Event reconstruction and Particle identification in the ALICE experiment at the LHC

Iouri Belikov for the ALICE Collaboration

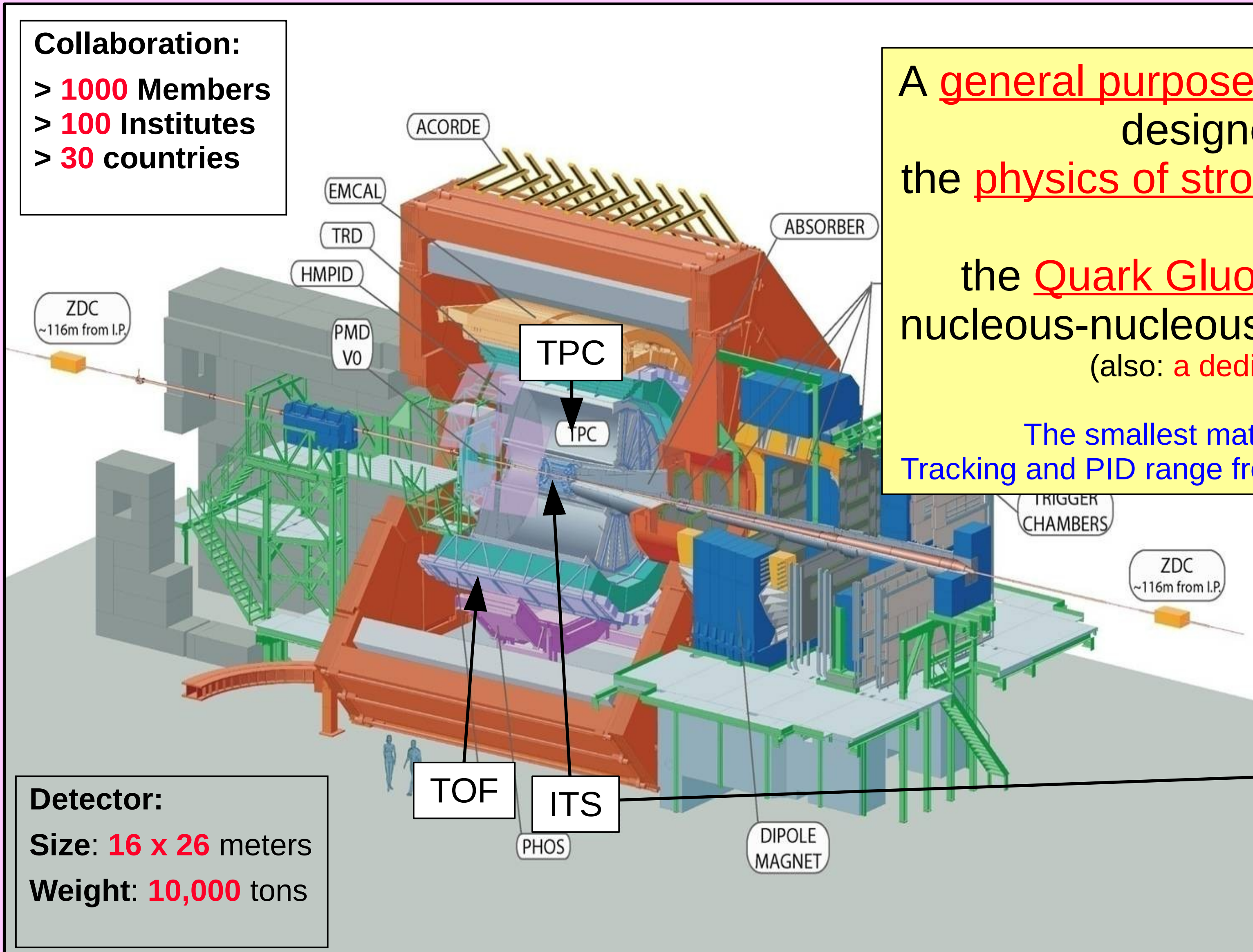
- ALICE experiment at the LHC
- Track and vertex reconstruction
  - What it is, strategy and methods, issues, performance
- Particle identification
  - What it is, strategy and methods, issues, performance
- Outlook and conclusions

The ALICE experiment

(see the talk of P. Christakoglou, plenary on Tuesday)

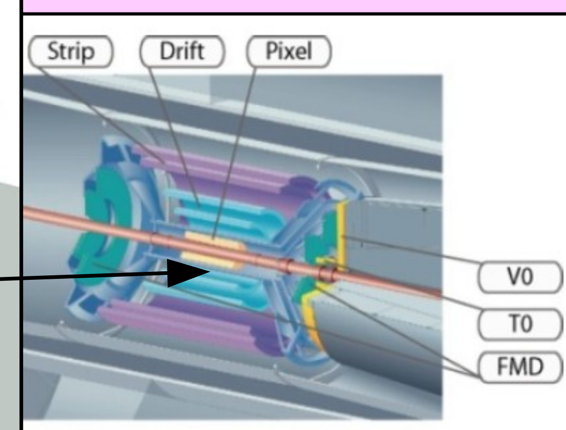
Collaboration:

> 1000 Members
> 100 Institutes
> 30 countries



A general purpose heavy-ion experiment
designed to study
the physics of strongly interacting matter
and
the Quark Gluon Plasma (QGP) in
nucleous-nucleous Collisions at the LHC.
(also: a dedicated pp program)

The smallest material budget at the LHC
Tracking and PID range from pt<0.1 and up to ~100 GeV/c



Detector:

Size: 16 x 26 meters

Weight: 10,000 tons

Event reconstruction (the challenge of)

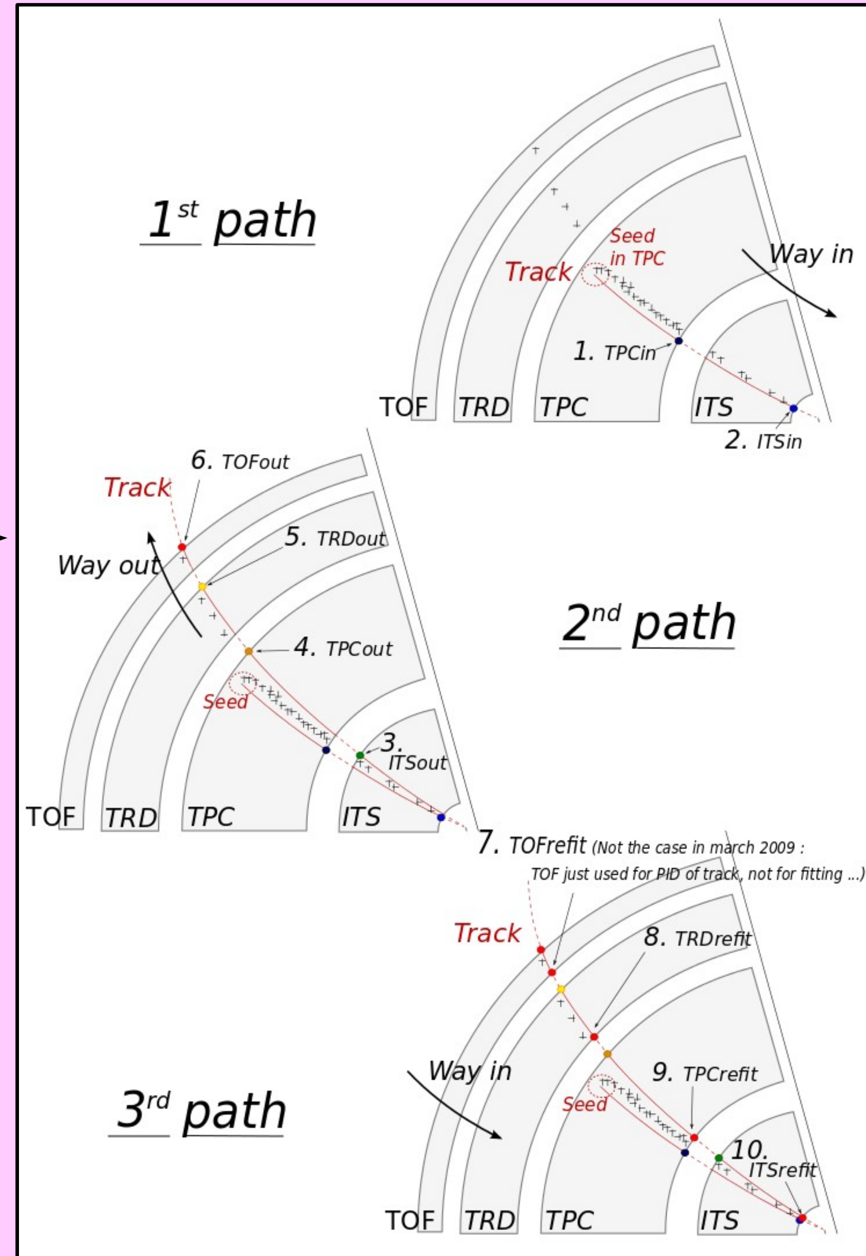
- The offline event reconstruction is software. (There is also quasi-online, and HLT)
- Reconstruction "converts” the Raw Data into the Event Summary Data.
  - The Raw Data are files containing encoded “positions & ionization” (in units of pad/wire number, ADC/TDC counts etc).
  - The Event Summary Data (ESD) are files containing fitted particle momenta, vertex positions, probabilities related to PID, etc (in units of GeV/c, cm, ...).

- The technical challenge of the reconstruction (PbPb)
  - A typical data taking rate: (at least) a few hundred events per second
  - A typical reconstruction rate: one event per (a few) minutes

There is a factor ~10^4 difference between the data taking and reco rates!
(of course, this is compensated by the number of CPUs. But... Anyway...)

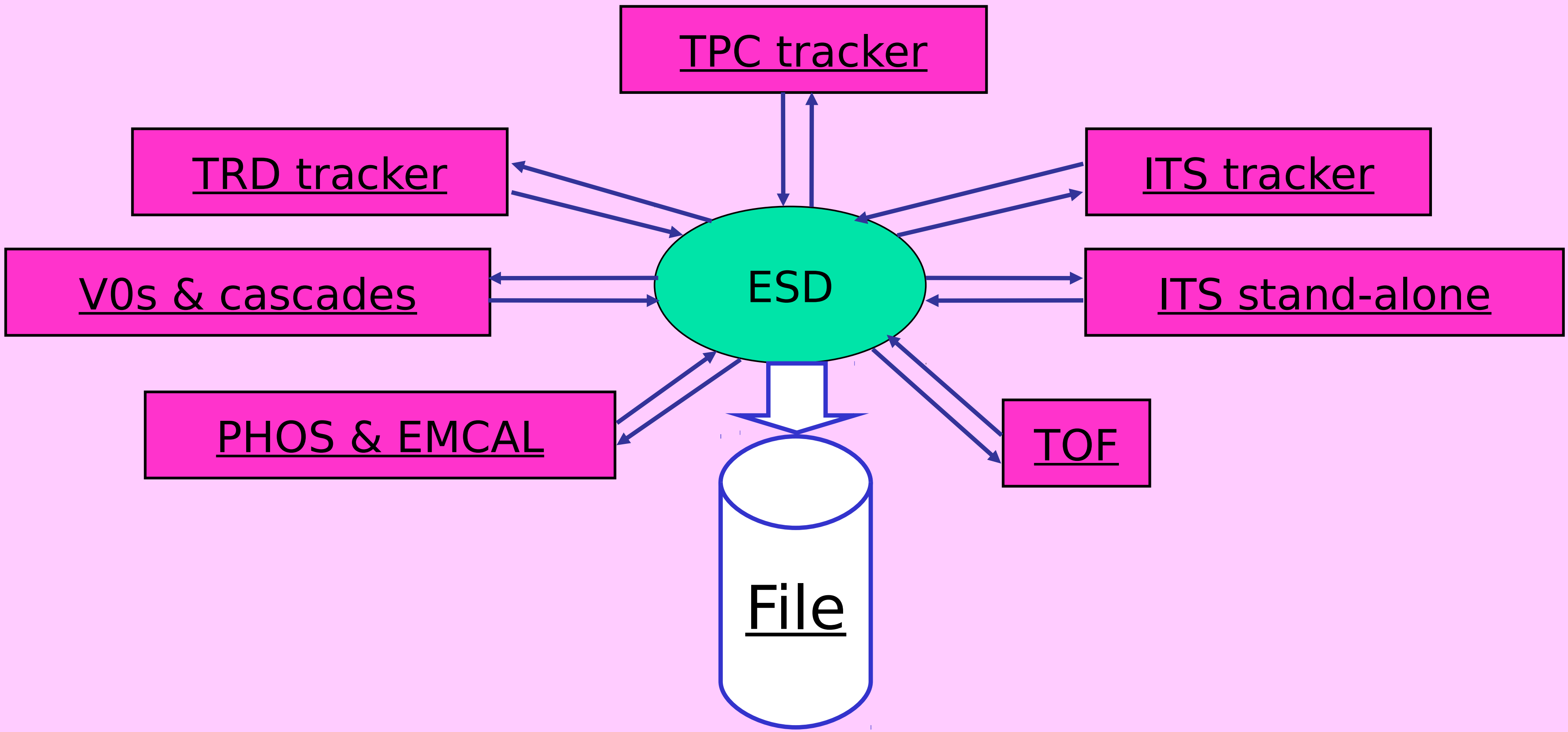
General reconstruction strategy

- Cluster finding in the detectors (centre of gravity)
  - Unfolding of overlapped clusters (optional)
- Primary vertex reconstruction using the ITS (SPD)
  - Pileup detection (optional)
- "Seeding" in TPC (with/out the vertex constraint)
  - Later, also the "seeding" in ITS and TRD (optional)
- Combined tracking (three passes) ⟶
  - On-the-fly kink and V0 reconstruction (optional)
- Primary vertex using the tracks
- Secondary vertices using the tracks (V0s,cascades)
  - Groups of tracks satisfying some cuts



The main tracking algorithm is Kalman Filter
(With some ad-hoc additions in the ITS)
also for vertex fitting...

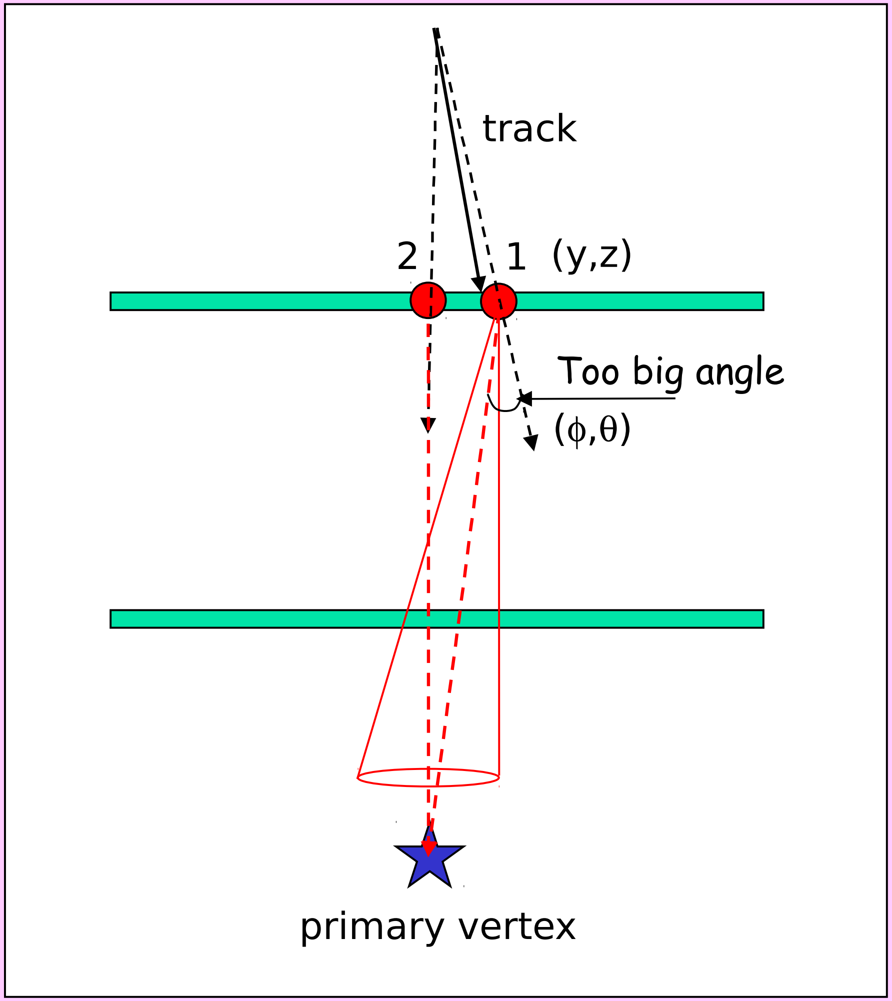
A bit on the software design...



- The process is configured/triggered by a special macro (rec.C).
- The general initialization and the processing sequence is defined centrally in by a dedicated class (AliReconstruction).

An issue: ad-hoc "vertex constraint" in the ITS

The ITS tracking "investigates" a full tree of possible track prolongations from the TPC



Looking at the cluster position only, the cluster #1 is better than the cluster #2.

But, taking into account the direction towards the primary vertex, the cluster #2 becomes preferable…

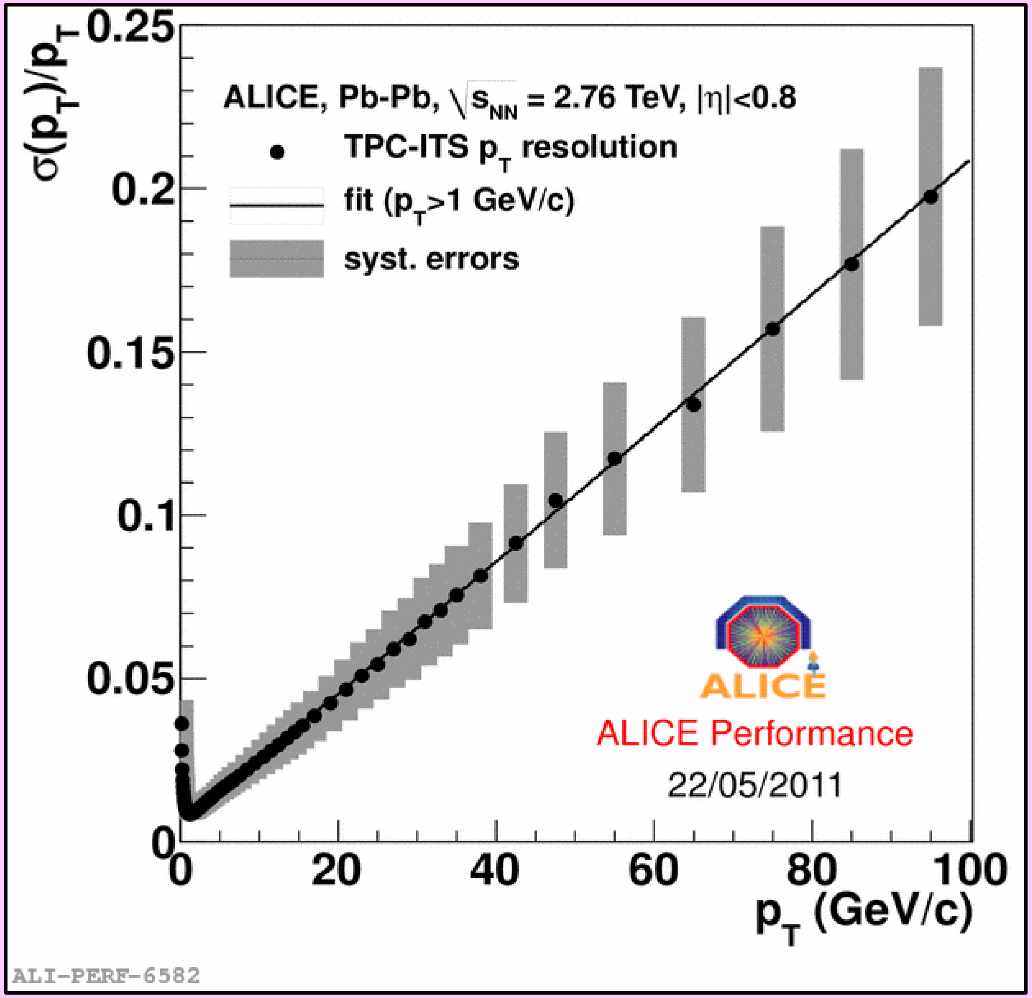
Technically, this is done by extending the "measurement" *(y,z) -> (y,z,ϕ,θ)*

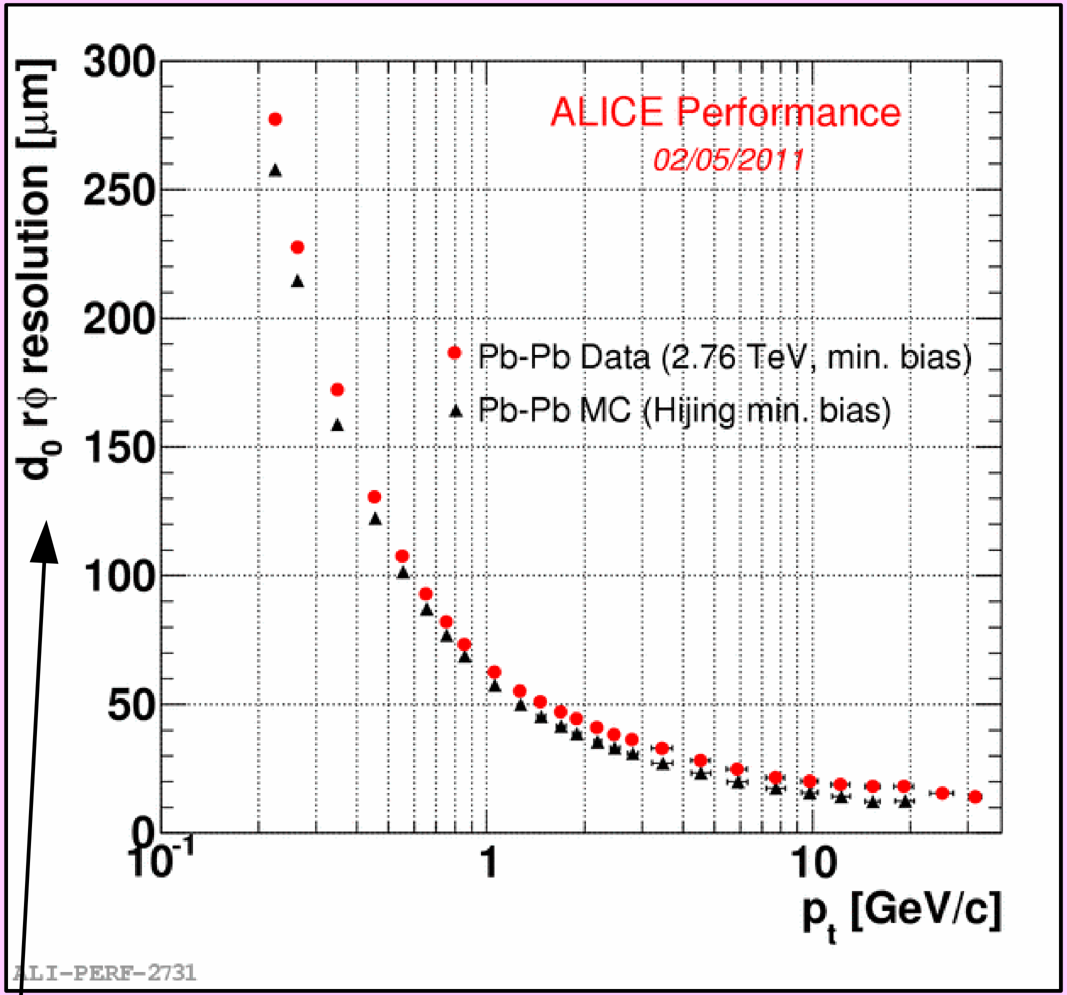
A question:

Can all this be justified ? Improved ?

Problem: it is applied to the same track repeatedly... (a re-fit is needed)

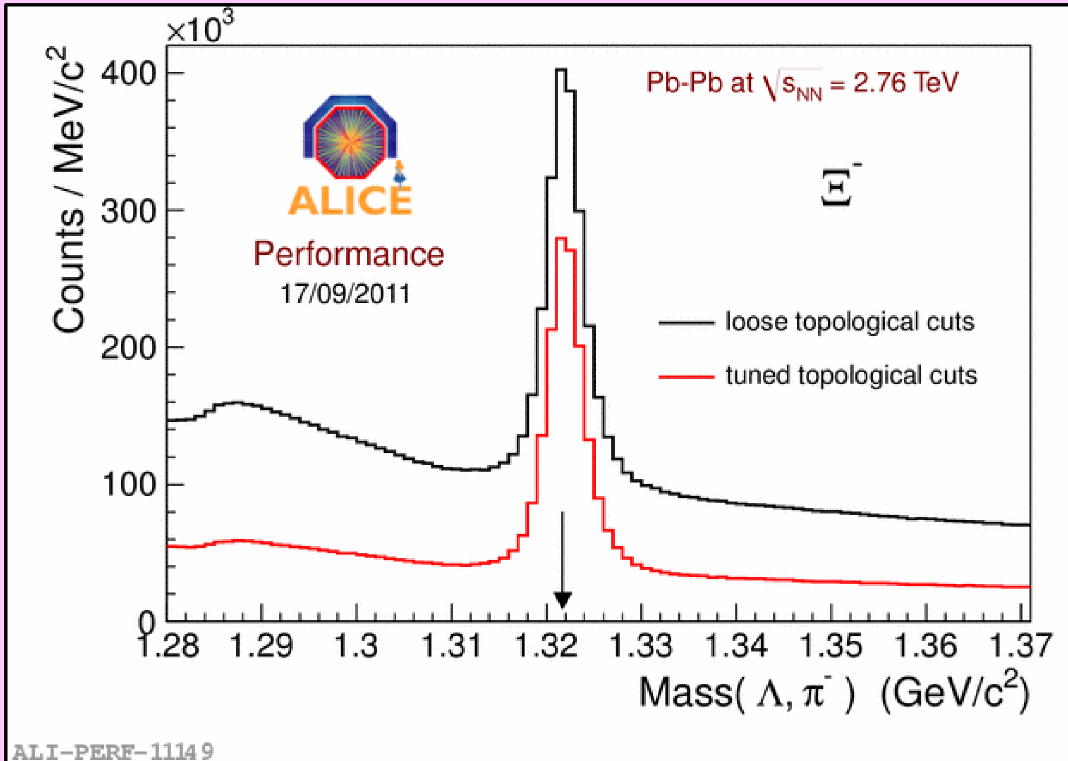
Tracking performance

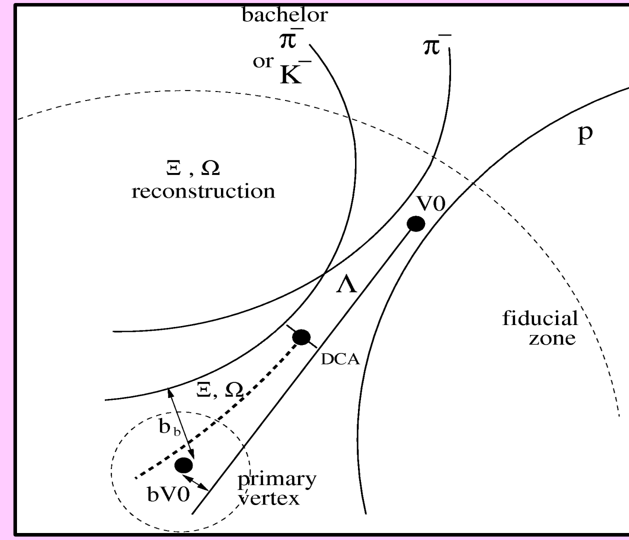


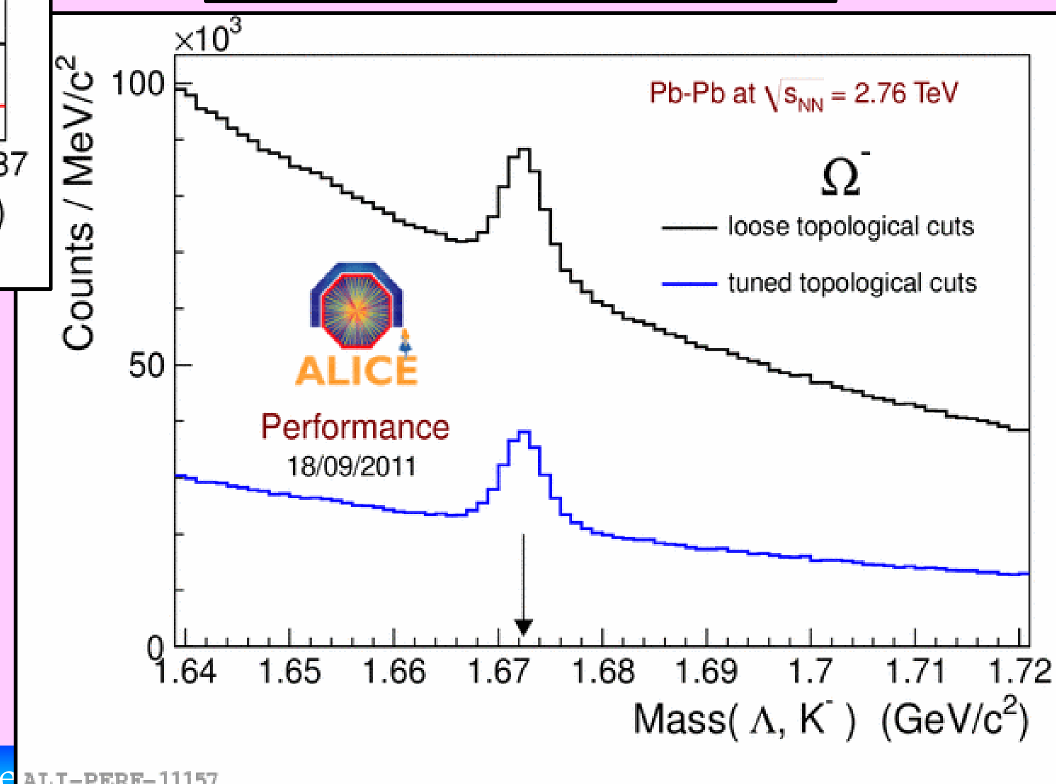


d0 is the minimal distance between a track and the primary vertex in the transverse plane.
Crucial for charm reconstruction !

Secondary vertex performance







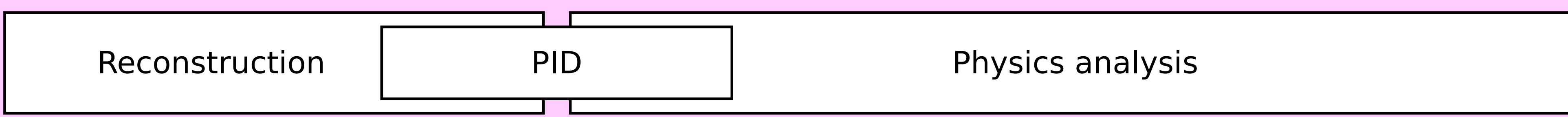
Optionally:

The vertices can be fitted using KF and applying (primary) vertex constraints

What the PID is

- Particle Identification (PID) is everything that provides some information about the masses of detected particles.
  - PID is tightly connected with the reconstruction (tracking needs masses, fitted momenta are needed for PID). PID is based on quite similar algorithms as many parts of reconstruction. PID contributes to ESD. <u>This is a part of reconstruction.</u>
  - However, PID extends beyond the reconstruction towards the physics analysis. <u>It is also a part of physics analysis !</u>

This is quite fundamental…



Bayesian approach

Example: a supervisor choosing a summer student

- What is the probability of choosing a girl or a boy ?
- The answer depends on:
  - Probability with which this supervisor chooses a girl *p(g)*, or a boy *p(b)*
  - The number of application submitted by girls Ng, and by boys Nb (the priors)
- The final probability is given by Bayes' formula:

$$P_g = \frac{p(g)Ng}{p(g)Ng + p(b)Nb}$$

$$P_b = \frac{p(b)Nb}{p(g)Ng + p(b)Nb}$$

Bayesian approach in PID

Probability to be a particle of ***i***-type (***i*** = *e*, *μ*, *π*, *K*, *p*, …),
if the PID signal in the detector is ***s***:

$$w(i \mid s) = \frac{C_i r(s|i)}{\sum_{k=e,\mu,\pi,\dots} C_k r(s \mid k)}$$

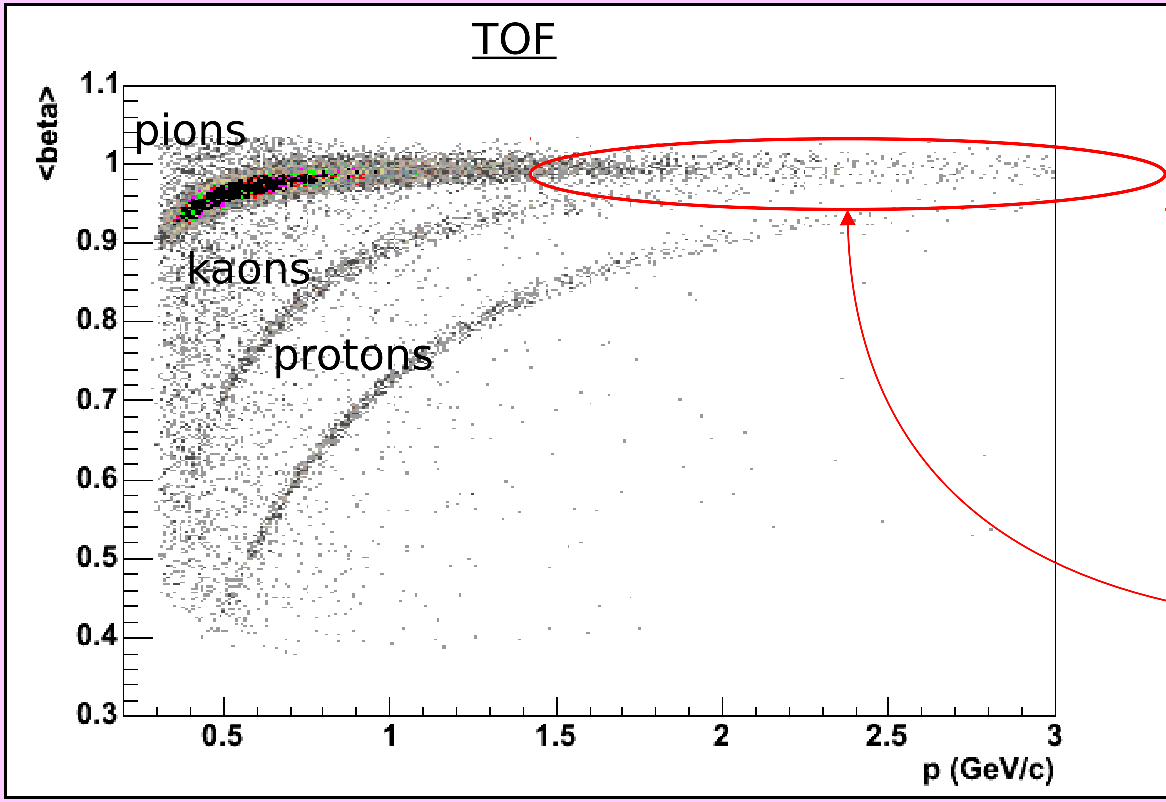
- ***C**_i* - *a priori* probabilities to be a particle of the ***i***-type.
  "Particle concentrations", that depend on the event and track selection.
- ***r(s|i)*** – conditional probability density functions to get the signal ***s***, if a particle of ***i***-type hits the detector.
  "Detector response functions", depend on properties of the detector.

In the case of *N* contributing detectors: $r(s_1,\dots,s_N \mid i) \sim \prod_{d=1}^{N} r(s_d \mid i)$

A complementary approach: n-sigma cuts

- Pros:
  - No need to deal with the priors
  - Guarantees a definite momentum independent efficiency.
- Cons:
  - Does not tell anything about the contamination
  - Does not maximize the signal/background ratio
  - Needs the “raw PID signals” (additional disk space)
- Everything that concerns the response functions is the same as for the Bayesian
  - A big piece of software can be shared by the two approaches

Issue 1: the high p_T limit



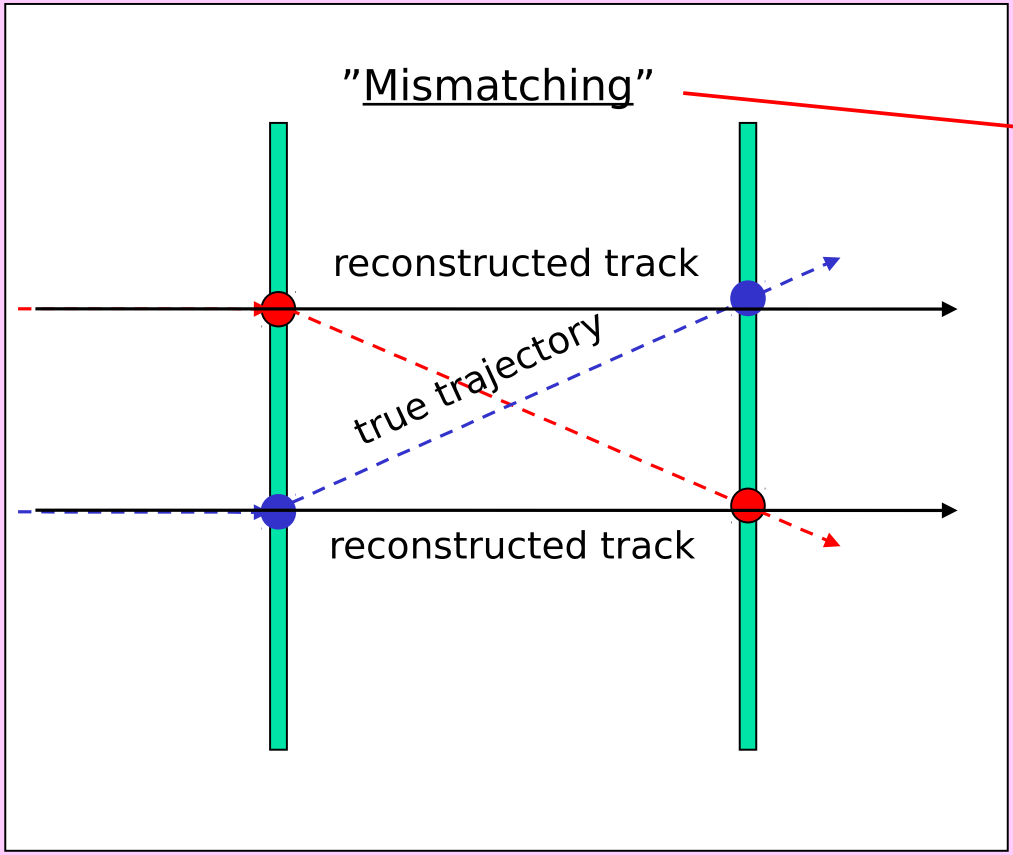
The Bayesian calculations nicely glue together the momentum sub-ranges, but, as the momentum goes up, the “separation power” vanishes, and…

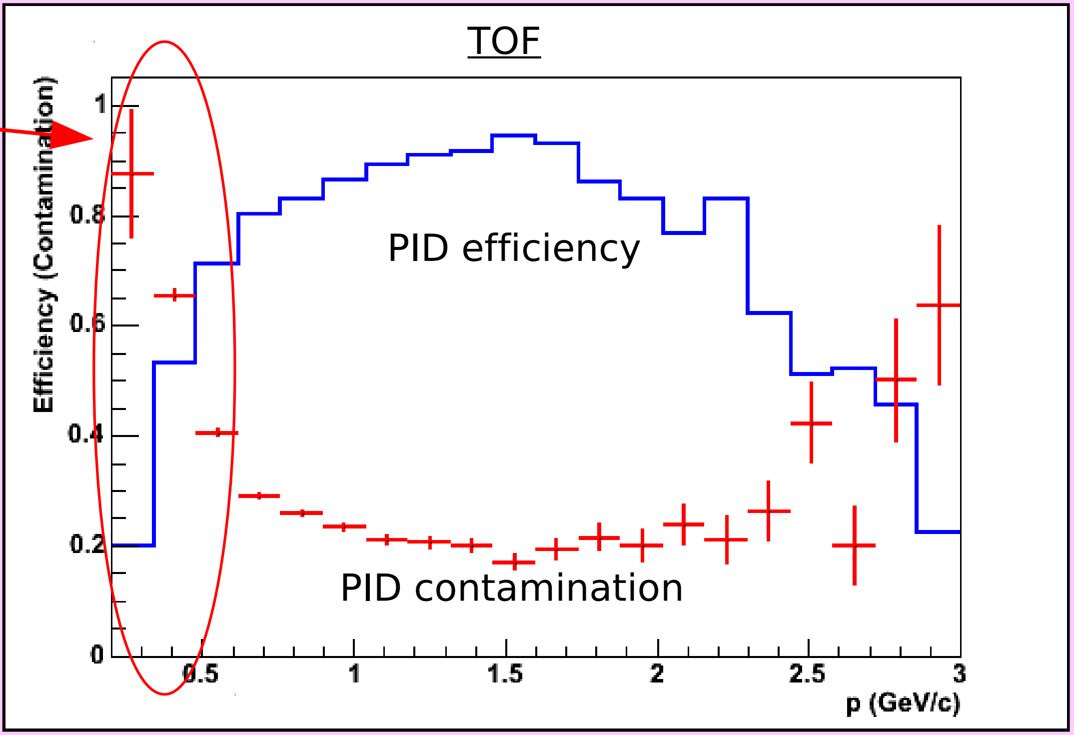
We are left with the bare priors ☹

Questions:

- The influence of the priors on the final result: Can it be somehow quantified ?
- For any PID approach: at what momentum should we stop doing the PID ?

Issue 2: track mismatching





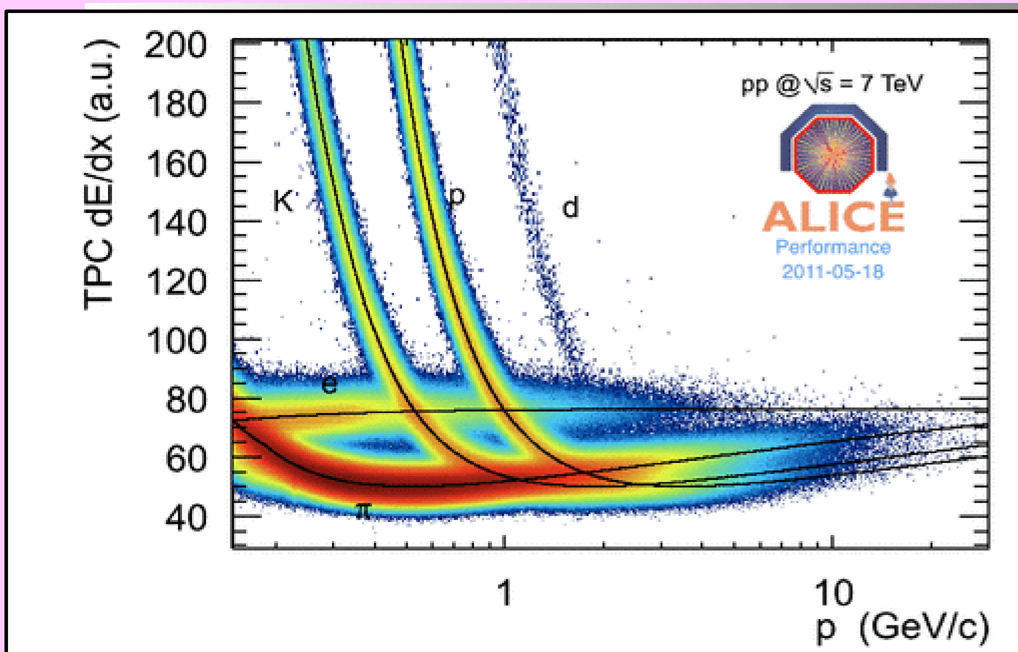
The mismatching biases the combining the PID information (any PID approach !), because the detectors do not register the same particle.

Question:

- How can we take the mismatching into account ?
- (The solution for the moment: different kinds of "vetoing")

PID performance

(see the talk of M. Spyropoulou-Stassinaki, HI on Monday)

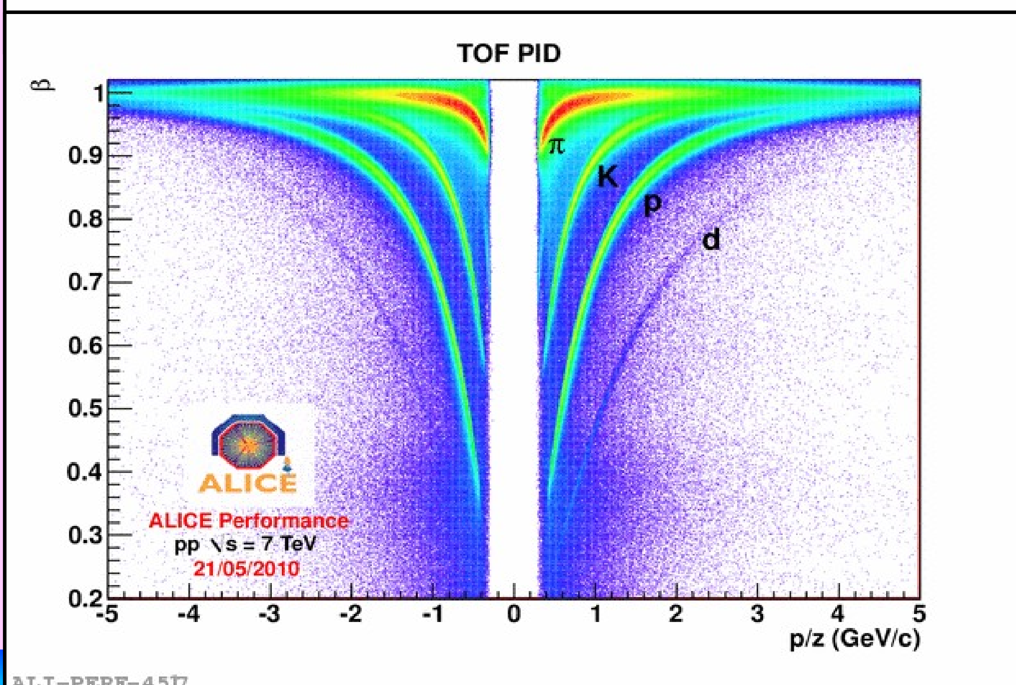


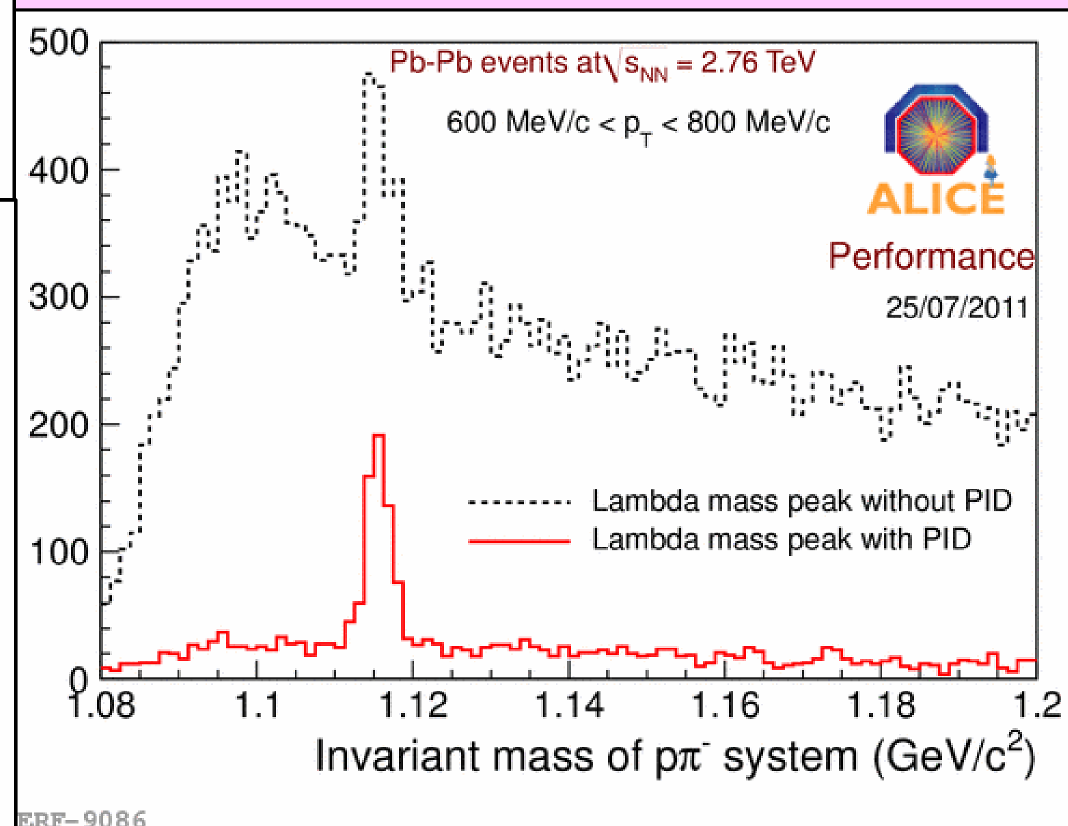
TPC: σ/<dE/dx> ~ 5.5 % (~150 clusters)

TOF:

σ ~ 90 ps (intrinsic)

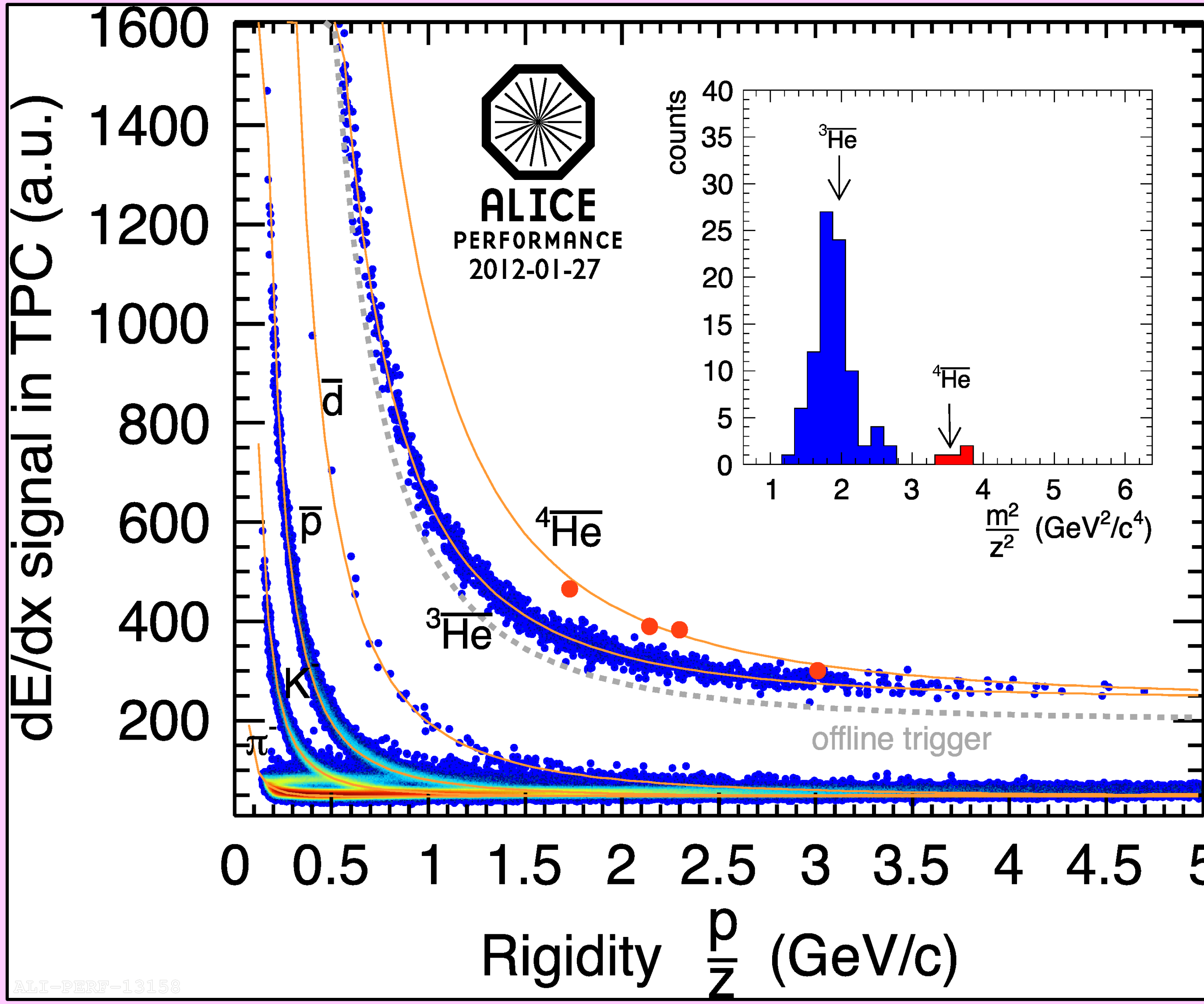
σ < 150 ps (depending on the 'start')





PID performance

- 16 M PbPb events
- 4 anti-alpha candidates, ID-ed combining with the TOF information.



Outlook and conclusions

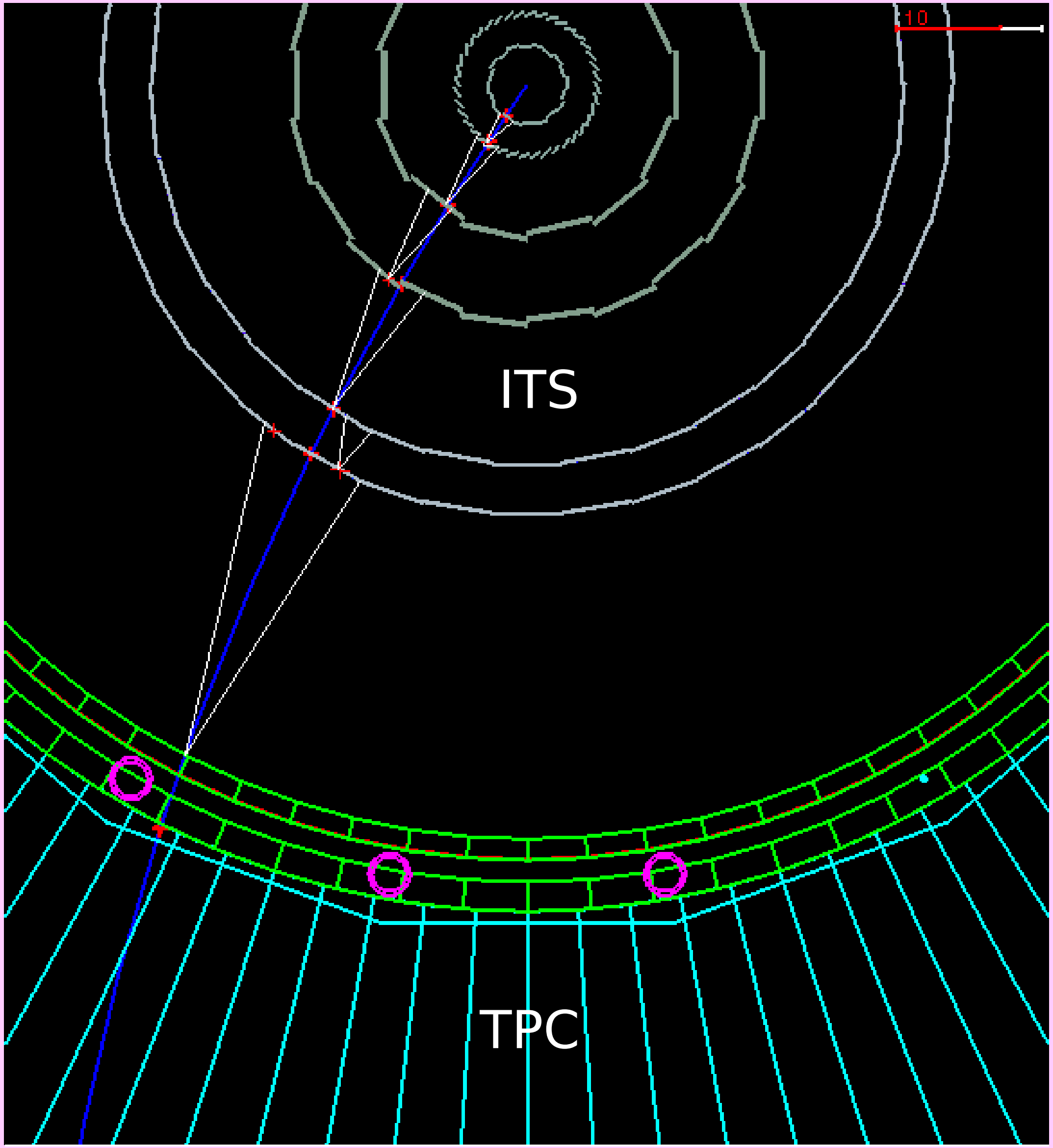
- The main track reconstruction algorithm in ALICE is Kalman Filter (optionally used also for vertex fitting). In addition, in ALICE, we do:
  - Full “tree” of possible track prolongations in the ITS
  - Ad-hoc “vertex constriant” in the ITS + subsequent re-fit w/o the contraint
- Efficiency is limited only by decays (at low p_t) or acceptance (at high p_t)
- Typical momentum resolution: ~1% at p_t~1 GeV/c, ~20% at ~100 GeV/c.
- Typical impact parameter resolution: ~50 μm at p_t~1 GeV/c, ~10 μm asymptotically

- Two complementary PID approaches: Bayesian (preciser), n-sigma (simpler)
  - The high-p_t limits need to be consolidated
  - The treatment of track mis-matching can be improved
- Typical (mentioned here) resolutions: 5.5% dE/dx in TPC, 90 ps in TOF (intrinsic)

> - Reconstruction and PID in ALICE are challenging and quite interesting.
> - However, ALICE physics is even more interesting.
>
> Let’s be doing physics ! ☺

Backup slides

Statistical problems with track finding in ITS



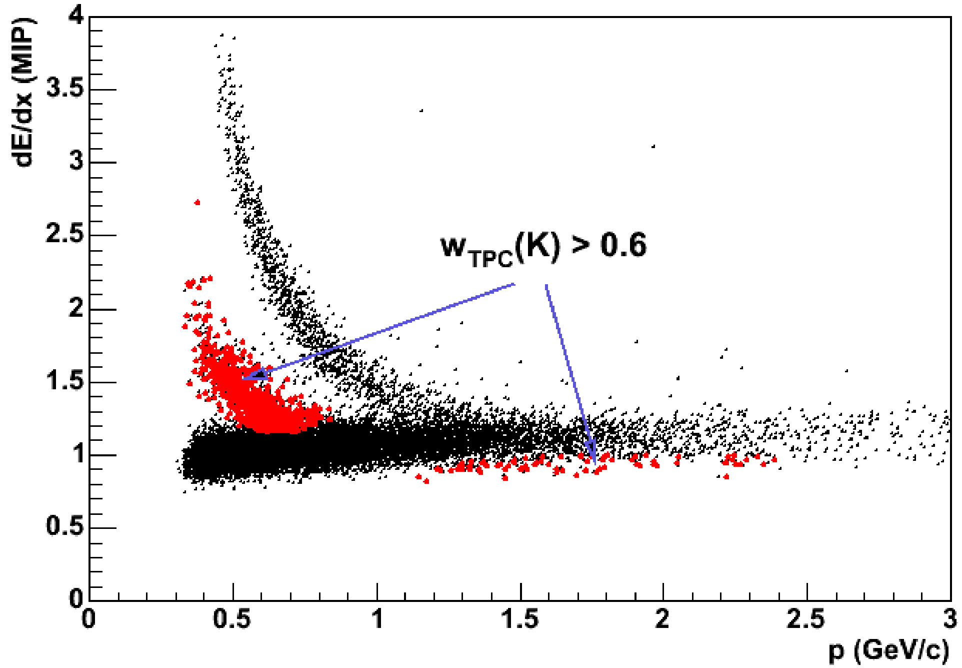
Several clusters within the “road” defined by the multiple scattering...

Suggested solution:

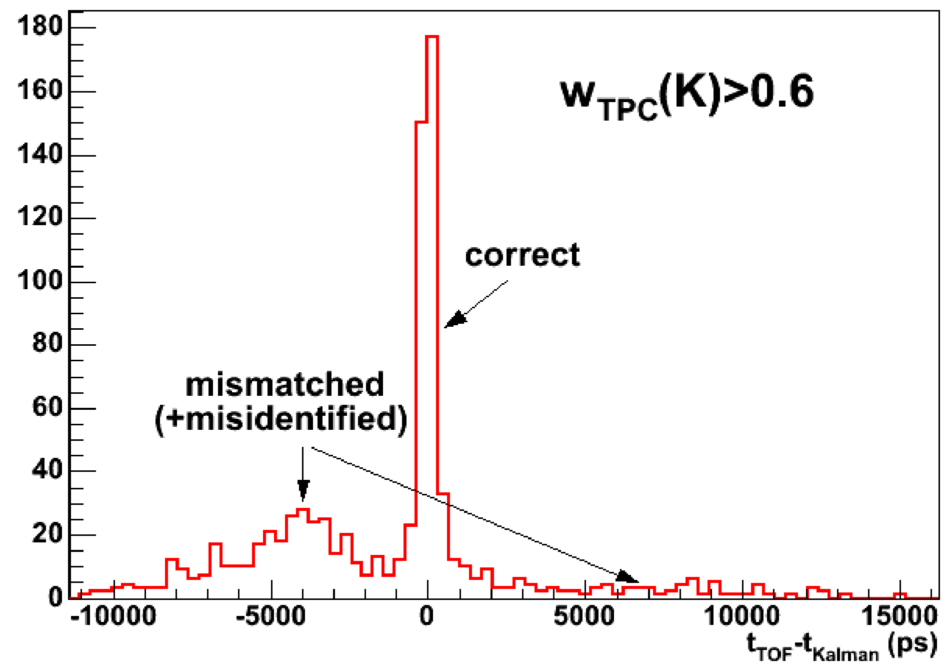
- Investigation of a whole tree of possible track prolongations.
- Applying an ad-hoc “vertex constraint” (1st pass).

Ad-hoc treatment of the track mismatching

TPC



TOF



Observing in one of the detectors the distribution of signals for a clean sample of particles pre-selected in other detectors, we can get the range of signals, where the probability of mismatching is "high" → Veto in the combining…

A question: Can it be somehow generalized ? Made "smooth" ? Optimized ?

Something like $w = (1-p_{12})w_1 + p_{12}w_{12}$ (p_{12} - prob. of a correct matching) ?