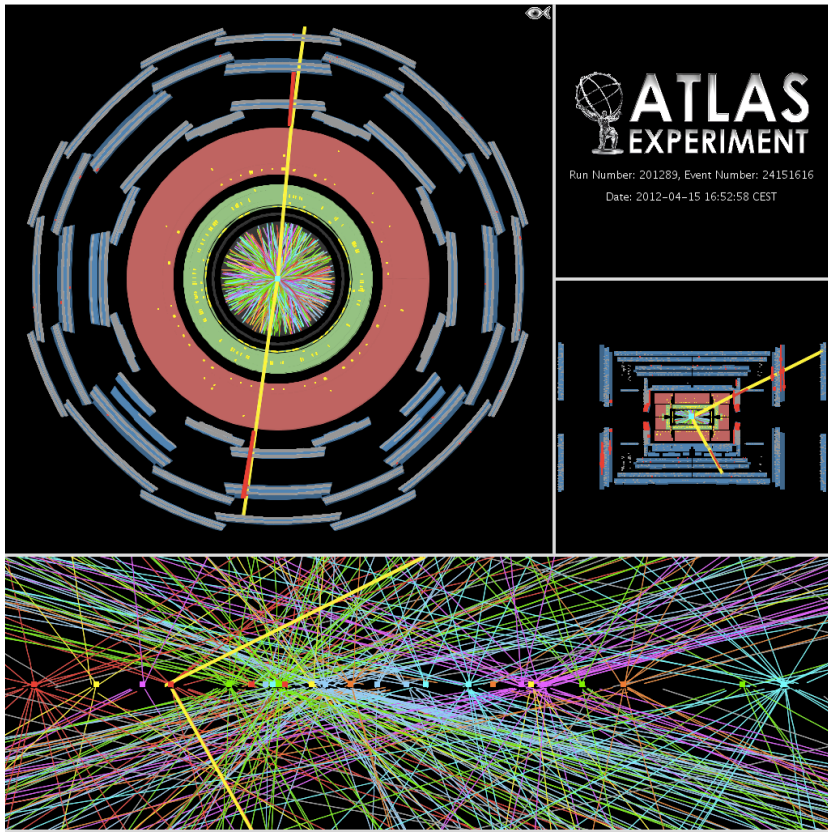
Use average MVA parameter configuration obtained from the *k*-folds.

Particle finding: tracking in Run 4 - case for another ML challenge



Tracking in Run 4

- 20x more tracks than now in Run 2
- x2-5 CPU shortage (offline)
- HEP pattern recognition techniques are more than 25 years old
- Are there better tools out there?



Why is Parallel Tracking so hard?

Algorithms: iterative (propagation, fitting), irregular (combinatorial searches with lots of branch points)

Data: sparse (space-points), non-local (magnetic field integration)

*Can **Machine Learning (ML)** provide a solution that uses regular, simple algorithms, and is naturally data parallel?*
(idea born @ “Connecting the dots”, LBNL, Feb 2015)

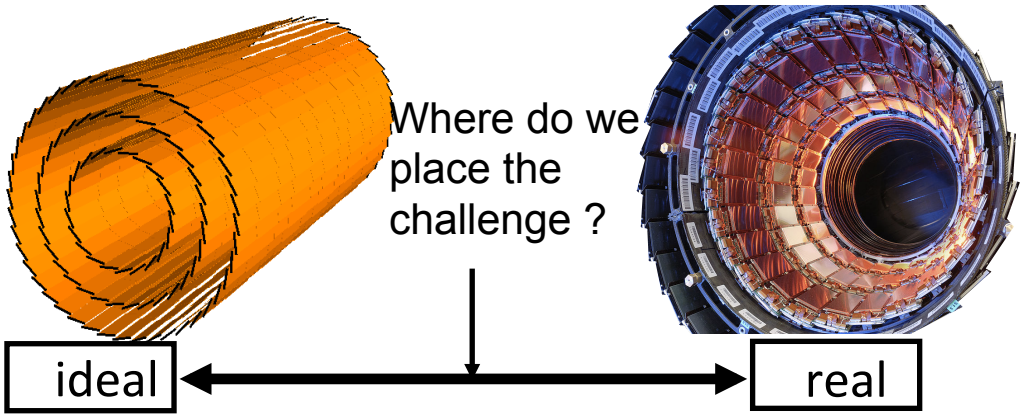
Defining the Tracking ML Challenge

- One question (follow-up questions possible)
- One evaluation metric (training function)
- Two data samples: Training (labelled), Test
- One “starting kit” (reference solution)

- Immediate Goal:
  - Build a fast, **scalable,** pattern recognition engine
- Long-term Goal:
  - Learn if-how-where to apply ML to reconstruction

Formulation of challenge is challenge itself

- What we want: given a list of 3D space-points, group them to form charged particle tracks, minimizing the number of wrong combinations of space-points, and maximizing the number of charged particles found
- Metric:
  - Positive weight for each space-point correctly assigned to track
  - Negative weight for fake space-points
  - Negative weight for algorithm complexity (proxy for execution time)
- How can timing aspects be taken into account?
  - Adding them to the metric would require reference hardware



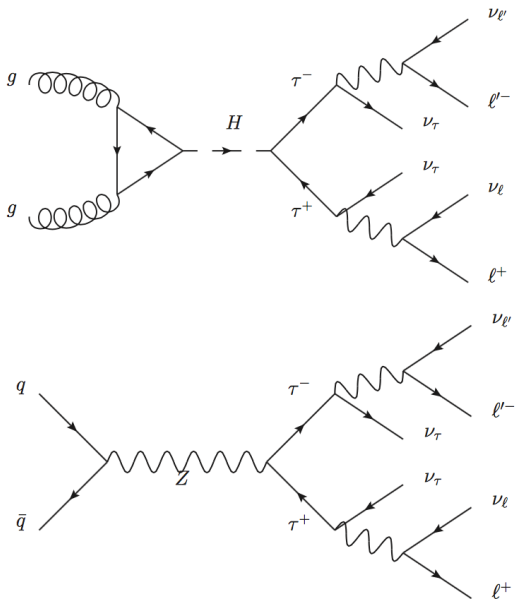
Interesting follow-up questions (Example: tracking in Run 4)

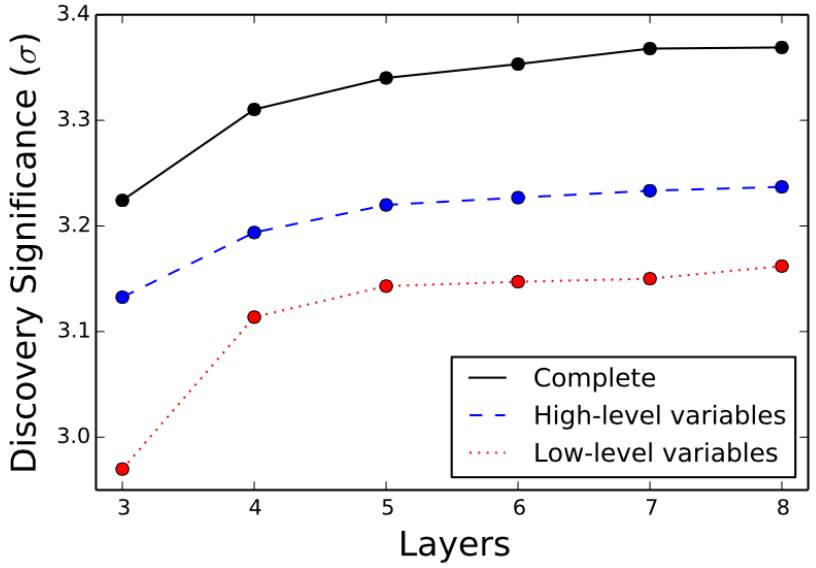
- How to communicate efficiently between HEP and ML communities?
  - Series of dedicated and coordinated LHC ML challenges
    - Building on top of each other
    - Focusing on different aspects of the kind of problem we have in HEP
    - Could become “standard problems” for ML community
  - In addition: individual ML experts work within HEP and focus on dedicated problems with a dedicated toolkit (most of them here in the room – hear their feedback?)
- How to communicate efficiently between HEP experiments?
  - Inter-Experimental LHC Machine Learning Working Group: http://iml.cern.ch
  - Platform to collect, share and develop expertise, ask questions, common software solutions,…

Event classification: Deep Learning

Example: DL & low-level inputs

- Apply Deep Learning to optimize event classification
  - Searches for new physics
  - Search for Higgs$\rightarrow\tau\tau$
  - Low level: momenta of final state objects
  - High-level: nonlinear combinations: invariant mass, opening angle, scalar sum of p_T, etc.
- Optimum: combination of low- and high-level variables





Automatic fault detection / anomalies

Automatic fault detection / anomalies

- Need to flag data with poor quality (sub-detector not operational, corrupt data,…)
  - Compare hundreds of distributions (characterizing the data quality) with references
  - References evolve with time
  - Perfect example for ML

- Dreaming out loud: search for New Physics = search for anomalies (with respect to our Standard Model)
  - ML as a generic search [ATLAS-CONF-2012-107] for anomalies?
  - Huge phase space
  - Correlated with anomaly due to poor data quality

Loose ends: HEP Particularities

- Mismodeling: data vs. simulation
  - Systematic uncertainties based on mismodeling uncertainty
  - The better the classification the larger the deviation (showstopper, e.g. photon ID)
  - (Limited) possibility to validate and calibrate MC to data
- In MC we use data with a large variation in relative weights / neg weights – problems for training
- Variable-length / non-continuous input feature phase space
- We usually have a model based on our physics knowledge – this leads to two extreme approaches:
  - Matrix Element Method (MEM): rely on “calculable” part of model
  - ML: let machine learn (still model dependence)
  - MEM pros & cons:
    - Pros: no need to train, no need for large statistics, make us of maximum available information
    - Cons: slow for complex final states, many approximations/simplifications of the model needed
- **Can we combine ML and physics input in a smart way?**
- Features may vary significantly e.g. with p_T or eta (analogy: facial expressions in face recognition)

Conclusion / Final Thoughts

- Promising opportunities for ML application in HEP, and vice versa (cross-fertilization)

- Reality check:
  - Are we using state-of-the-art tools in HEP
  - How close are we to the “best solution”

- HEP computing needs will become more pronounced in future: strong link with ML to prepare for it
  - Systematic approach & know-how: how to apply toolkit, tune algorithms, troubleshoot,…

Thanks for input to

Adrian Bevan

Paolo Calafiura

Andrea Coccaro

Mauro Donega

Markus Elsing

Michael Kagan

Marie Lanfermann

Ben Nachman

Olaf Nackenhorst

Luke de Oliveira

David Rousseau

Andreas Salzburger

Steven Schramm

Thomas Stevenson

Jean-Roch Vlimant

Daniel Whiteson

And many others…