ALICE: Tracking and computing resources

David Rohr

drohr@cern.ch, CERN

SPHENIX Computing Workfest

Brookhaven National Lab, NY

13.1.2020

ALICE in Run 3: 50 kHz Pb-Pb

Record large minimum bias sample.

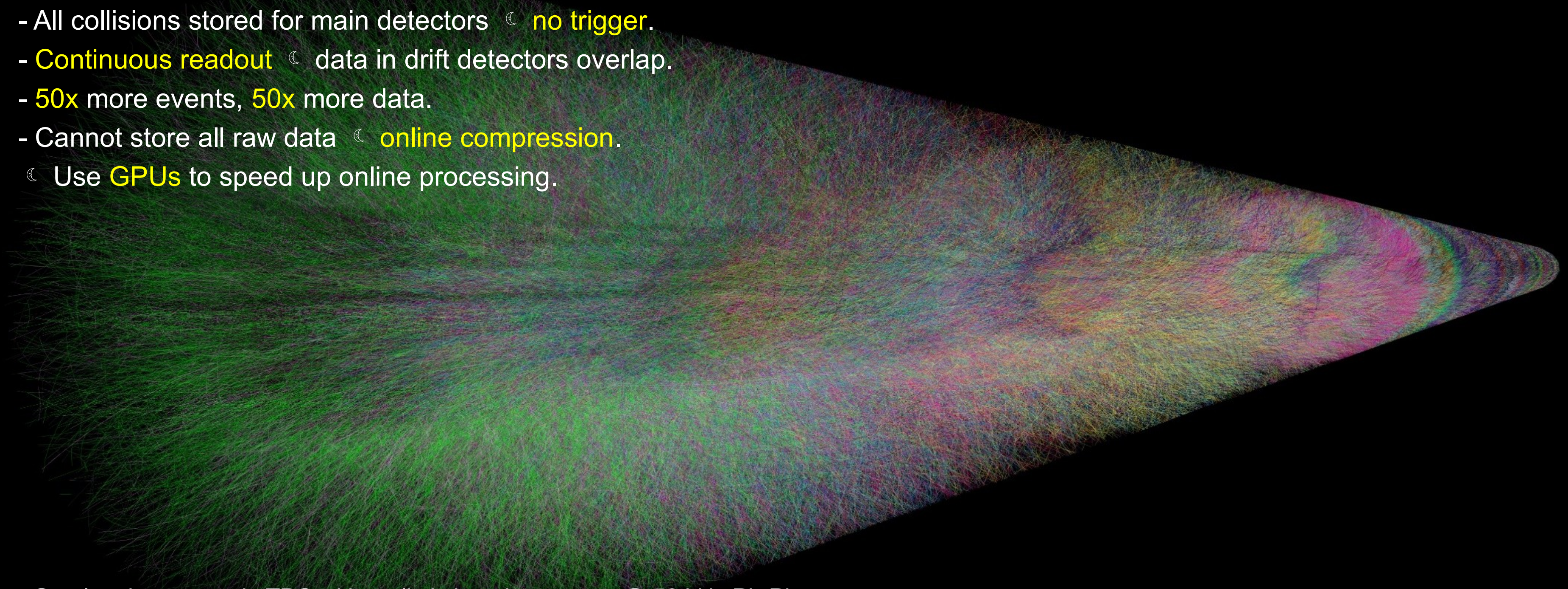
- All collisions stored for main detectors ☾ no trigger.

- Continuous readout ☾ data in drift detectors overlap.

- 50x more events, 50x more data.

- Cannot store all raw data ☾ online compression.

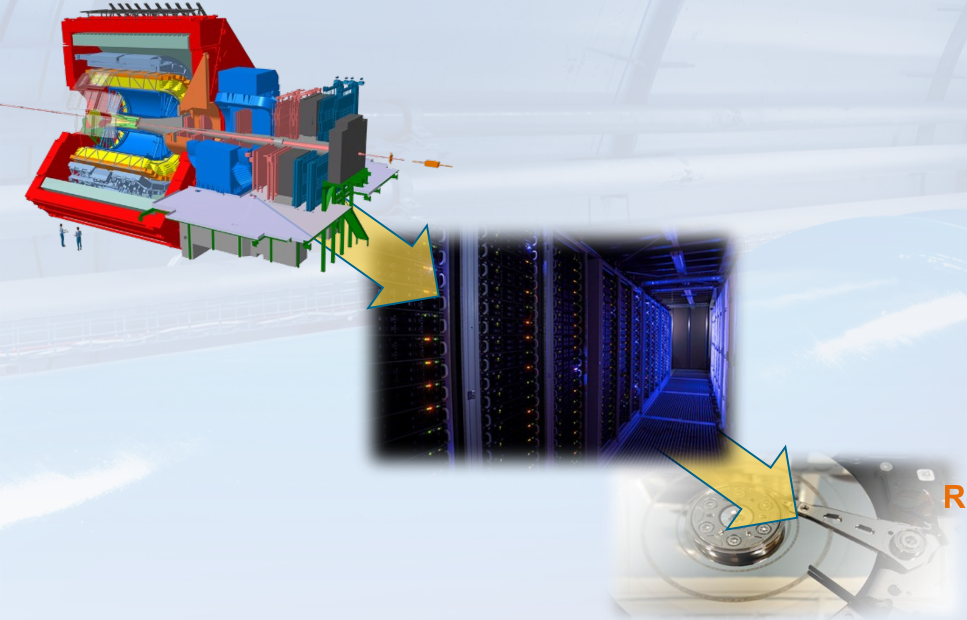
☾ Use GPUs to speed up online processing.

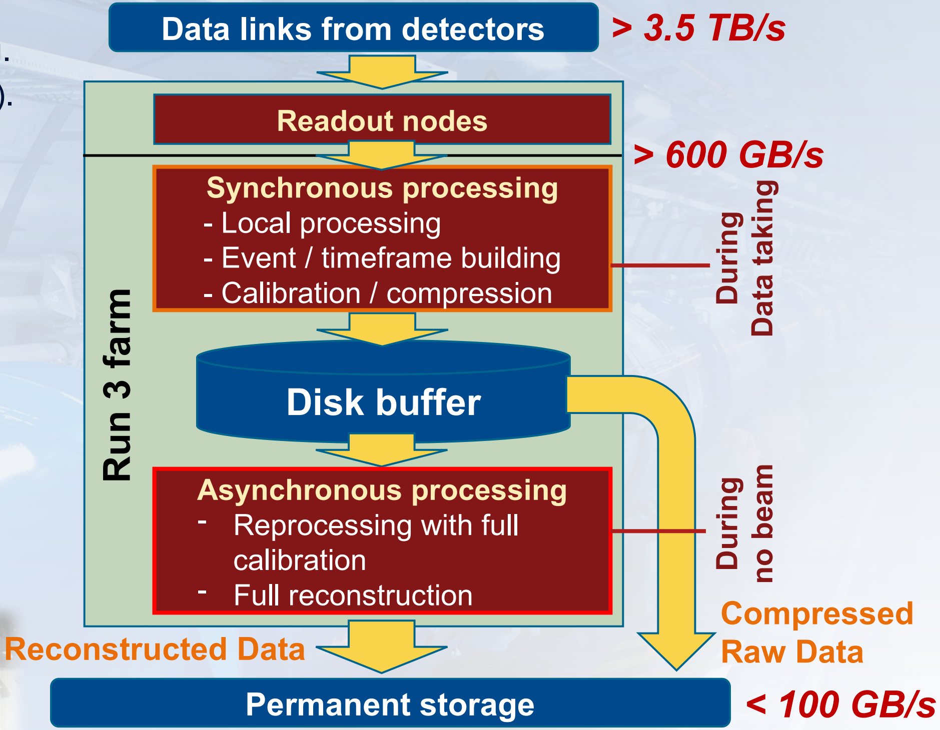


- Overlapping events in TPC with realistic bunch structure @ 50 kHz Pb-Pb.
- Timeframe of 2 ms shown (will be 10 – 20 ms in production).
- Tracks of different collisions shown in different colors.

Online / Offline Computing in ALICE in Run 3

- **ALICE computing strategy for Run 3**
  - On-site server farm for synchronous (online) processing.
  - When not taking data ☾ used for asynchronous (offline).





Data links from detectors

> 3.5 TB/s

Run 3 farm

Readout nodes

> 600 GB/s

Synchronous processing
- Local processing
- Event / timeframe building
- Calibration / compression

During Data taking

Disk buffer

Asynchronous processing
- Reprocessing with full calibration
- Full reconstruction

During no beam

Reconstructed Data

Compressed Raw Data

Permanent storage

< 100 GB/s

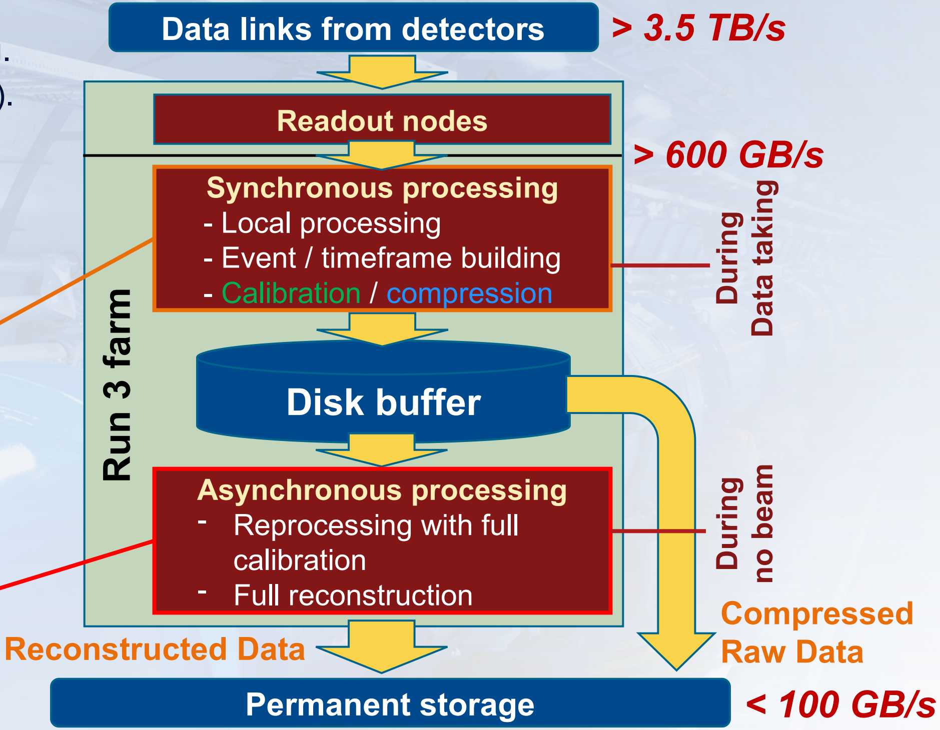
Online / Offline Computing in ALICE in Run 3

- **ALICE computing strategy for Run 3**
  - On-site server farm for **synchronous** (online) processing.
  - When not taking data ☾ used for **asynchronous** (offline).

- **Partial ITS + TPC + TRD tracking for TPC calibration**
  - reduced statistics sufficient
  (TPC calibration based on matching of TPC / ITS / TRD tracks)
- *Other detectors without significant CPU load*

- **Full TPC tracking for TPC compression**
  - cluster to track residuals ☾ better entropy coding
  - removal of tracks not used for physics
- *Entropy coding for other detectors*

Final reconstruction pass with final calibration



Data links from detectors

> 3.5 TB/s

Run 3 farm

Readout nodes

> 600 GB/s

Synchronous processing
- Local processing
- Event / timeframe building
- Calibration / compression

During Data taking

Disk buffer

Asynchronous processing
- Reprocessing with full calibration
- Full reconstruction

During no beam

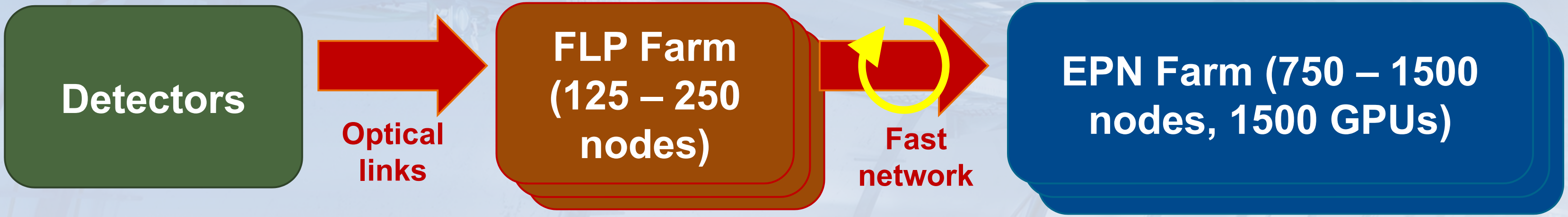
Compressed Raw Data

Reconstructed Data

Permanent storage

< 100 GB/s

ALICE Run 3 Data Taking & Online Computing Scheme



Detectors

Optical links

FLP Farm (125 – 250 nodes)

Fast network

EPN Farm (750 – 1500 nodes, 1500 GPUs)

- Sub-event building
- Processing steps that need access to all data from a link (e.g. integrated digital current)

Round-robin distribution:
An EPN receives a full timeframe, and has 30 seconds for processing it

- **Synchronous:**
  - Full TPC tracking for compression (on GPUs).
  - ITS / TRD tracking of subset of events for calibration.
  - Integrated digital currents produced for calibration.
- **Postprocessing:**
  - Create space-charge and calibration map.
- **Asynchronous:**
  - Full reconstruction with final calibration for all detectors. (TPC not so dominant, split between Run 3 farm and GRID).

- **Synchronous TPC tracking @50kHz Pb-Pb defines peak-load for Run 3 farm.**
  - → GPU usage for TPC tracking mandatory!
  - Asynchronous reconstruction should leverage available GPU resources as good as possible.
  - Must run efficiently on CPUs on the GRID.
  - GPU reconstruction code written in a general way that runs on CPU & GPU with identical result.

Tracking in ALICE in Run 3

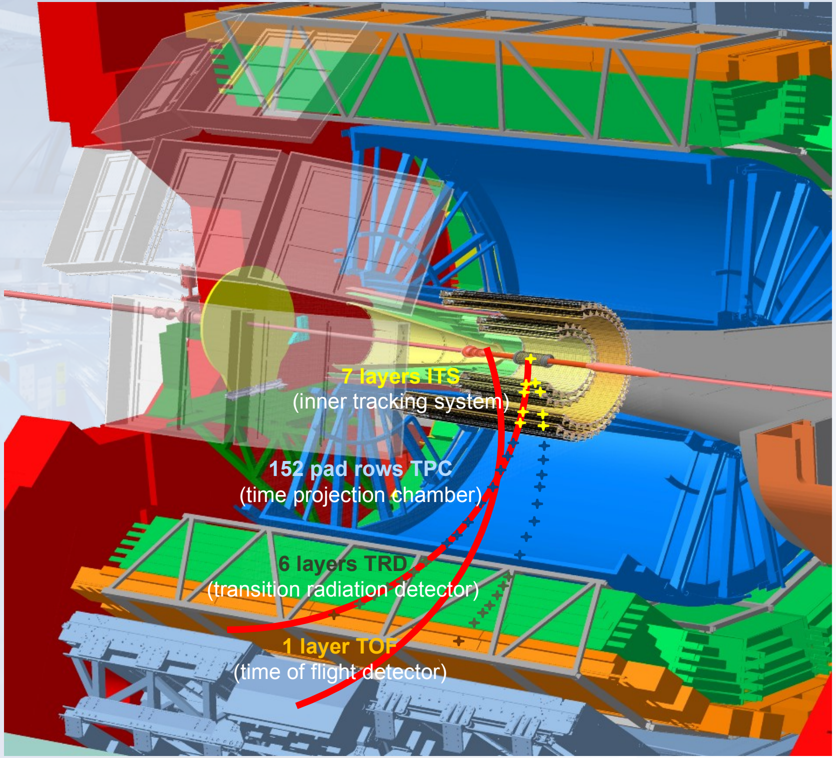
- Bulk of computing workload:

Synchronous

- >90% TPC tracking / compression
- Low load for other detectors

Asynchronous

- TPC among largest contributors
- Other detectors also significant


7 layers ITS
(inner tracking system)
152 pad rows TPC
(time projection chamber)
6 layers TRD
(transition radiation detector)
1 layer TOF
(time of flight detector)

Tracking in ALICE in Run 3

- **Bulk of computing workload:**

Synchronous

- >90% TPC tracking / compression
- Low load for other detectors

Asynchronous

- TPC among largest contributors
- Other detectors also significant

- **ALICE GPU processing strategy**

Baseline solution
(almost available today):
TPC + part of ITS tracking on GPU

- **Mandatory** solution to keep up with the data rate online.
- **Defines** number of servers / **GPUs**.

Optimistic solution
(what could we do in the ideal case):
Run most of tracking + X on GPU.

- Extension of baseline solution to make best use of GPUs.
  - Ideally, **full barrel tracking** without ever leaving the GPU.
  - In the end, we will probably be somewhere in between.

Asynchronous phase should make use of the available GPUs.

- Available in the O² farm anyway.
- Future HPC / grid sites may have GPUs.

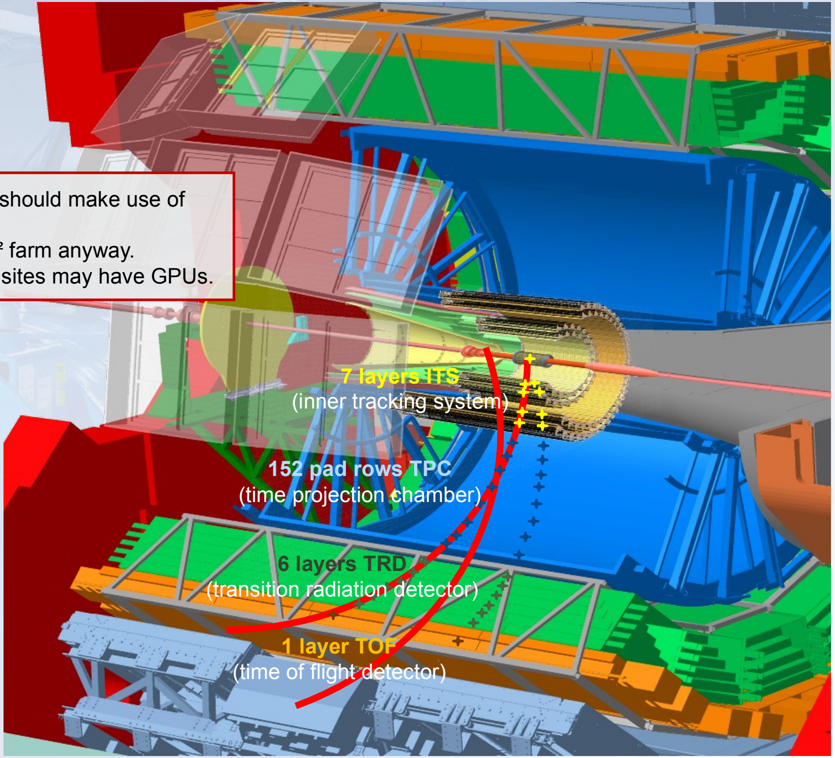
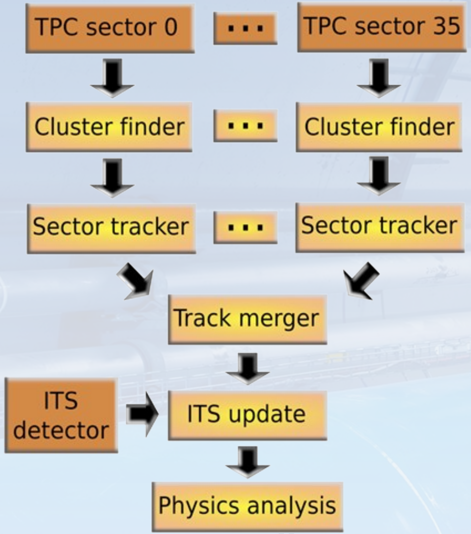

7 layers ITS
(inner tracking system)
152 pad rows TPC
(time projection chamber)
6 layers TRD
(transition radiation detector)
1 layer TOF
(time of flight detector)

Illustration of splitting in sectors

- **Sectors of ALICE TPC:**



TPC sector 0 … TPC sector 35

Cluster finder … Cluster finder

Sector tracker … Sector tracker

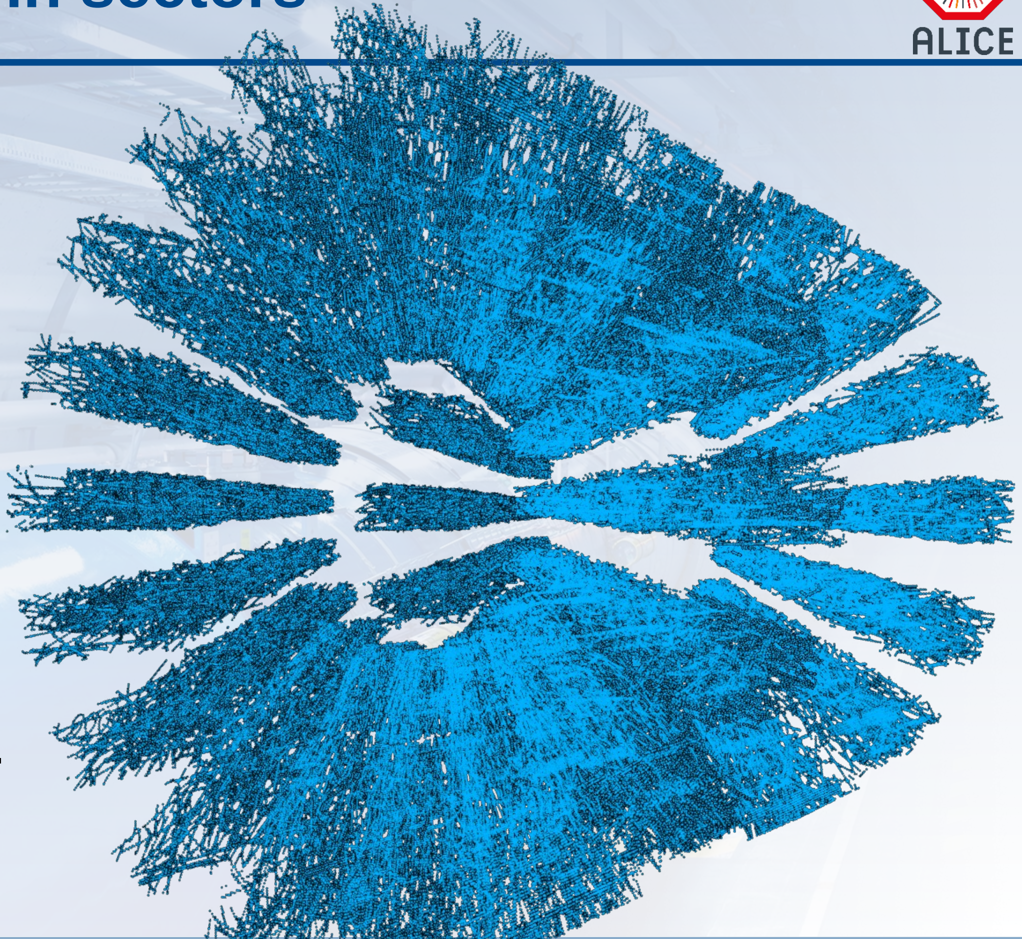
Track merger

ITS detector

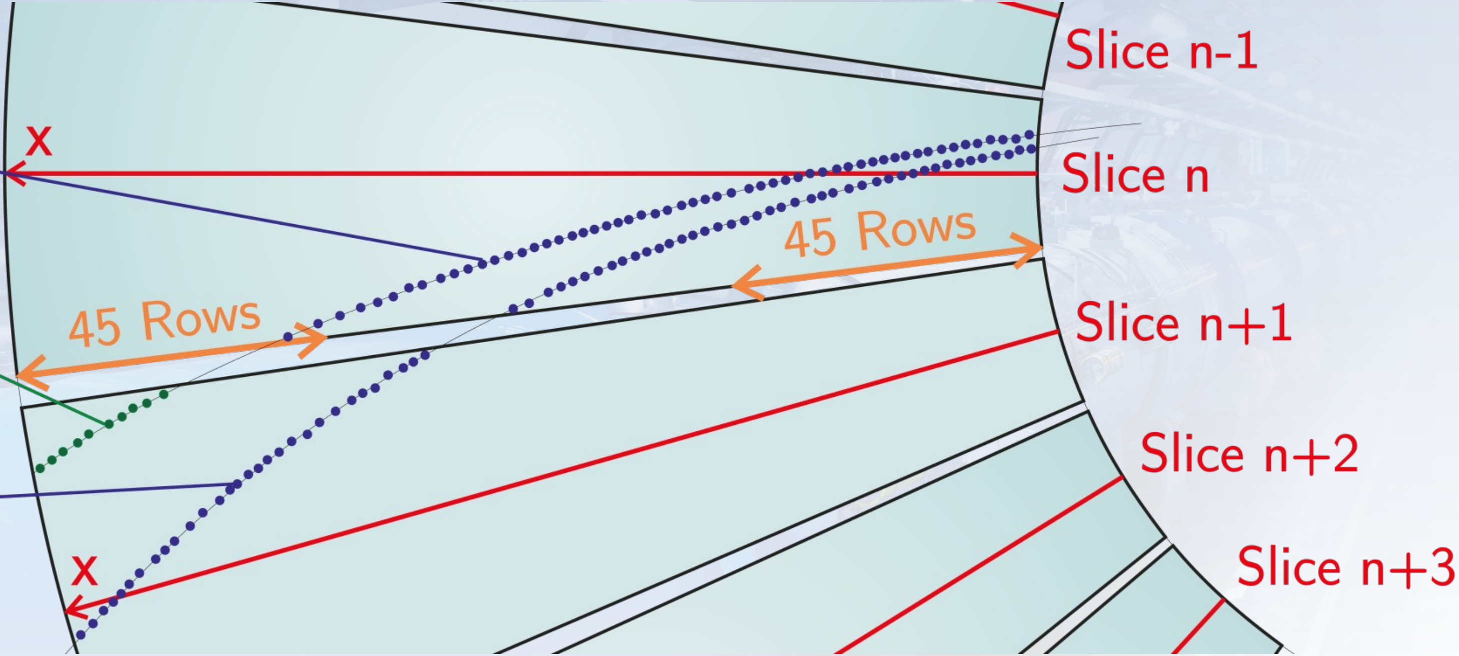
ITS update

Physics analysis

- **Sector-local seeding can lead to some inefficiencies at sector borders.**



Global Tracking



Segment used as seed for Global Tracking

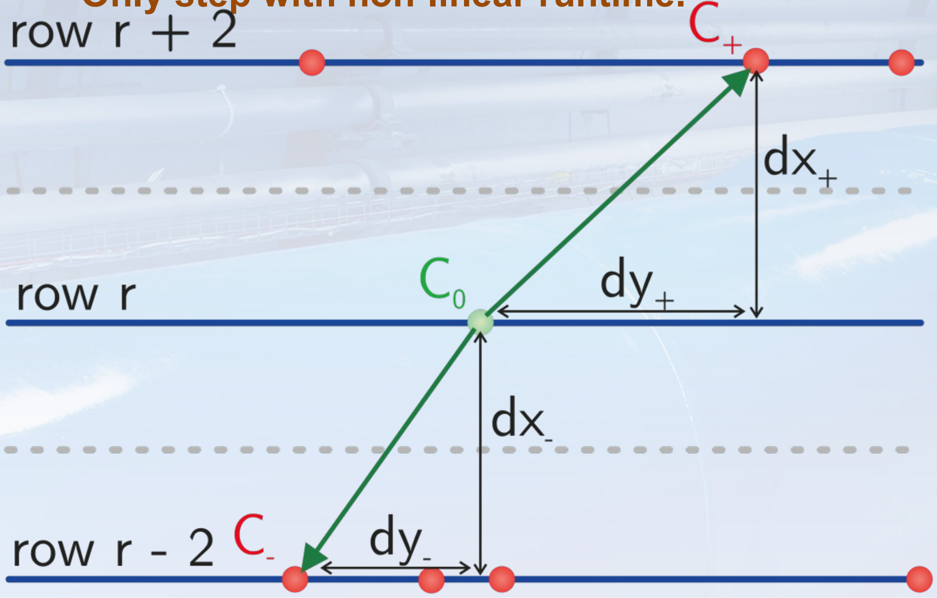
Segment with less than 30 clusters, found only with Global Tracking

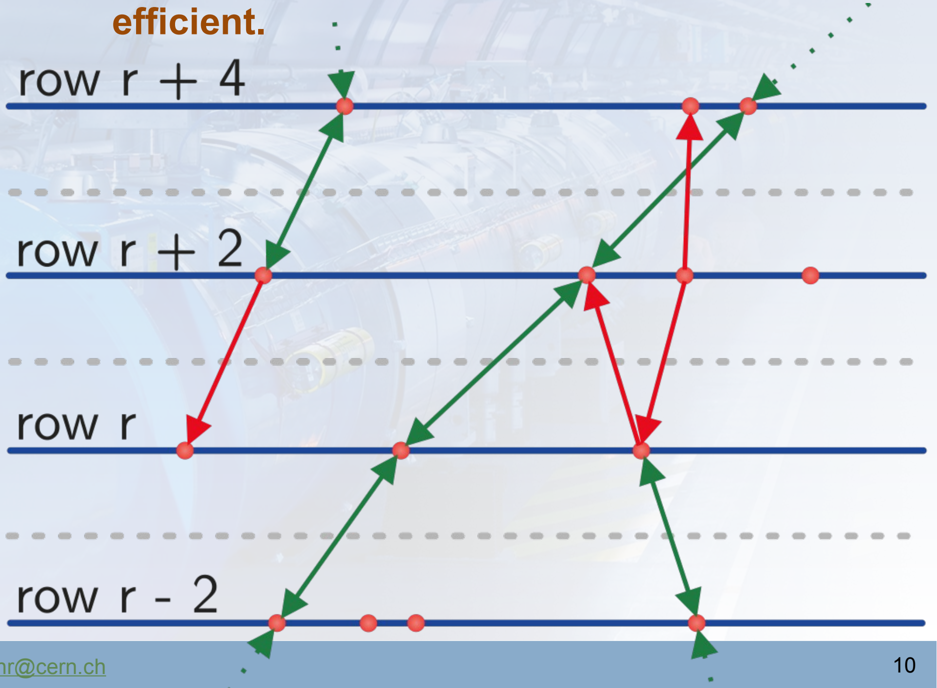
Segment with more than 30 clusters; Global Tracking not necessary

Step 1 (Seeding)

- Step 1: Combinatorial seeding
- Searches for three clusters composing straight line
- Concatenates straight lines
- Only step with non-linear runtime.

- Strategy: deal with the combinatorics as early as possible.
- Seed everywhere, each track has at least some seedable part, no need to be 100% efficient.





Step 2 (Track Following)

- Step 2 (Simplified Kalman Fit):
- Track parameters are fit to the seed.
- Trajectory is extrapolated to adjacent TPC row.
- Cluster closest to extrapolated position is found.
- Fit is improved with new cluster.

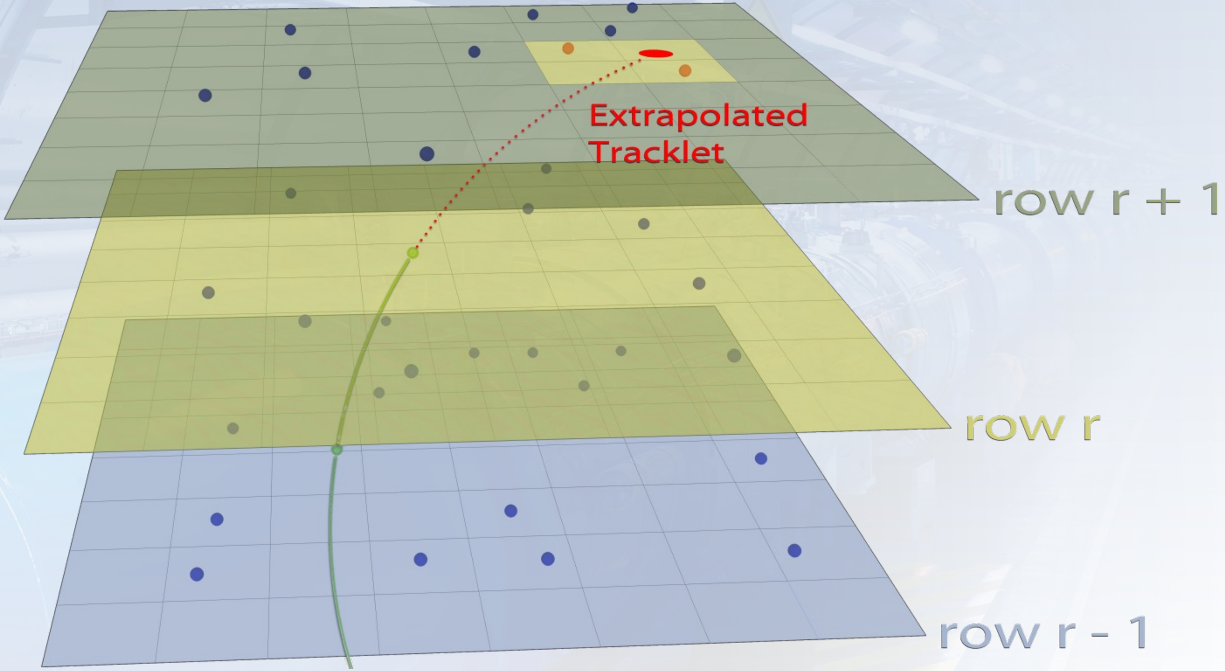


Illustration of Tracking

- **Illustration of step 1:**
- **(Seeding)**

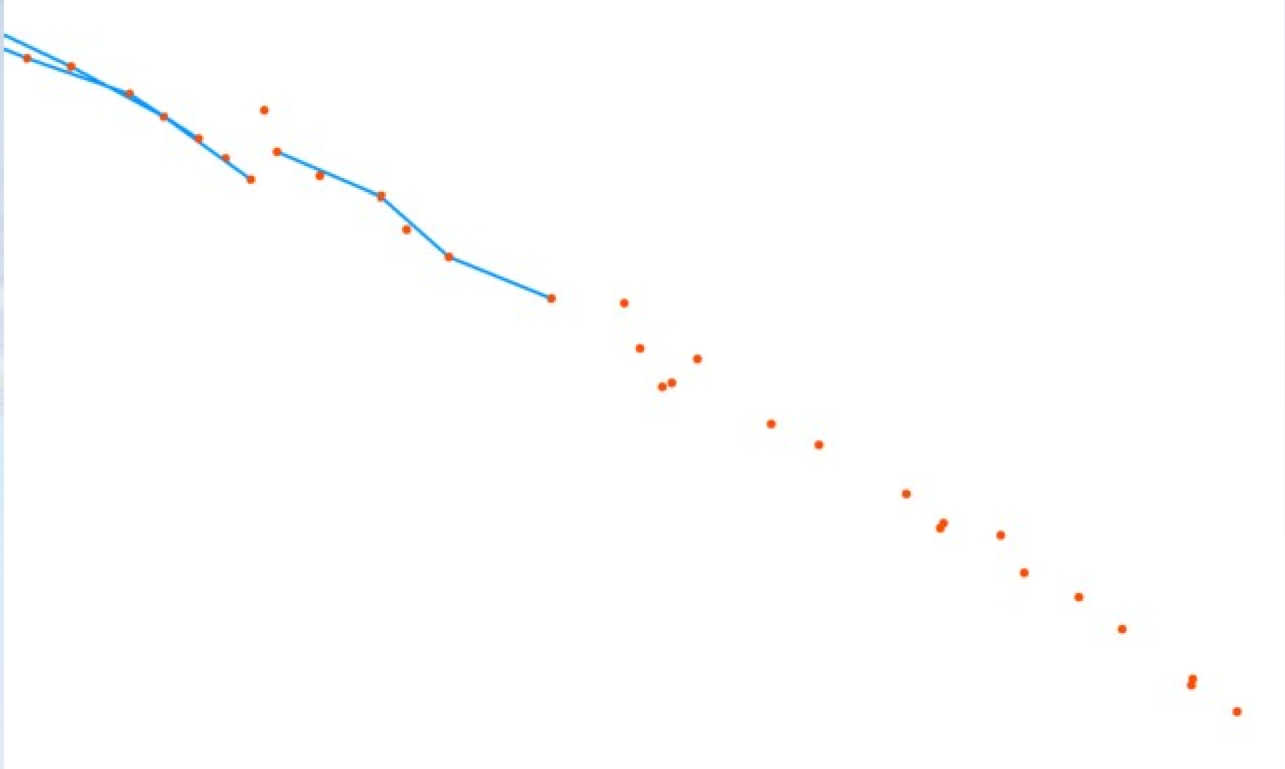


Illustration of Tracking

- Illustration of step 2:
- (Track following)

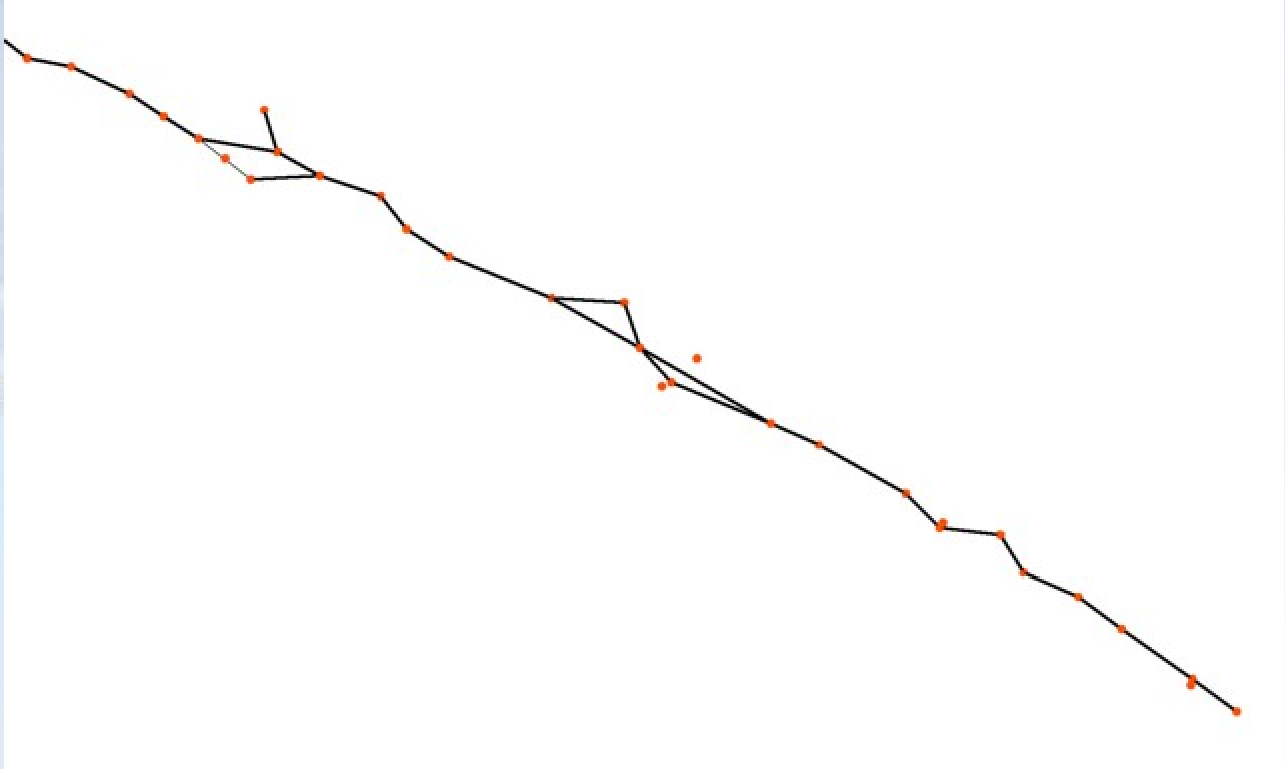
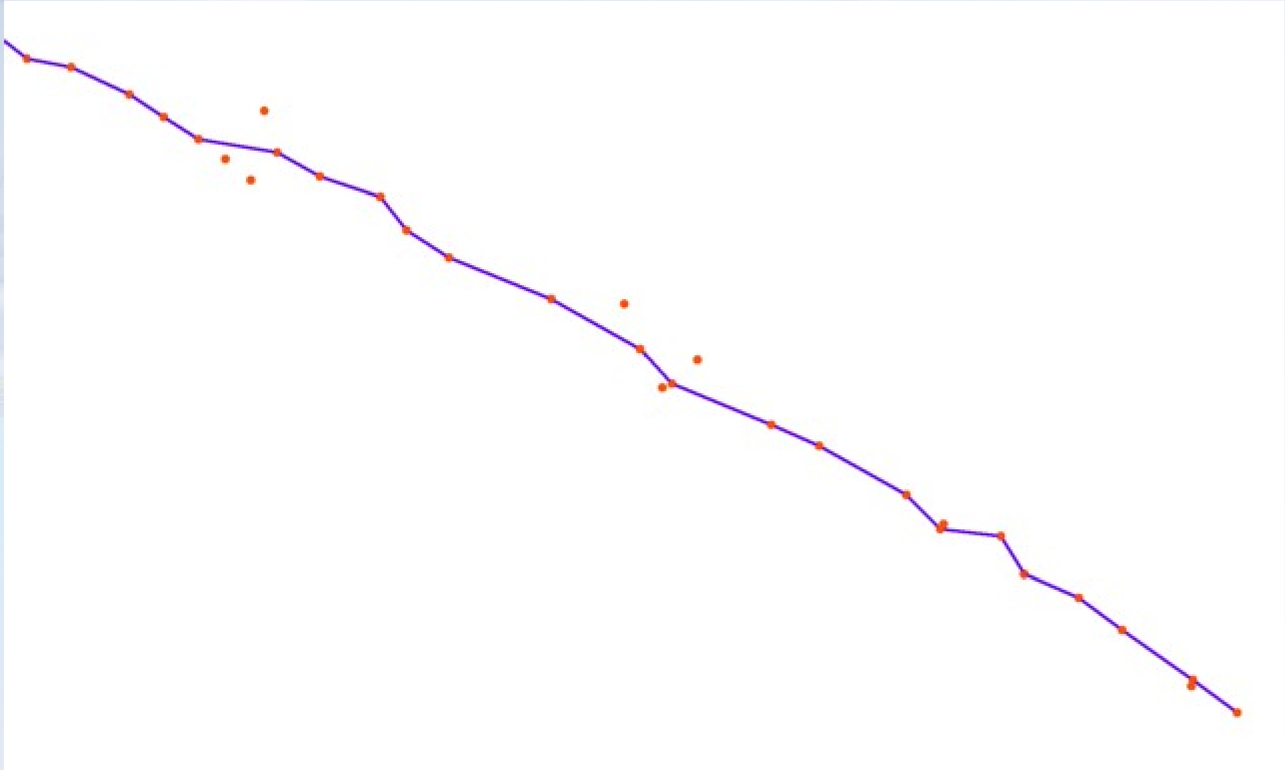


Illustration of Tracking

- Parallel track following can find the track several times.

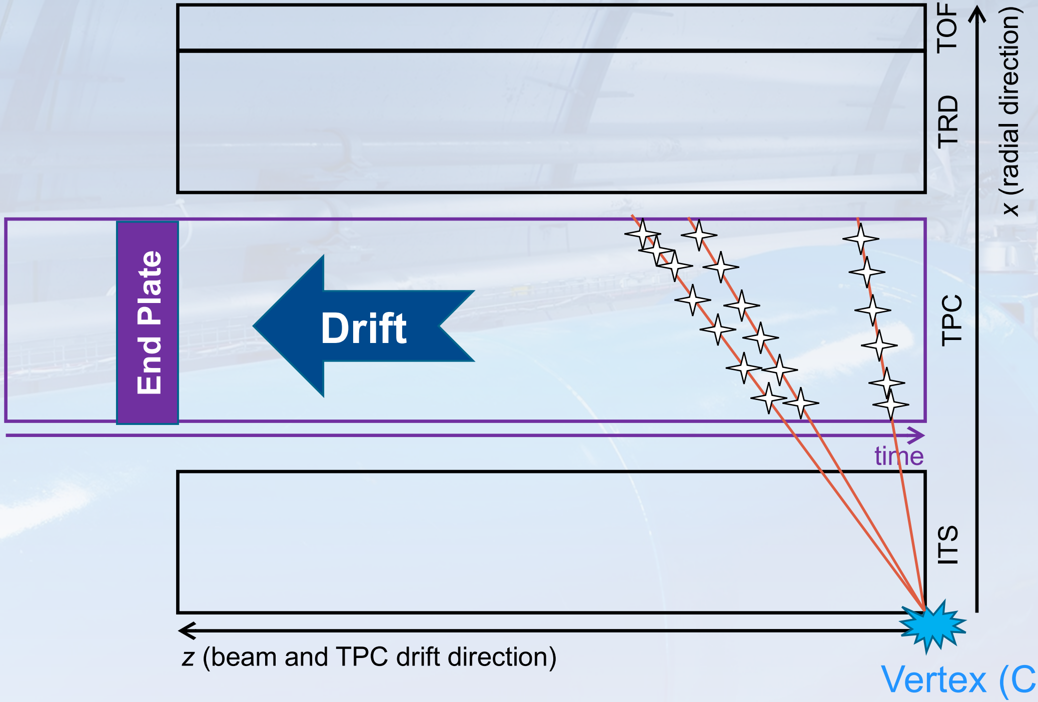
- In this case, an additional step selects the best instance of the track.

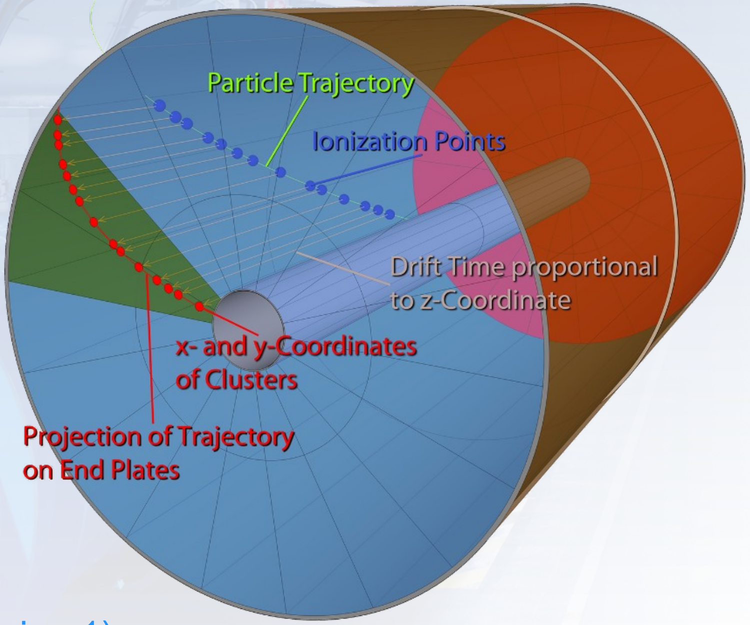


The tracking challenge

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.

- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**

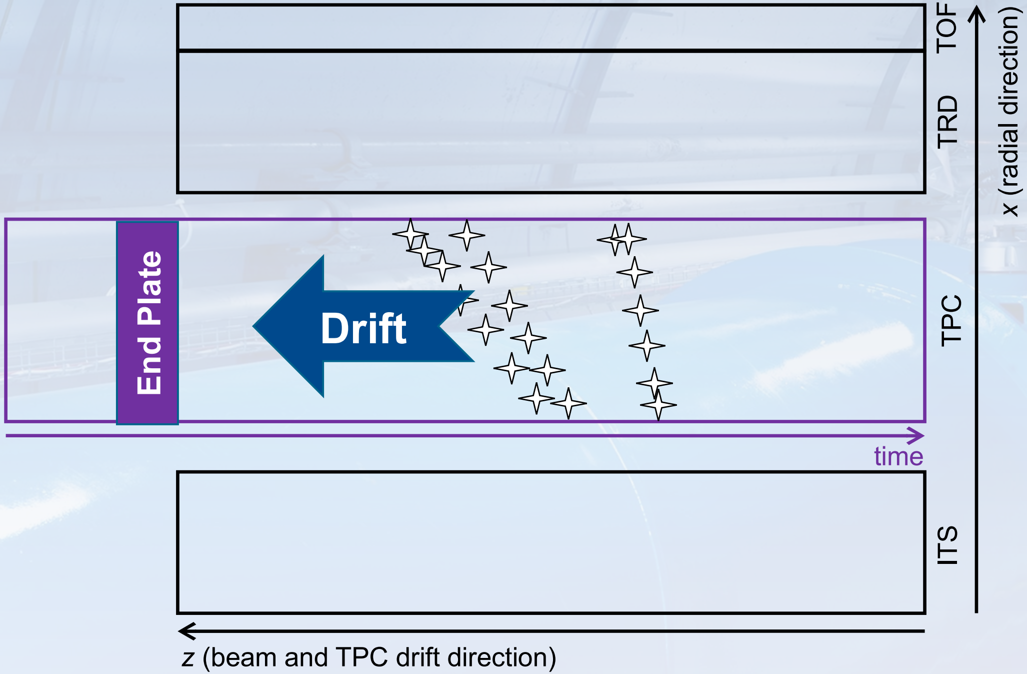




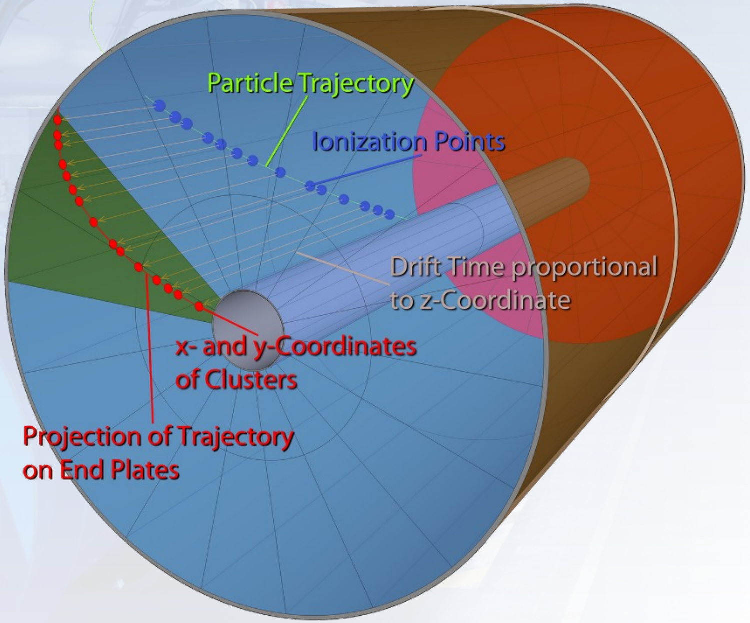
Particle Trajectory
Ionization Points
Drift Time proportional to z-Coordinate
x- and y-Coordinates of Clusters
Projection of Trajectory on End Plates

The tracking challenge

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.



- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**

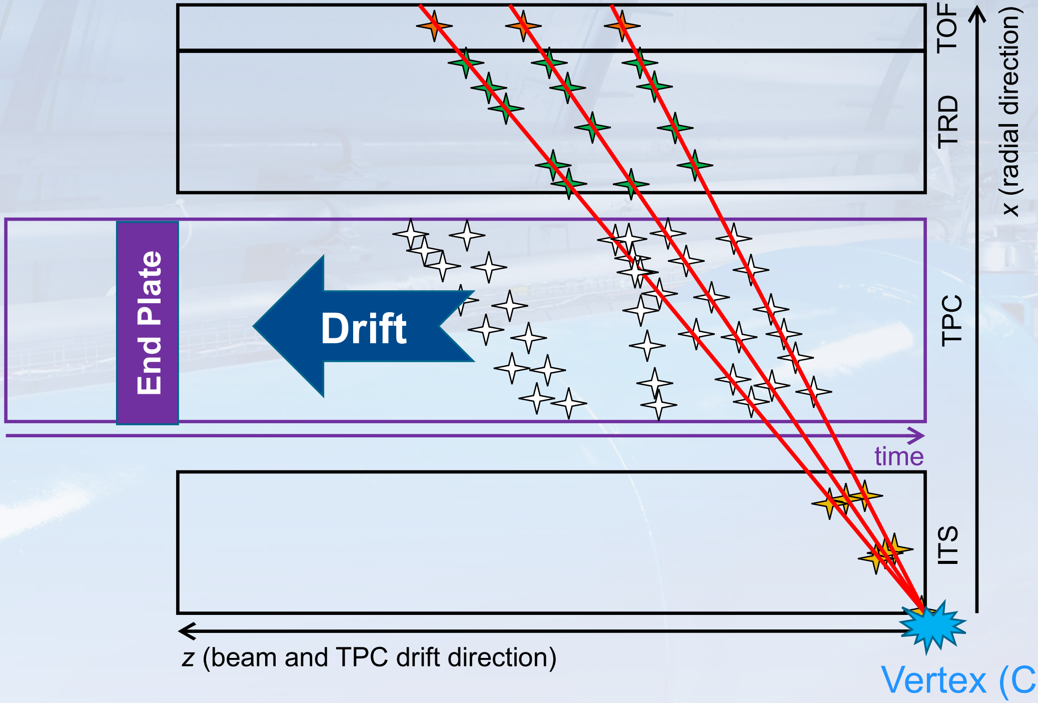


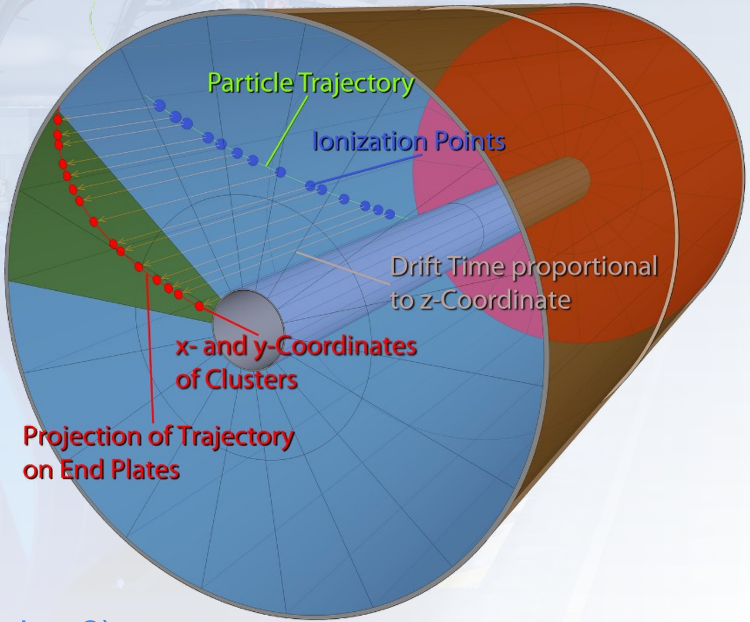
Particle Trajectory
Ionization Points
Drift Time proportional to z-Coordinate
x- and y-Coordinates of Clusters
Projection of Trajectory on End Plates

The tracking challenge

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.

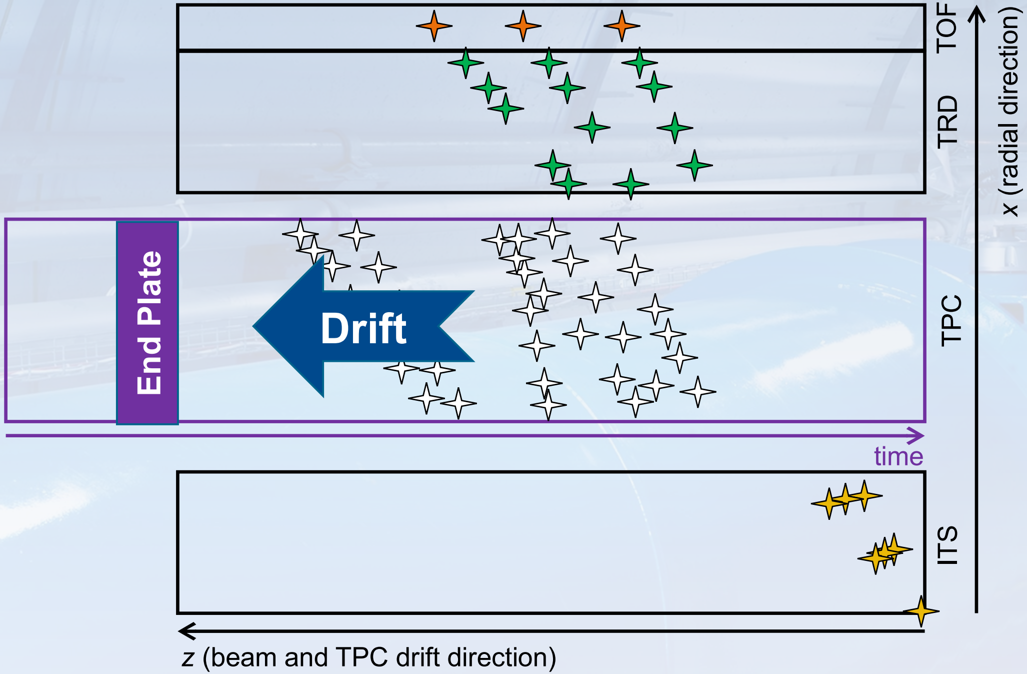
- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**



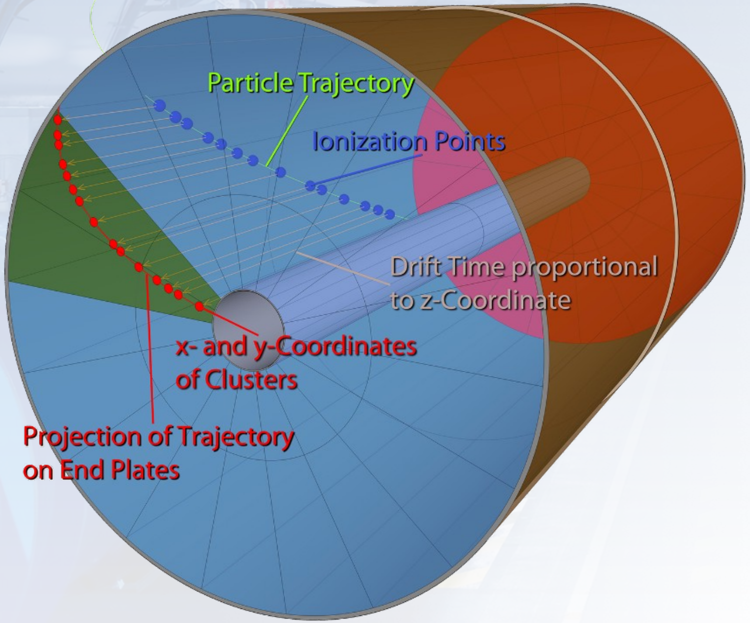

Particle Trajectory
Ionization Points
Drift Time proportional to z-Coordinate
x- and y-Coordinates of Clusters
Projection of Trajectory on End Plates

The tracking challenge

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.



- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**

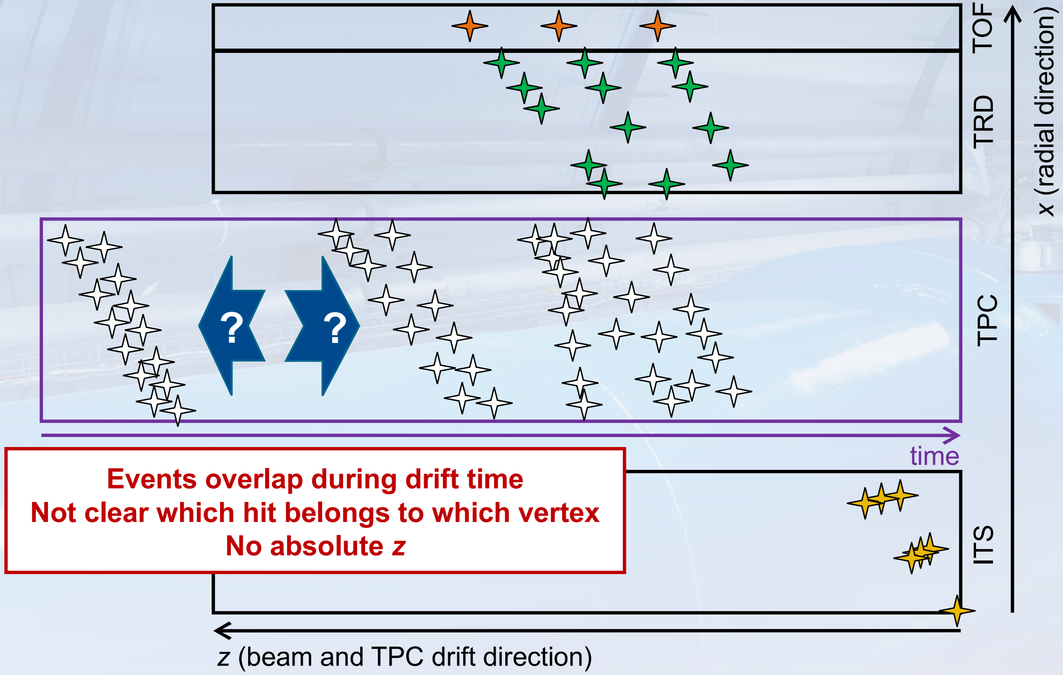


Particle Trajectory
Ionization Points
Drift Time proportional to z-Coordinate
x- and y-Coordinates of Clusters
Projection of Trajectory on End Plates

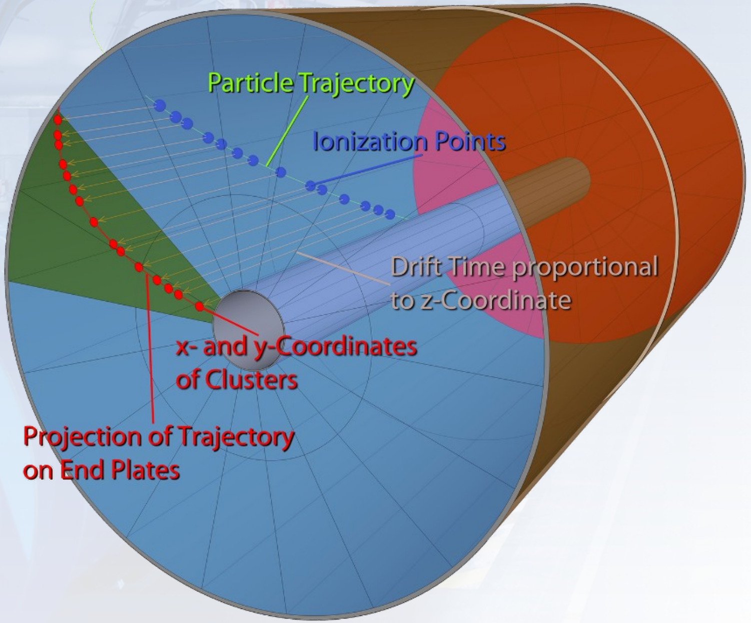
The tracking challenge

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.

- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**



Events overlap during drift time
Not clear which hit belongs to which vertex
No absolute *z*

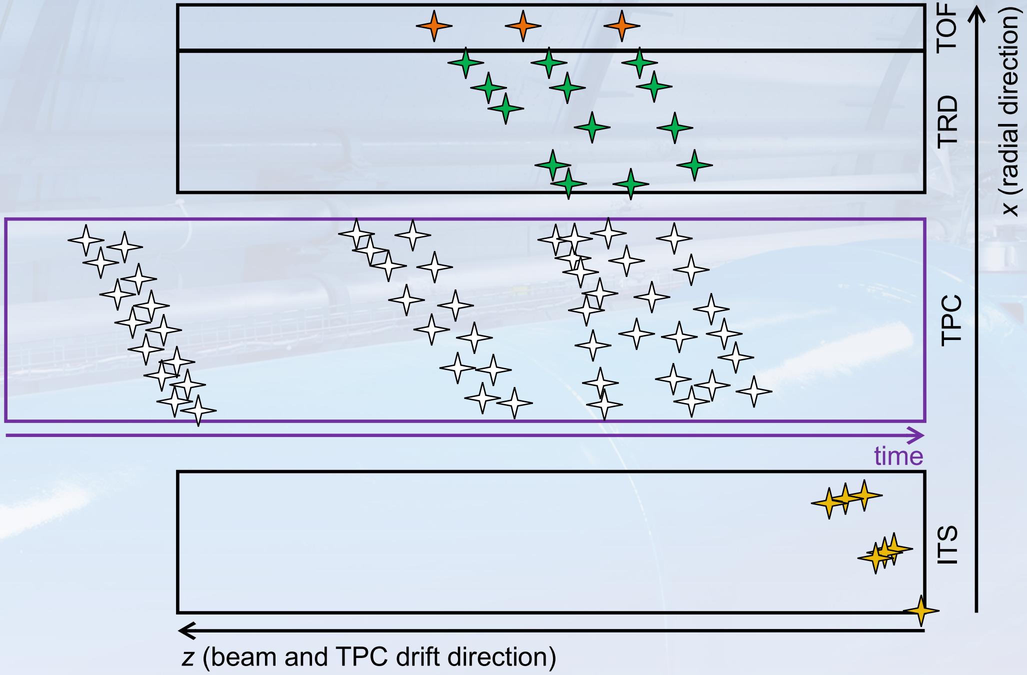


Particle Trajectory
Ionization Points
Drift Time proportional to z-Coordinate
x- and y-Coordinates of Clusters
Projection of Trajectory on End Plates

The tracking challenge

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.

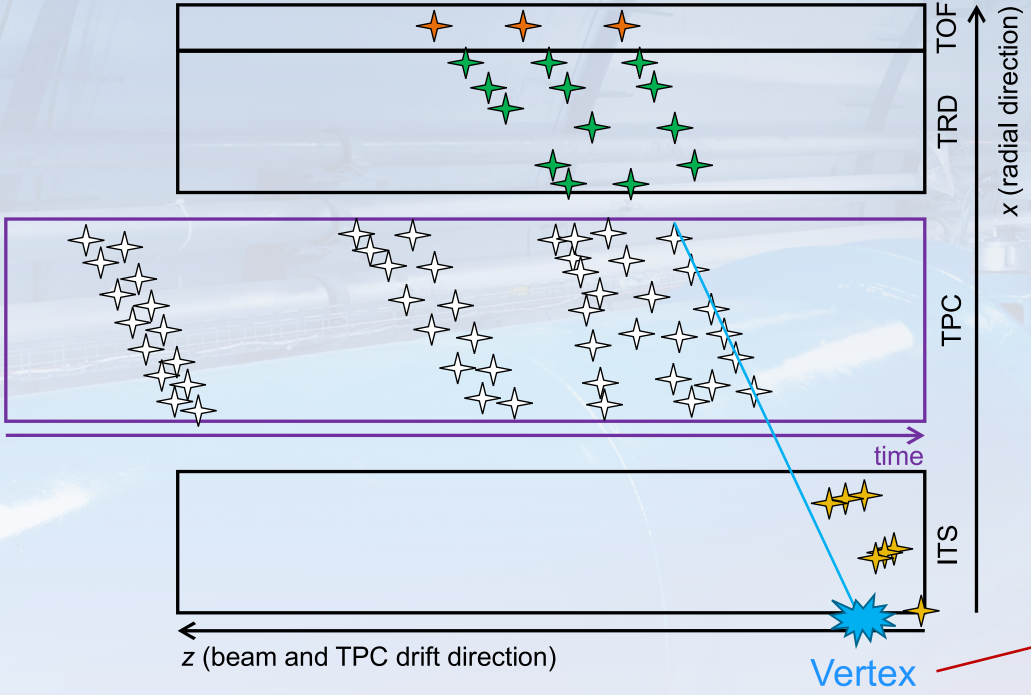
- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**



The tracking challenge

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.

- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**

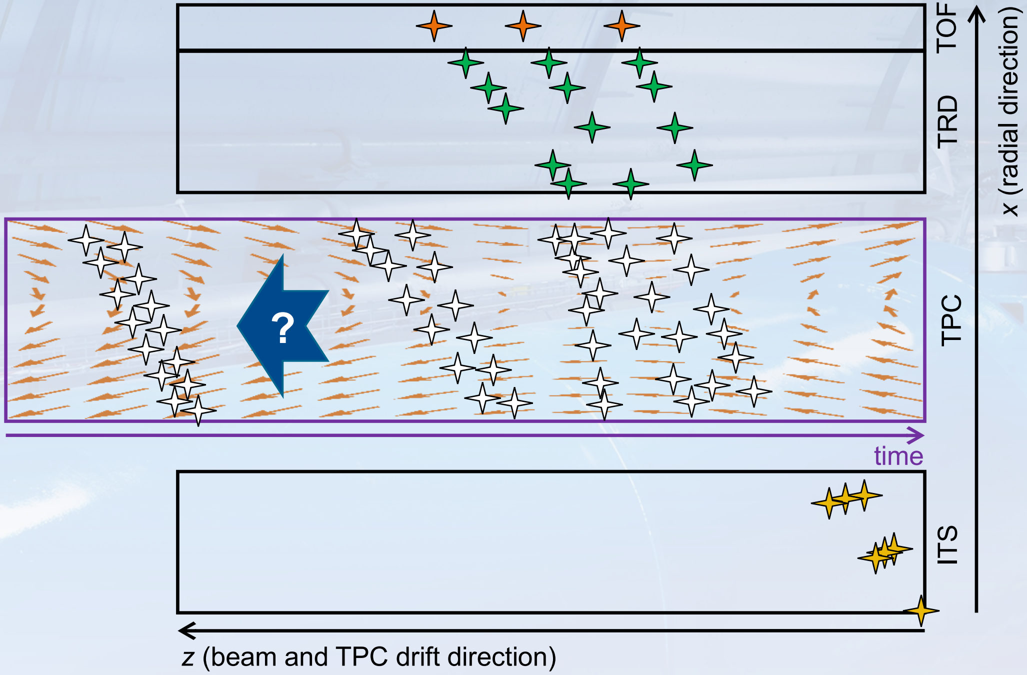


$z \sim t - t_{\mathrm{Vertex}}$

→ Need to identify the primary vertex, before assigning final *z* to cluster.

The tracking challenge

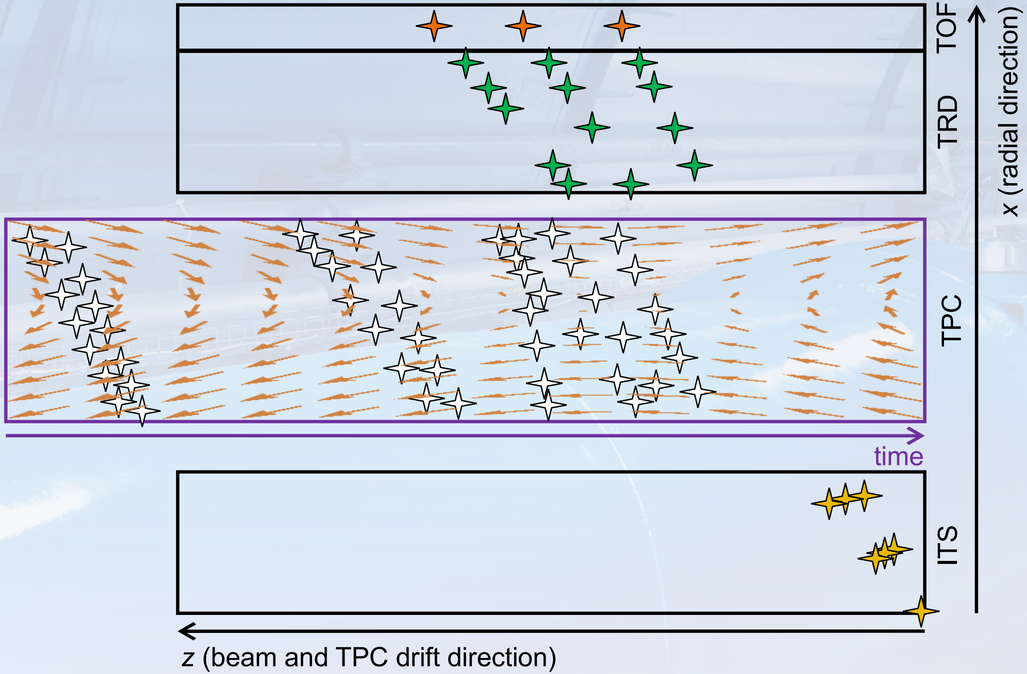
- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.



- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**
- **GEM amplification produces ions that deflect the electrons during the drift. The correction of these space-charge distortions requires the absolute *z* position.**

The tracking challenge

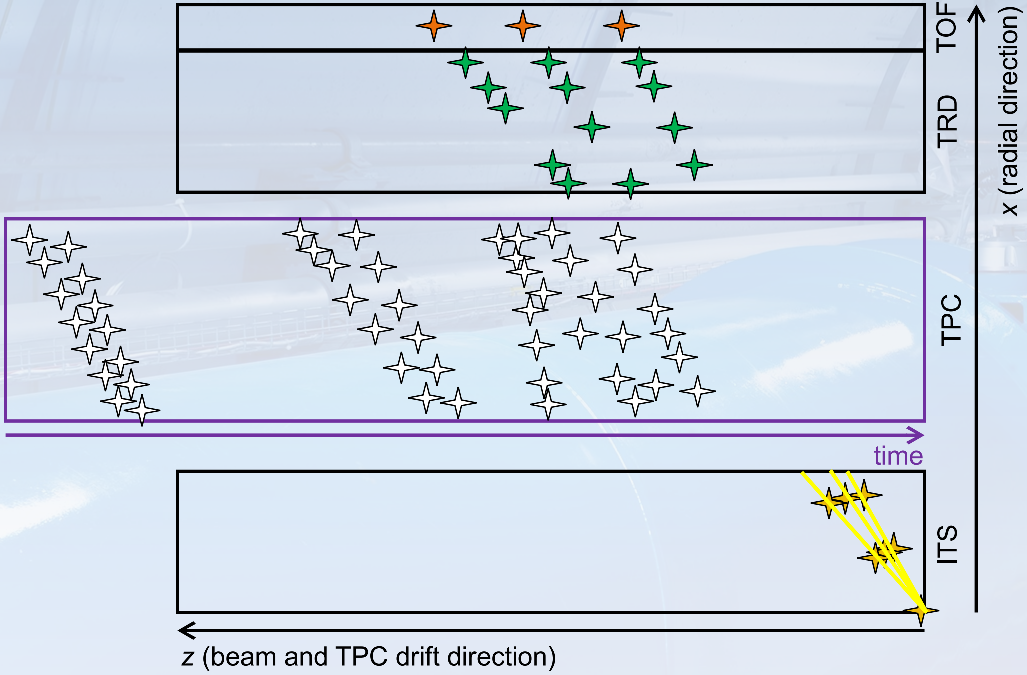
- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.



- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**
- **GEM amplification produces ions that deflect the electrons during the drift. The correction of these space-charge distortions requires the absolute *z* position.**

The tracking challenge – How the tracking will work

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.

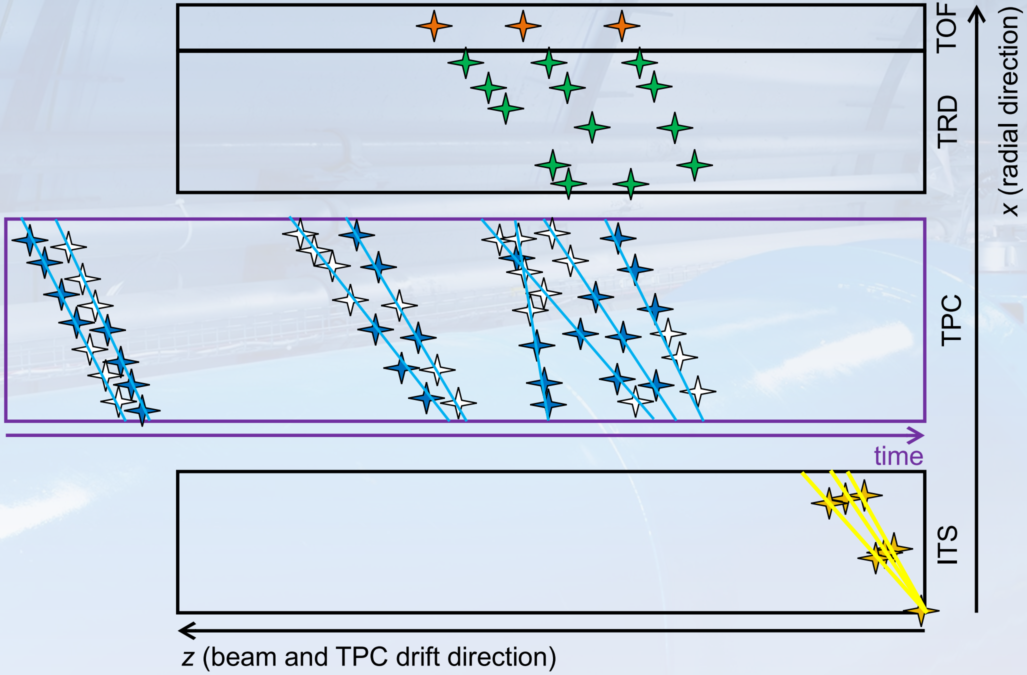


- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**
- **GEM amplification produces ions that deflect the electrons during the drift. The correction of these space-charge distortions requires the absolute *z* position.**

- Standalone ITS tracking.

The tracking challenge – How the tracking will work

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.



- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**
- **GEM amplification produces ions that deflect the electrons during the drift. The correction of these space-charge distortions requires the absolute *z* position.**

- Standalone ITS tracking.
- Standalone TPC tracking, scaling *t* linearly to an arbitrary *z*.

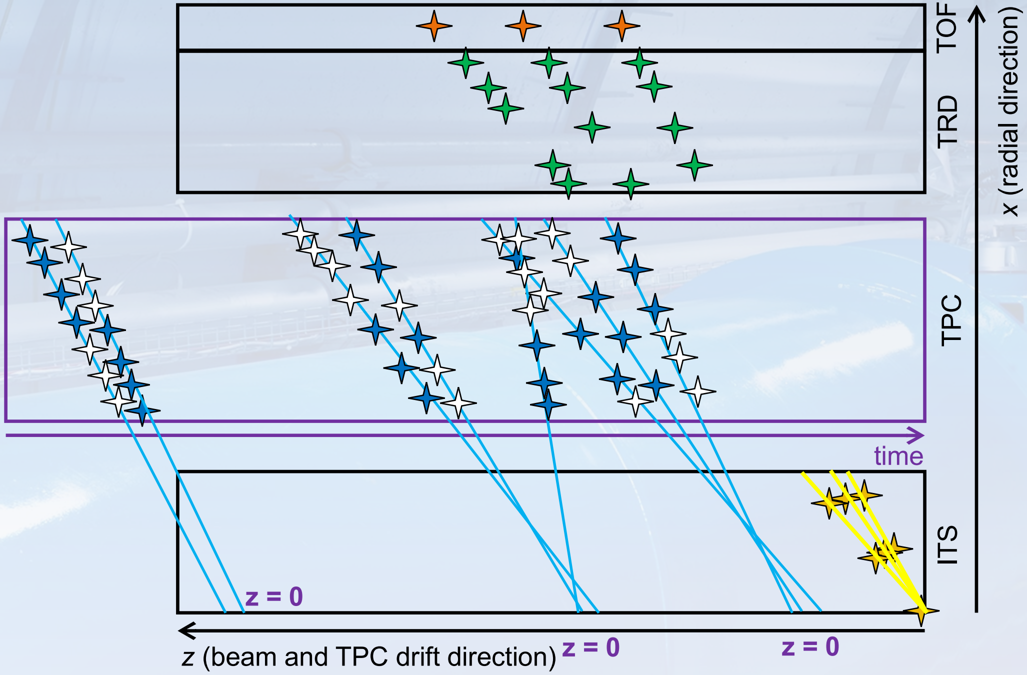
Precise tracking needs *z* for:

- Cluster error parameterization
- Inhomogeneous B-field
- Distortion correction

Effects smooth ☾ irrelevant for initial trackletting

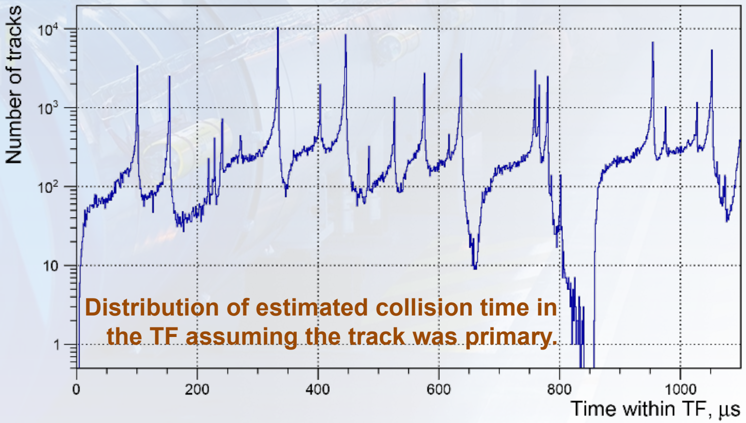
The tracking challenge – How the tracking will work

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.



- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**
- **GEM amplification produces ions that deflect the electrons during the drift. The correction of these space-charge distortions requires the absolute *z* position.**

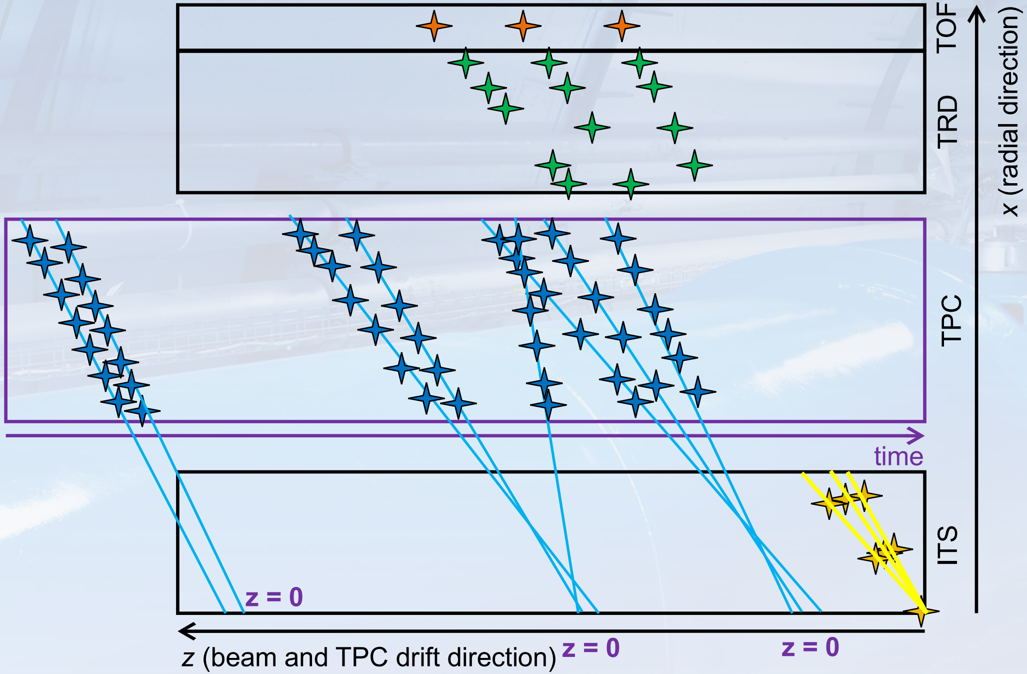
- Standalone ITS tracking.
- Standalone TPC tracking, scaling *t* linearly to an arbitrary *z*.
- Extrapolate to *x* = 0, define *z* = 0 as if the track was primary.



Distribution of estimated collision time in the TF assuming the track was primary.

The tracking challenge – How the tracking will work

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.

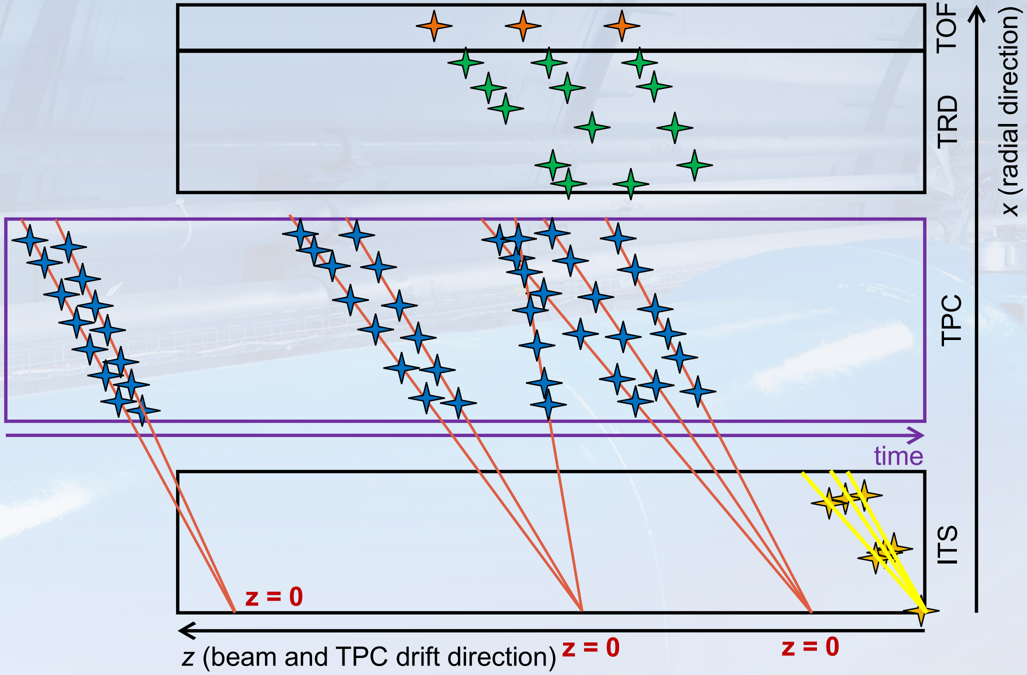


- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**
- **GEM amplification produces ions that deflect the electrons during the drift. The correction of these space-charge distortions requires the absolute *z* position.**

- Standalone ITS tracking.
- Standalone TPC tracking, scaling *t* linearly to an arbitrary *z*.
- Extrapolate to *x* = 0, define *z* = 0 as if the track was primary.
- Track following to find missing clusters. For cluster error parameterization, distortions, and B-field, shift the track such that *z* = 0 at *x* = 0.

The tracking challenge – How the tracking will work

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.

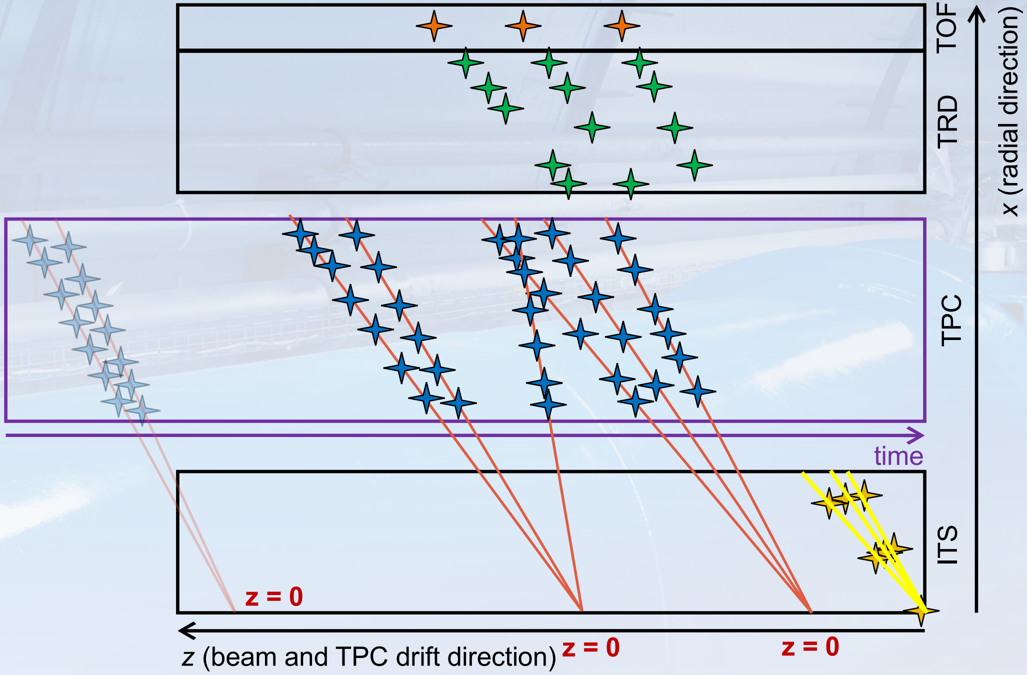


- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**
- **GEM amplification produces ions that deflect the electrons during the drift. The correction of these space-charge distortions requires the absolute *z* position.**

- Standalone ITS tracking.
- Standalone TPC tracking, scaling *t* linearly to an arbitrary *z*.
- Extrapolate to *x* = 0, define *z* = 0 as if the track was primary.
- Track following to find missing clusters. For cluster error parameterization, distortions, and B-field, shift the track such that *z* = 0 at *x* = 0.
- Refine *z* = 0 estimate, refit track with best precision

The tracking challenge – How the tracking will work

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.

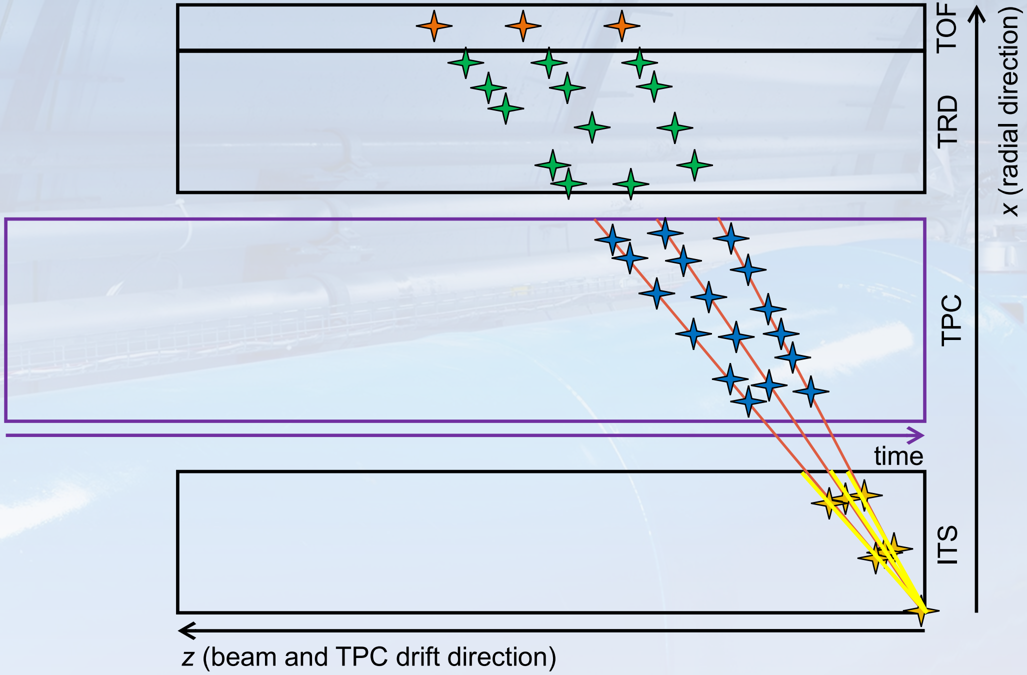


- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**
- **GEM amplification produces ions that deflect the electrons during the drift. The correction of these space-charge distortions requires the absolute *z* position.**

- Standalone ITS tracking.
- Standalone TPC tracking, scaling *t* linearly to an arbitrary *z*.
- Extrapolate to *x* = 0, define *z* = 0 as if the track was primary.
- Track following to find missing clusters. For cluster error parameterization, distortions, and B-field, shift the track such that *z* = 0 at *x* = 0.
- Refine *z* = 0 estimate, refit track with best precision
- For the tracks in one ITS read out frame, select all TPC tracks with a compatible time (from *z* = 0 estimate).

The tracking challenge – How the tracking will work

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.

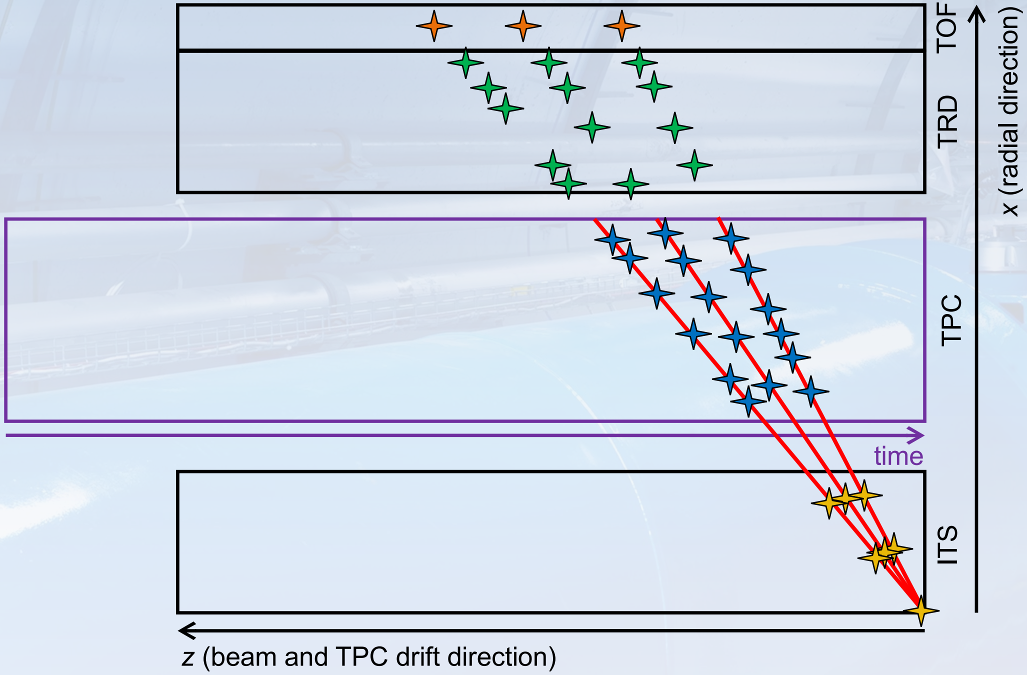


- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**
- **GEM amplification produces ions that deflect the electrons during the drift. The correction of these space-charge distortions requires the absolute *z* position.**

- Standalone ITS tracking.
- Standalone TPC tracking, scaling *t* linearly to an arbitrary *z*.
- Extrapolate to *x* = 0, define *z* = 0 as if the track was primary.
- Track following to find missing clusters. For cluster error parameterization, distortions, and B-field, shift the track such that *z* = 0 at *x* = 0.
- Refine *z* = 0 estimate, refit track with best precision
- For the tracks in one ITS read out frame, select all TPC tracks with a compatible time (from *z* = 0 estimate).
- Match TPC track to ITS track, fixing z-position and time of the TPC track.

The tracking challenge – How the tracking will work

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.

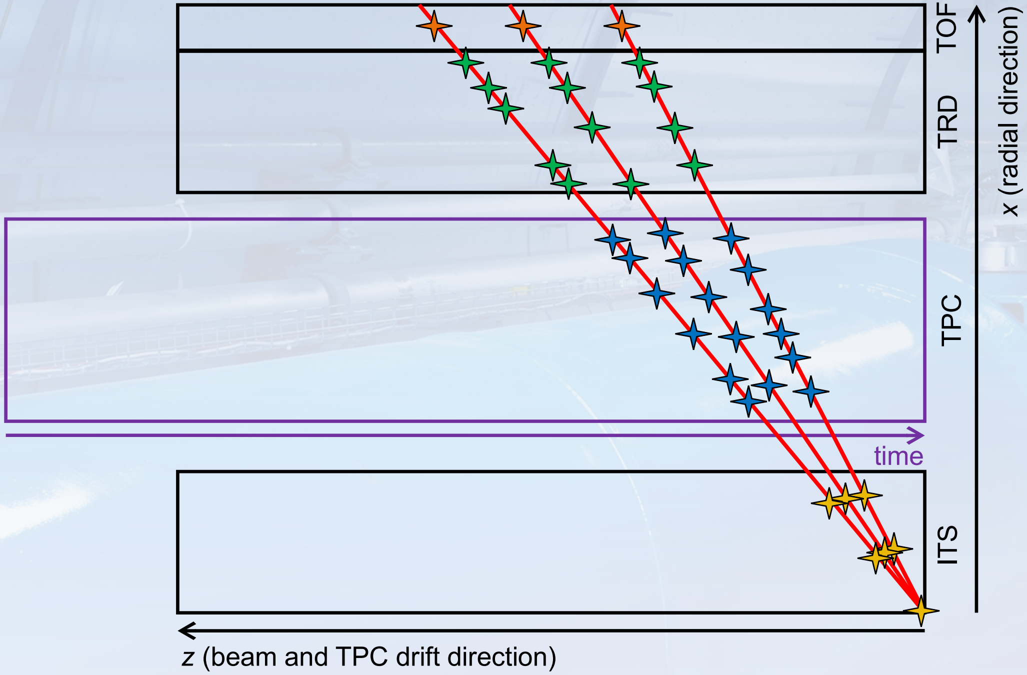


- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**
- **GEM amplification produces ions that deflect the electrons during the drift. The correction of these space-charge distortions requires the absolute *z* position.**

- Standalone ITS tracking.
- Standalone TPC tracking, scaling *t* linearly to an arbitrary *z*.
- Extrapolate to *x* = 0, define *z* = 0 as if the track was primary.
- Track following to find missing clusters. For cluster error parameterization, distortions, and B-field, shift the track such that *z* = 0 at *x* = 0.
- Refine *z* = 0 estimate, refit track with best precision
- For the tracks in one ITS read out frame, select all TPC tracks with a compatible time (from *z* = 0 estimate).
- Match TPC track to ITS track, fixing z-position and time of the TPC track.
- Refit ITS + TPC track outwards.

The tracking challenge – How the tracking will work

- **Tracking continuous data…**
  - The TPC sees **multiple overlapped collisions** (shifted in time).
  - Other detectors know the (rough) time of the collision.



- **Problem: TPC clusters have no defined *z*-position but only a time. They can be shifted in *z* arbitrarily.**
- **GEM amplification produces ions that deflect the electrons during the drift. The correction of these space-charge distortions requires the absolute *z* position.**

- Standalone ITS tracking.
- Standalone TPC tracking, scaling *t* linearly to an arbitrary *z*.
- Extrapolate to *x* = 0, define *z* = 0 as if the track was primary.
- Track following to find missing clusters. For cluster error parameterization, distortions, and B-field, shift the track such that *z* = 0 at *x* = 0.
- Refine *z* = 0 estimate, refit track with best precision
- For the tracks in one ITS read out frame, select all TPC tracks with a compatible time (from *z* = 0 estimate).
- Match TPC track to ITS track, fixing z-position and time of the TPC track.
- Refit ITS + TPC track outwards.
- Prolong into TRD / TOF.

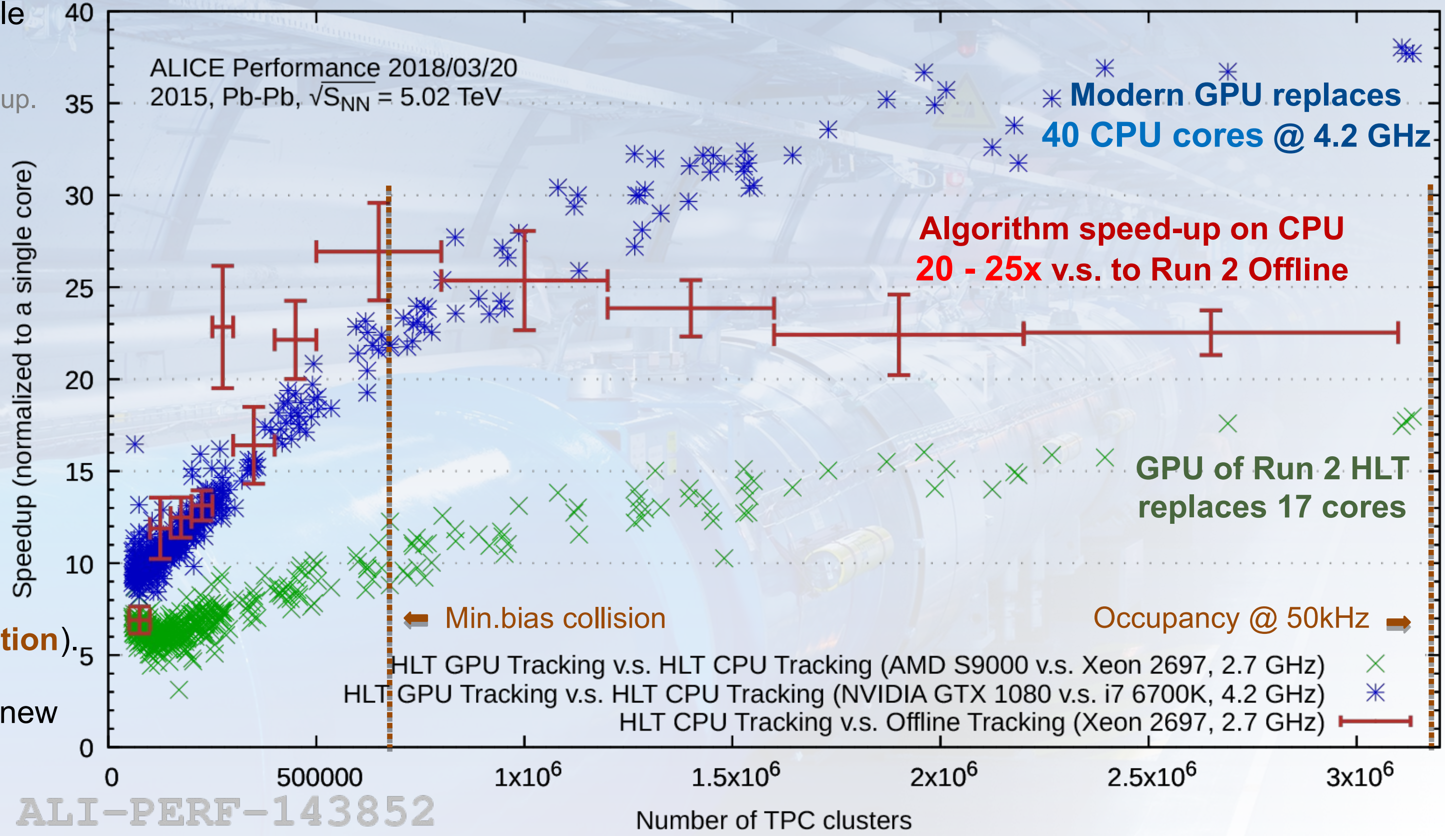
TPC Tracking performance

- Speed-up normalized to single CPU core.
    - Red curve: algorithm speed-up.
    - Other curves: GPU v.s. CPU speed-up corrected for CPU resources.
        - How many cores does the GPU replace.

- Significant gain with newer GPU (blue v.s. green).

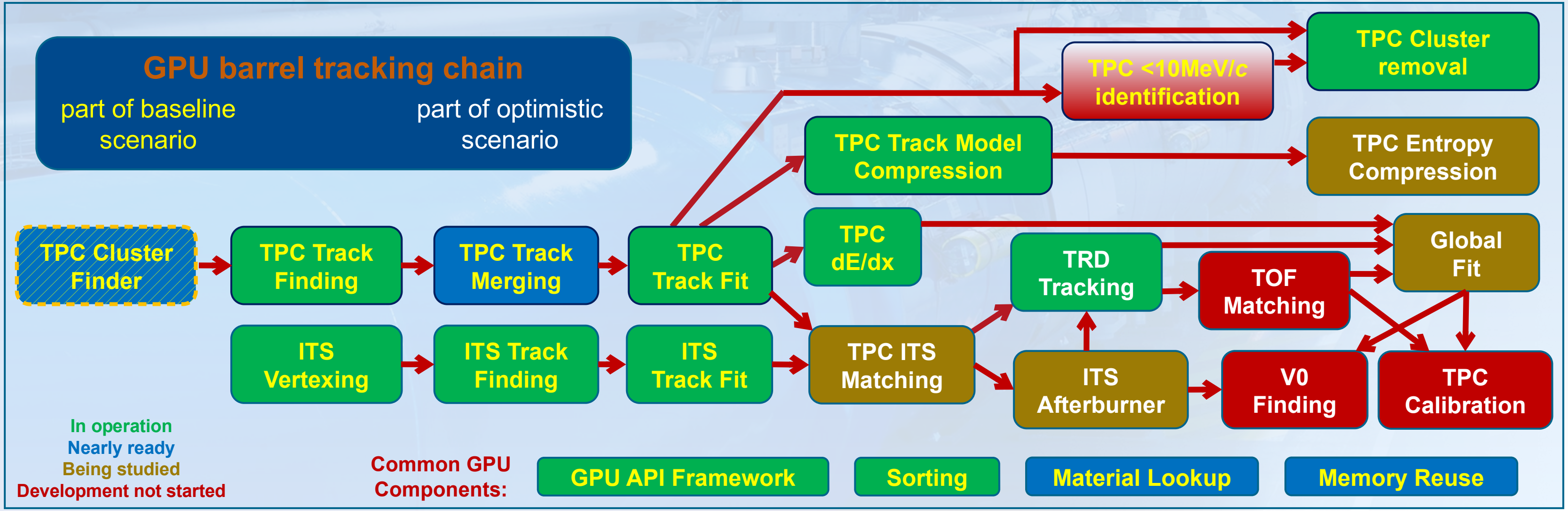
- GPU with Run 3 algorithm replaces **> 800 CPU cores** Running Run 2 algorithm. (**blue * red**). (**at same efficiency / resolution**).

- We see ~**30%** speedup with new GPU generation (RTX 2080 v.s. GTX 1080)



Reconstruction steps on GPU (Barrel Tracking)

- **Status of reconstruction steps on GPU:**
  - All TPC steps during synchronous reconstruction are **required** on the GPU.
  - Synchronous ITS tracking and TPC dE/*dx* in good shape, thus considered **baseline** on the GPU.
  - Remaining steps in tracking chain part of **optimistic scenario**, being ported step by step to GPU.
    - Porting order follows topology of chain, to avoid unnecessary data transfer for ported steps – current blocker is **TPC ITS matching**.



GPU barrel tracking chain

part of baseline scenario

part of optimistic scenario

TPC Cluster Finder

TPC Track Finding

TPC Track Merging

TPC Track Fit

TPC dE/dx

TPC Track Model Compression

TPC Entropy Compression

TPC <10MeV/*c* identification

TPC Cluster removal

ITS Vertexing

ITS Track Finding

ITS Track Fit

TPC ITS Matching

TRD Tracking

ITS Afterburner

TOF Matching

V0 Finding

Global Fit

TPC Calibration

In operation

Nearly ready

Being studied

Development not started

Common GPU Components:

GPU API Framework

Sorting

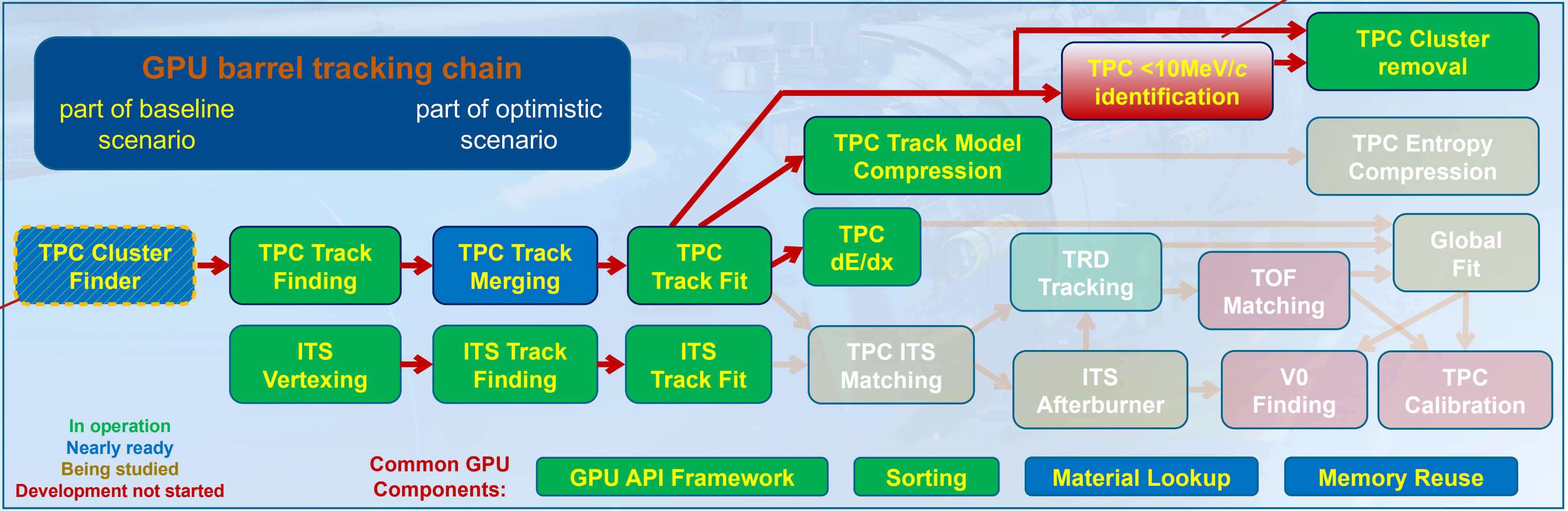
Material Lookup

Memory Reuse

Reconstruction steps on GPU (Barrel Tracking)

- **Status of reconstruction steps on GPU:**
  - Baseline scenario: all steps almost ready

Not clear if necessary on GPU, depends on rejection strategy (see later)



GPU barrel tracking chain

part of baseline scenario

part of optimistic scenario

TPC Cluster Finder

TPC Track Finding

TPC Track Merging

TPC Track Fit

ITS Vertexing

ITS Track Finding

ITS Track Fit

TPC Track Model Compression

TPC <10MeV/c identification

TPC Cluster removal

TPC Entropy Compression

TPC dE/dx

TPC ITS Matching

TRD Tracking

ITS Afterburner

TOF Matching

V0 Finding

Global Fit

TPC Calibration

In operation

Nearly ready

Being studied

Development not started

Common GPU Components:

GPU API Framework

Sorting

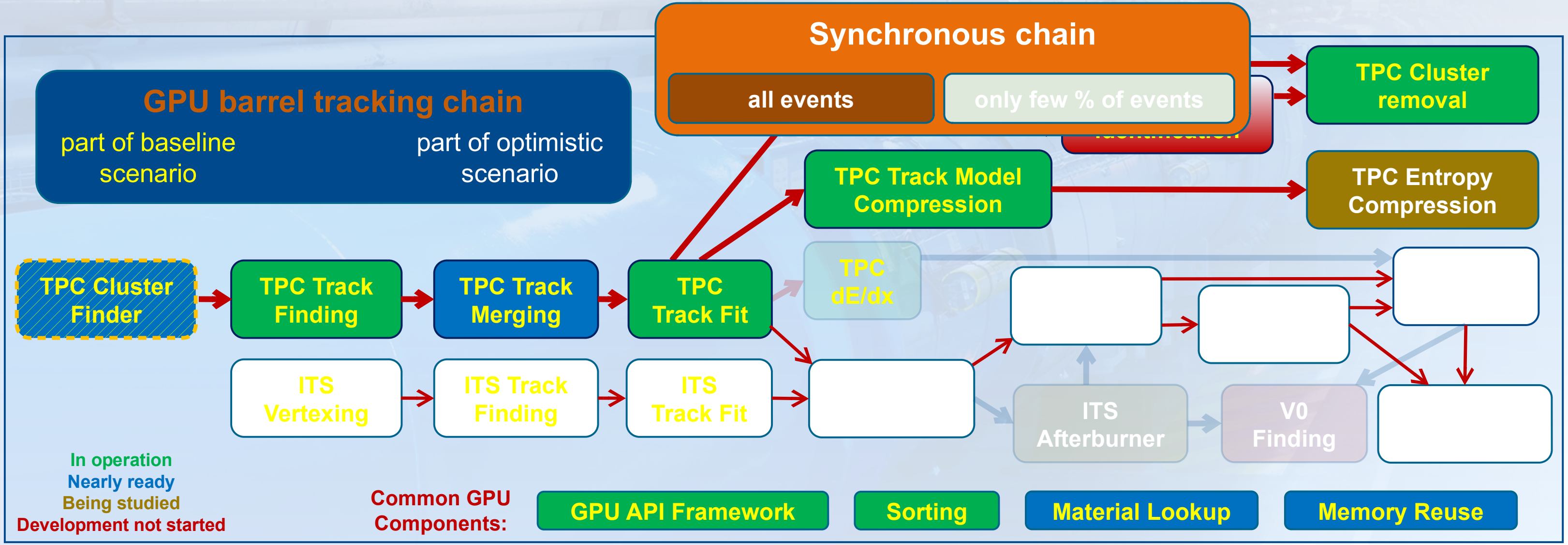
Material Lookup

Memory Reuse

New requirement arose few months ago, since clusterizer does not fit in FPGA.

Reconstruction steps on GPU (Barrel Tracking)

- **Status of reconstruction steps on GPU:**
  - Different reconstruction steps enabled in **synchronous** and **asynchronous** reconstruction.



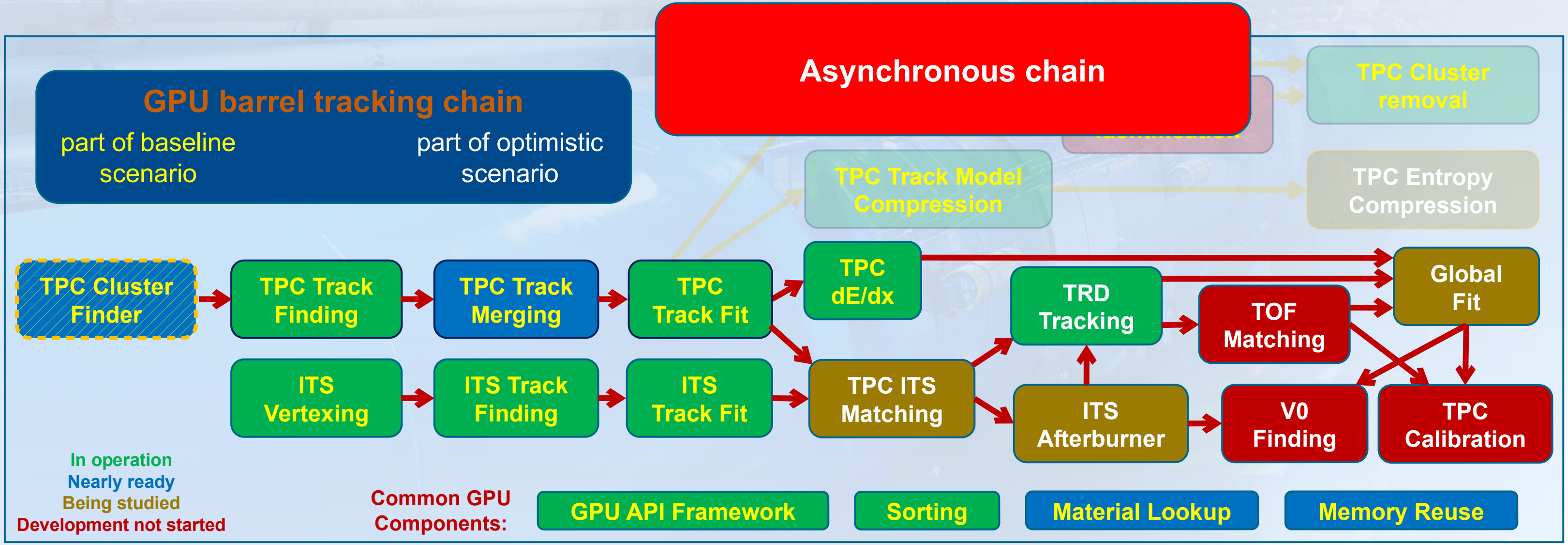
In operation
Nearly ready
Being studied
Development not started

Common GPU Components: GPU API Framework, Sorting, Material Lookup, Memory Reuse

Reconstruction steps on GPU (Barrel Tracking)

- **Status of reconstruction steps on GPU:**
  - Different reconstruction steps enabled in synchronous and asynchronous reconstruction.

TRD tracking / TPC calibration: see poster of Ole Schmidt
Space point calibration of the ALICE TPC with track residuals



Barrel Tracking Chain

- Many steps of barrel tracking run consecutively.

→ port consecutive steps to GPU, avoid data transfer.

- (Not strictly needed, dependend on data rates: TPC clusters most critical.)

Part of baseline scenario

In operation

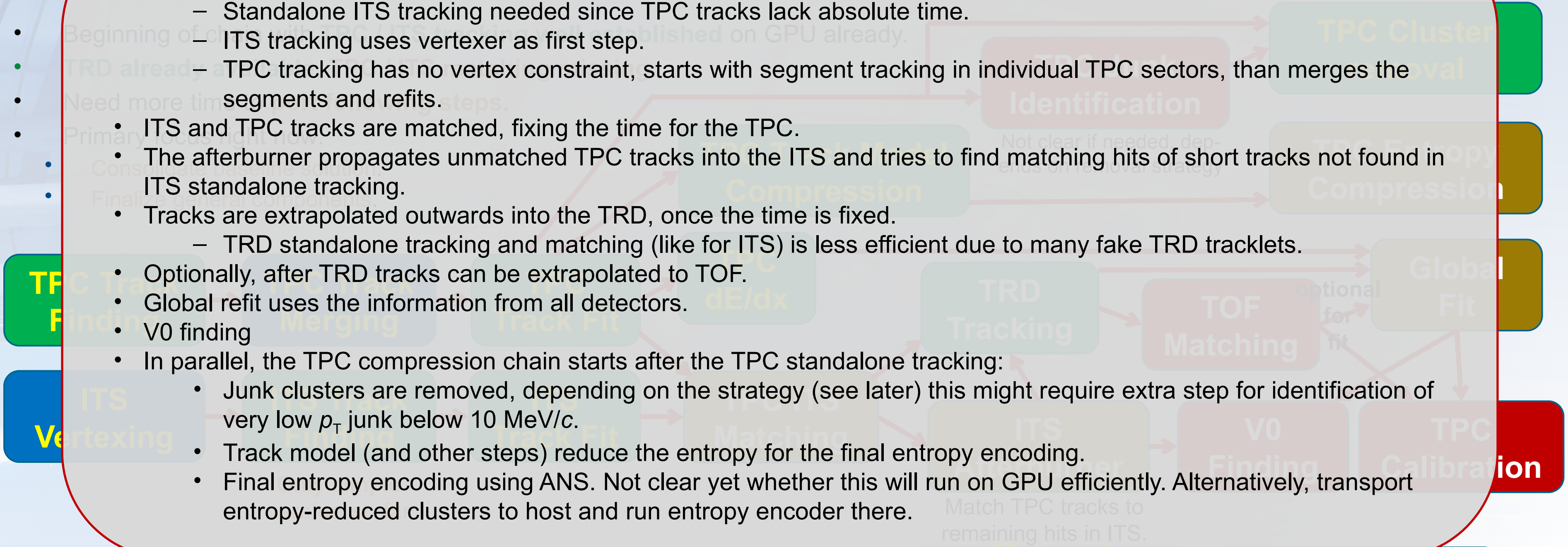
Nearly ready

Being studied

Not started

Strategy:

- Start with standalone TPC and ITS tracking.
  - Standalone ITS tracking needed since TPC tracks lack absolute time.
  - ITS tracking uses vertexer as first step.
  - TPC tracking has no vertex constraint, starts with segment tracking in individual TPC sectors, than merges the segments and refits.
- ITS and TPC tracks are matched, fixing the time for the TPC.
- The afterburner propagates unmatched TPC tracks into the ITS and tries to find matching hits of short tracks not found in ITS standalone tracking.
- Tracks are extrapolated outwards into the TRD, once the time is fixed.
  - TRD standalone tracking and matching (like for ITS) is less efficient due to many fake TRD tracklets.
- Optionally, after TRD tracks can be extrapolated to TOF.
- Global refit uses the information from all detectors.
- V0 finding
- In parallel, the TPC compression chain starts after the TPC standalone tracking:
  - Junk clusters are removed, depending on the strategy (see later) this might require extra step for identification of very low p_T junk below 10 MeV/c.
  - Track model (and other steps) reduce the entropy for the final entropy encoding.
  - Final entropy encoding using ANS. Not clear yet whether this will run on GPU efficiently. Alternatively, transport entropy-reduced clusters to host and run entropy encoder there.

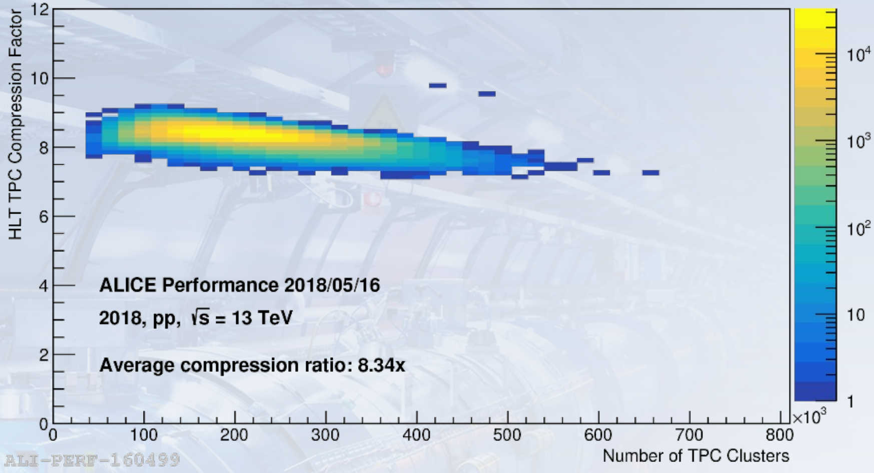


Common GPU Components: GPU API Framework | Sorting | Material Lookup | Memory Reuse

TPC Data Compression

- TPC Data compression involves 3 steps:
  1. Entropy reduction (Track model, logarithmic precision, etc.)
  2. Entropy encoding (Huffman, Arithmetic, ANS)
  3. Removal of tracks not used for physics.

- Steps 1 + 2 implemented for Run 2.
  - Current compression factor **8.3x**.

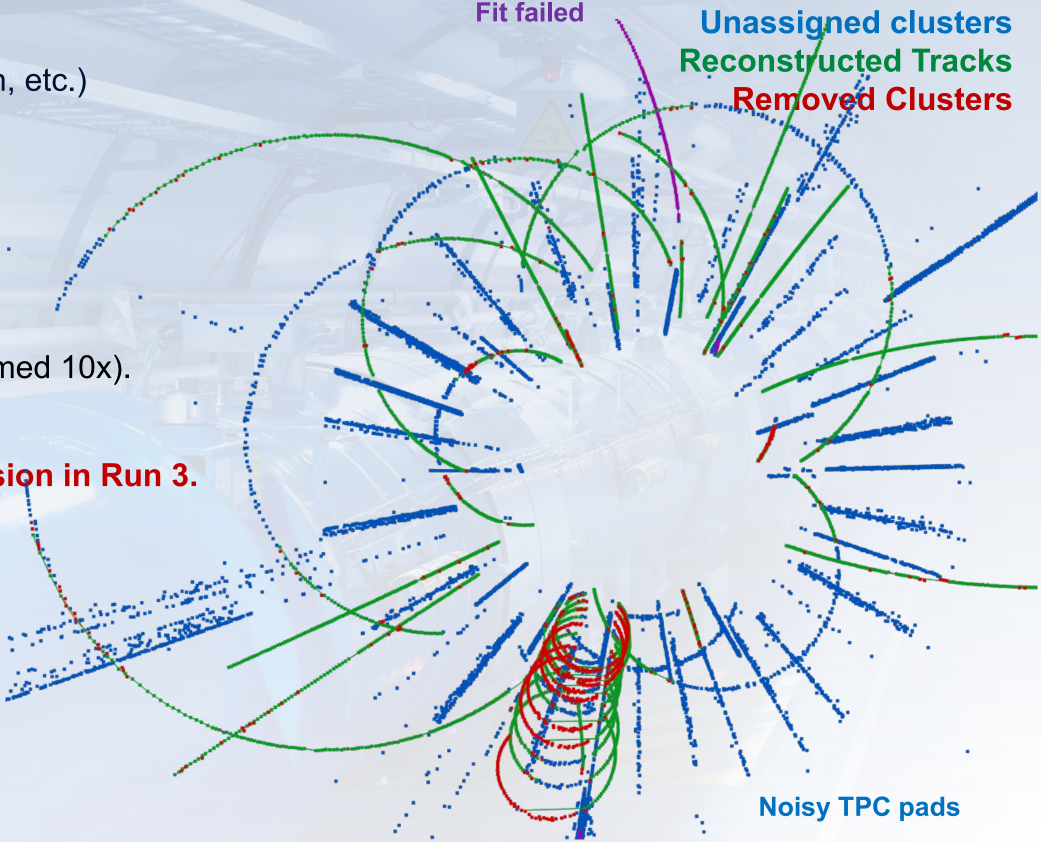


TPC Data Compression

- TPC Data compression involves 3 steps:
  1. Entropy reduction (Track model, logarithmic precision, etc.)
  2. Entropy encoding (Huffman, Arithmetic, ANS)
  3. Removal of tracks not used for physics.

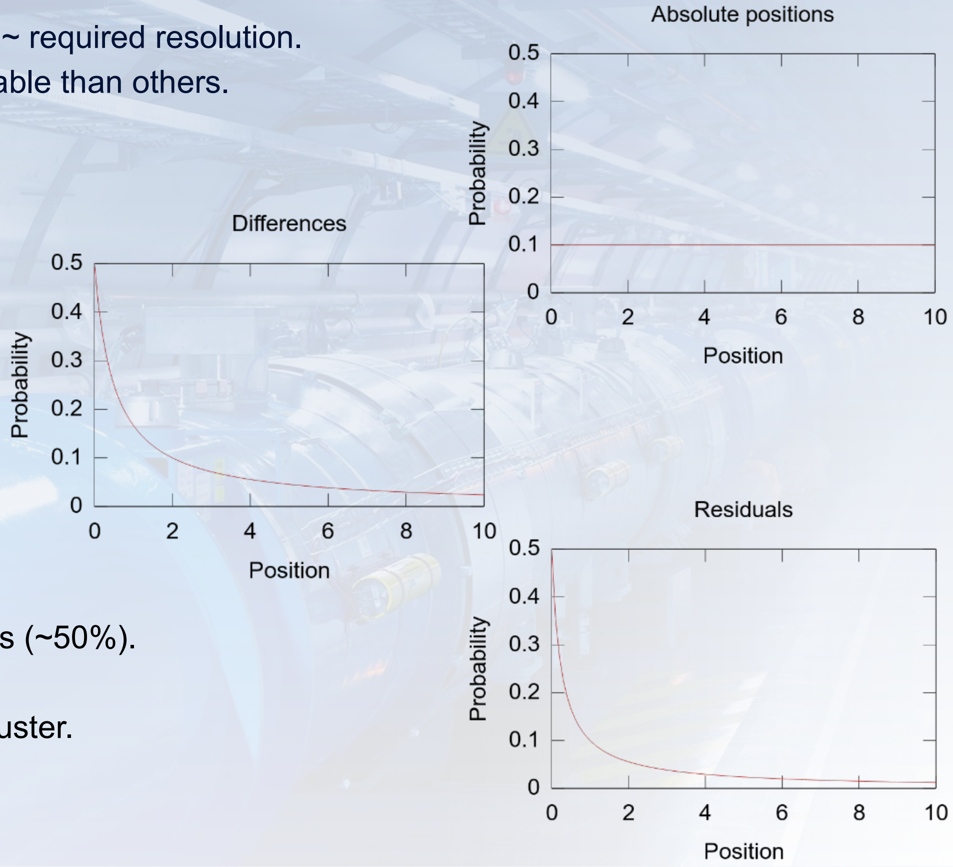
- Steps 1 + 2 implemented for Run 2.
  - Current compression factor **8.3x**.
  - Prototype for Run 3 achieves factor **9.1x** (TDR assumed 10x).

- **Step 3 must close the gap to the required compression in Run 3.**
  - Remove clusters from background / looping tracks.
    - Adjacent to low-p_T track < 50 MeV.
    - Adjacent to secondary leg of low-p_T track < 200 MeV.
    - Adjacent to any track with φ > 70° in the fit.
  - Protect clusters of physics tracks.
    - Not Adjacent to any physics-track (except φ > 70°).
- In addition:
  - Use reconstructed track quantities to reduce entropy.



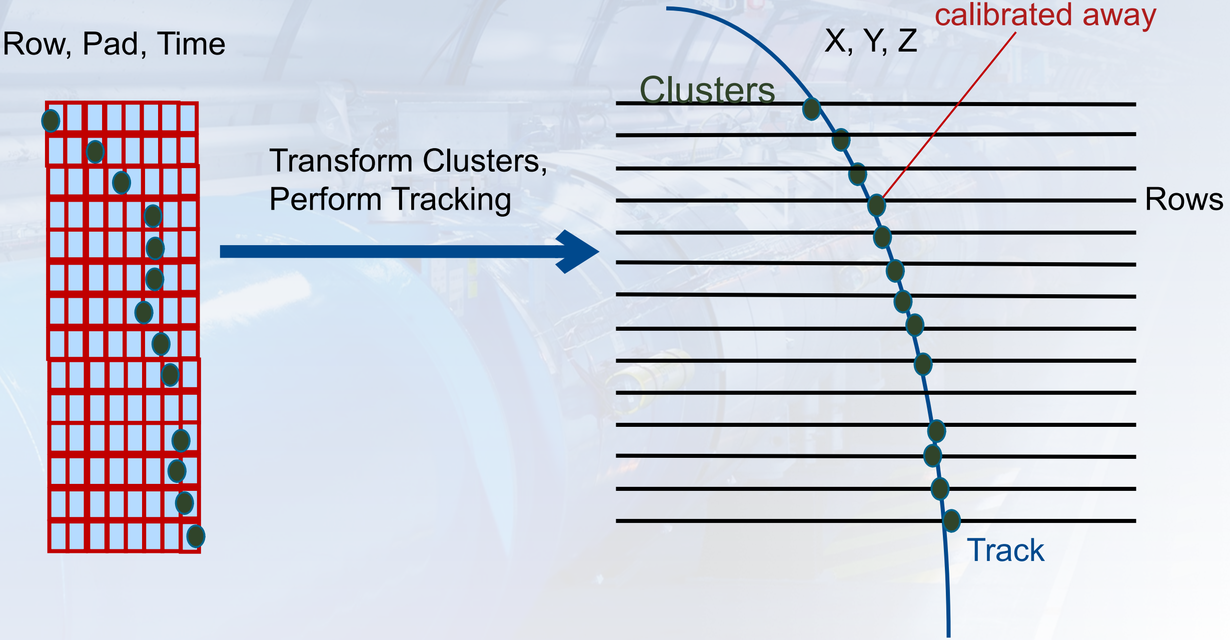
TPC Cluster Entropy Reduction

- Cluster Properties stored in integer format, such that 1 bit ~ required resolution.
- Exploit entropy encoding, i.e. some values are more probable than others.
  - Does not work well for absolute positions:
    - All positions have equal probability.
  - Can sort clusters and store only position differences.
    - Order is not important.
    - At high occupancy, all differences should be small.
  - With tracking, store only cluster to track residuals
    - Even less entropy for attached clusters.
    - Stick to differences for unattached clusters.
      - Unfortunately, less clusters stored as differences (~50%).
      - → Larger differences (~ factor 2 ☾ 1 bit).
      - → Need 0.5 more bits (1 * 50%) per unattached cluster.
      - Net compression still better.



TPC Cluster Entropy Reduction: Track Model

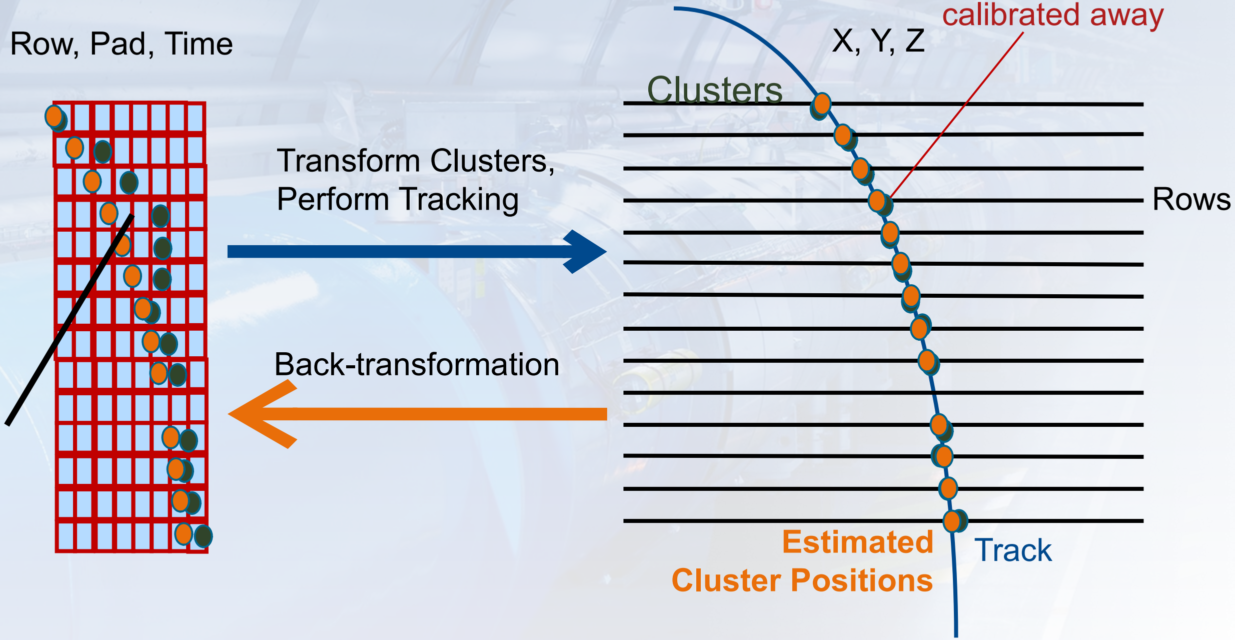
- **Minimize residuals**. (**Smaller entropy** ☾ **Better Huffman compression**.)



TPC Cluster Entropy Reduction: Track Model

- **Minimize residuals**. (**Smaller entropy ☾ Better Huffman compression**.)
  - **Constraint**: Clusters shall be stored in native TPC coordinates (Row, Pad, Time), independent from calibration.

- **Problems**:
  - Helix prolongation yields **large residuals** ☾ inefficient compression.
    - Does not account for space charge distortions.
  - Linear back-transformation **cannot revert transformation** based on full calibration.



Row, Pad, Time

Transform Clusters, Perform Tracking

Back-transformation

Large residuals to raw coordinates

X, Y, Z

Local distortions calibrated away

Clusters

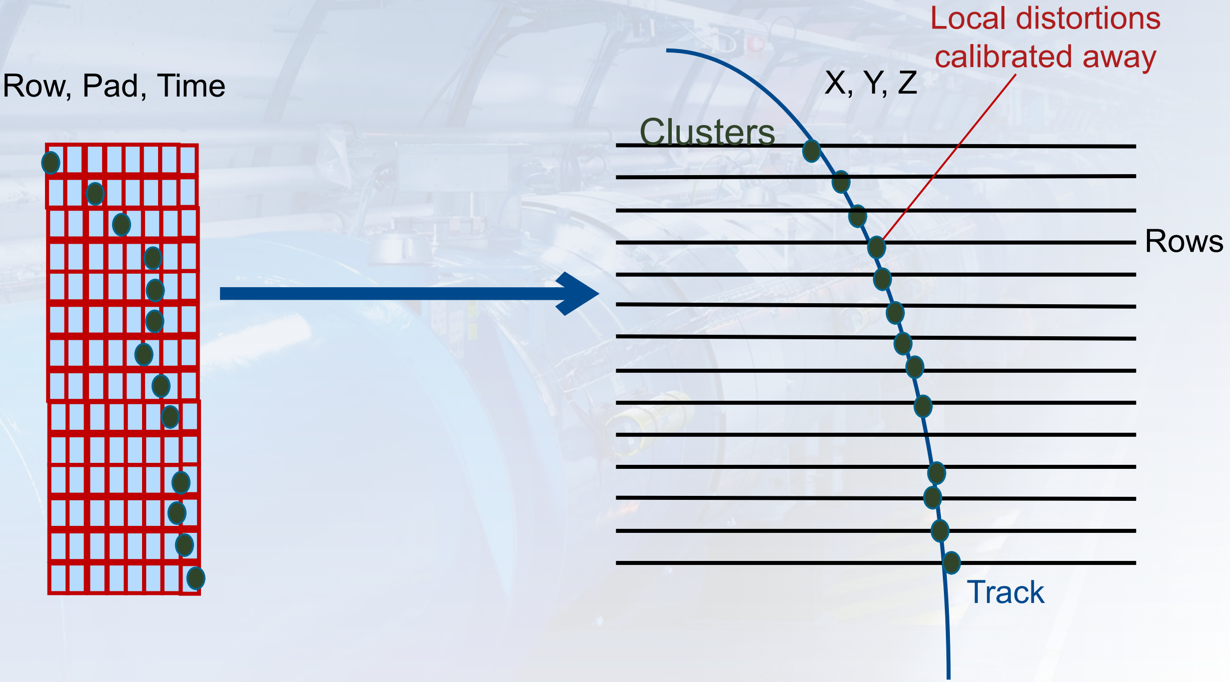
Rows

Estimated Cluster Positions

Track

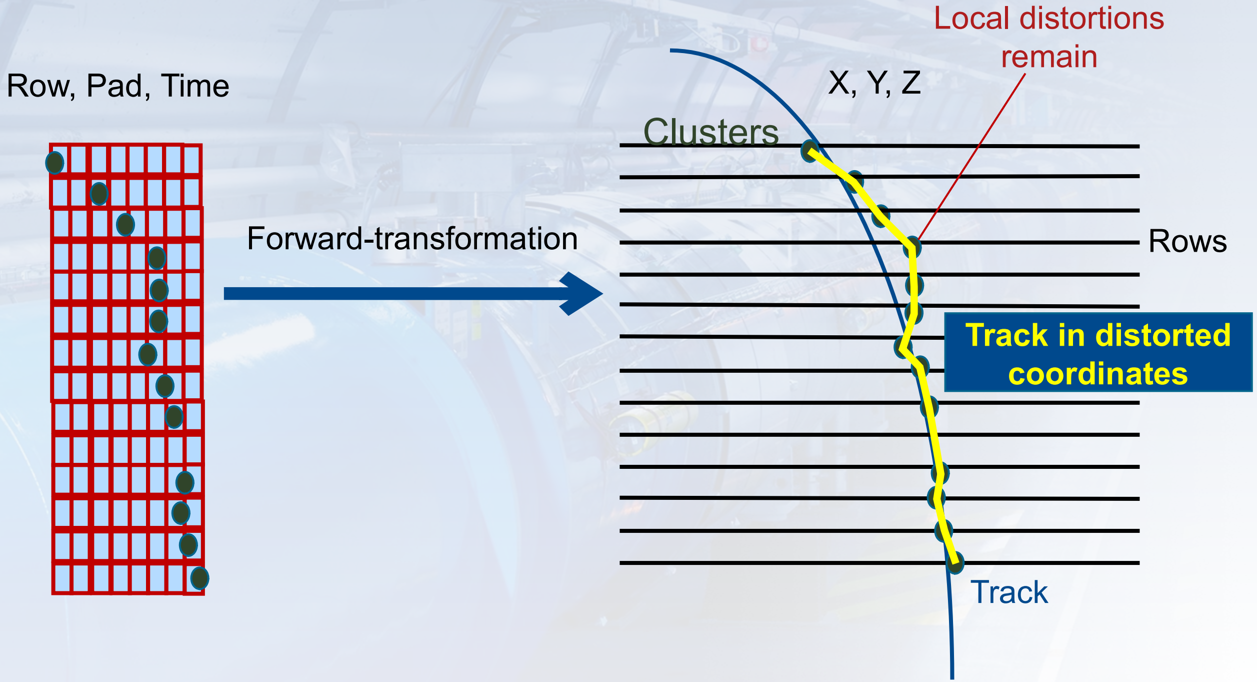
TPC Cluster Entropy Reduction: Track Model

- **Minimize residuals**. (**Smaller entropy** ☾ **Better Huffman compression**.)
- **Back to start!**



TPC Cluster Entropy Reduction: Track Model

- **Minimize residuals**. (**Smaller entropy** ☾ **Better Huffman compression**.)
  - Employ **fast, reversible linear approximation**. *(In principle, every transformation works.)*
  - **Refit track in distorted coordinate system**.

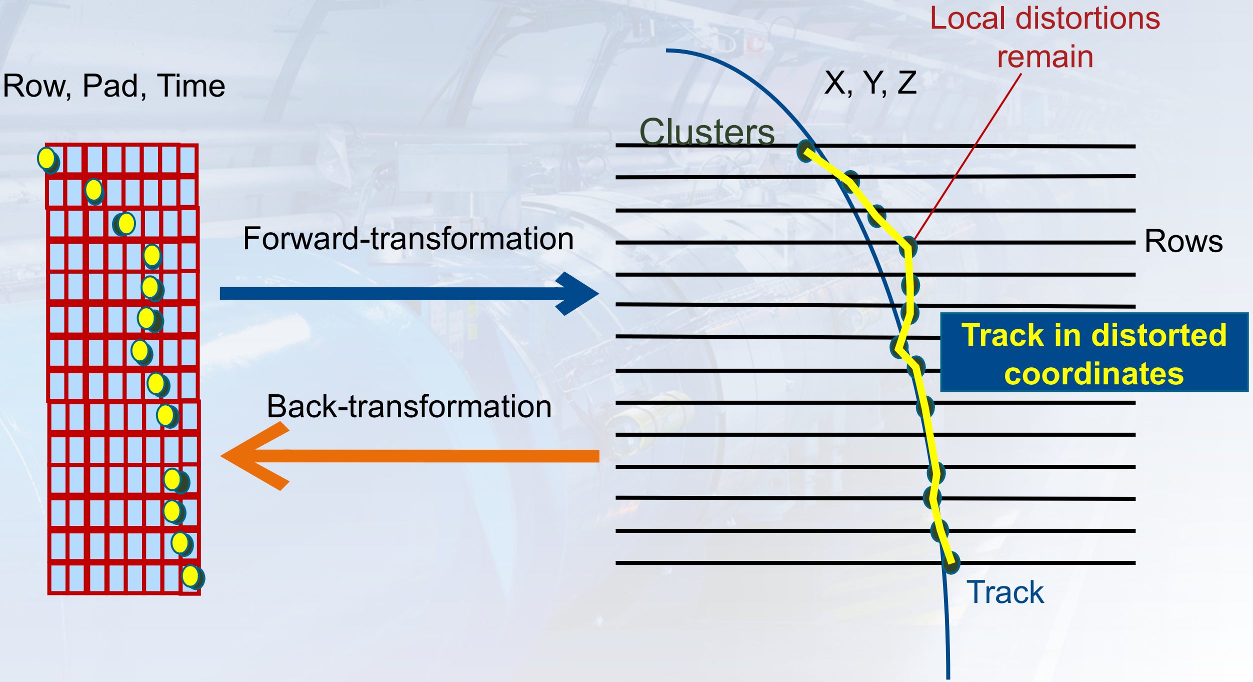


TPC Cluster Entropy Reduction: Track Model

- **Minimize residuals**. (**Smaller entropy** ☾ **Better Huffman compression**.)
  - Employ **fast, reversible linear approximation**. *(In principle, every transformation works.)*
  - **Refit track in distorted coordinate system**.

- **Store residuals in pad, time.**
  - Currently, storing initial q/p_T, sin(φ), and tan(λ) with low precision.
  - During decompression, perform the same refit with linear transformation.

- **Additional benefit:**
  - Cluster to track association is stored intrinsically.



TPC Cluster Entropy Reduction

1. Can use track information for other properties: e.g. clusters of a track should have similar charge.
   - Can store charge wrt. average track charge.
   - Better use truncated mean / median to compensate fake clusters, could consider track angle (basically dE/dx).

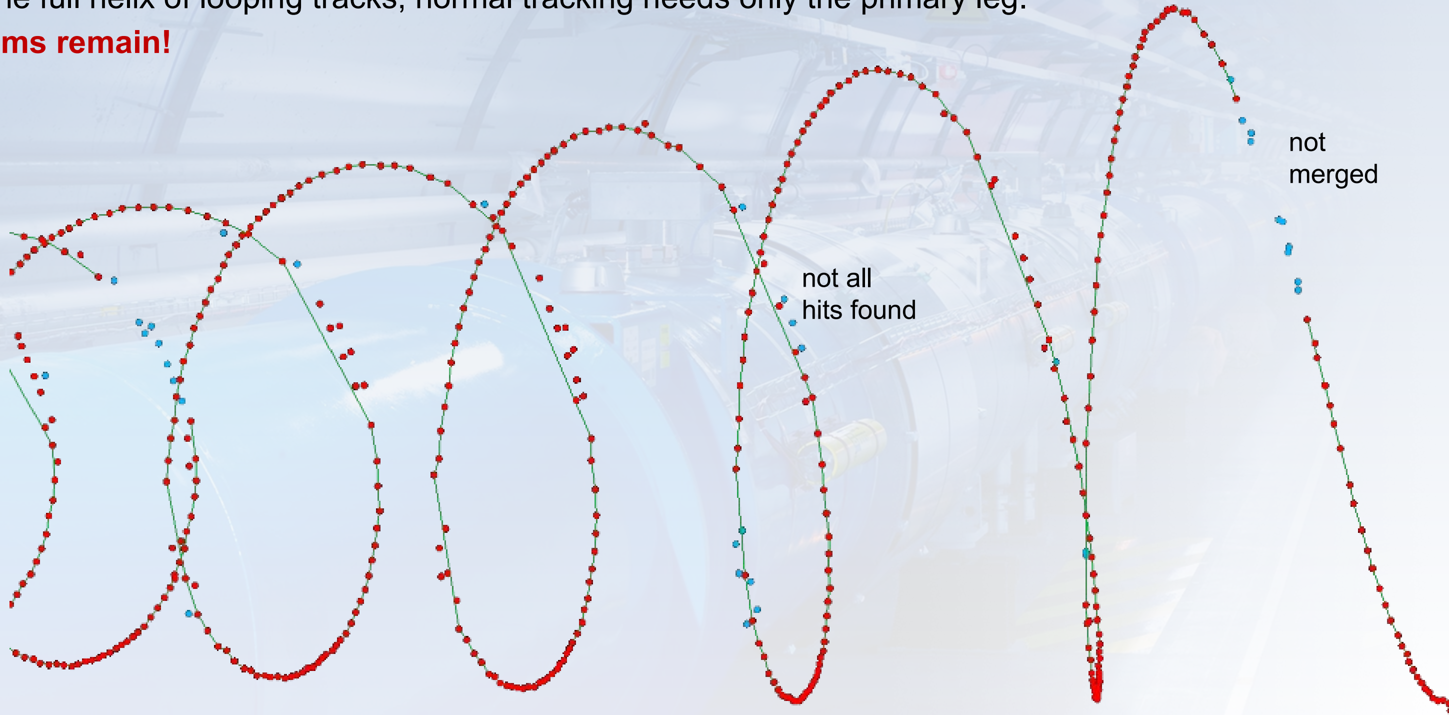
2. No need to store charge / width always with full precision.
   - Need only n significant bits (least significant bits irrelevant for large charge).
   - Basically we need a custom floating point format (no sign, custom size of exponent / mantissa).
   - Instead, we use our integer format and force all insignificant bits to 0 (with correct rounding).
     (insignificant = **n** bits after first non-zero bit: 00110111 ☾ 00111000 for **n** = 3.)
     - Many values are prohibited, entropy coding assigns optimal short representations for allowed values.

3. Unfortunately, the gains of these two strategies do not accumulate directly:
   - **Strategy 1** reduces the numbers in general (and introduces negative numbers), while **strategy 2** yields same-size representation for all values.
   - Might only be able to reduce the **n** of **strategy 2** further in combination.

4. Can do the same for cluster shape / size.

TPC Cluster rejection

- ALICE has implemented several improvements for low-p_T track finding.
- No impact on physics performance.
- Challenge: Needs to find the full helix of looping tracks, normal tracking needs only the primary leg.
- **Currently, several problems remain!**



ALICE Run 3 Data Rates

- **Data rates of ALICE detectors with large data contribution.**
- **All rates in GB/s during 50 kHz Pb-Pb (peak rates).**
- **For reference: Data rates assumed in TDR: 88 (66.5 – 105.2).**

- **TPC Biggest contributor to data rate.**

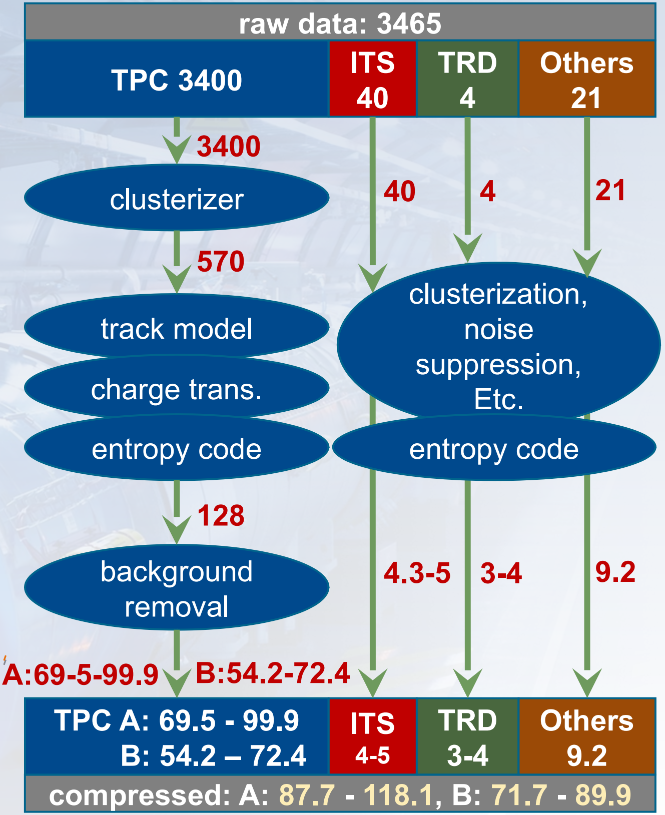
→ **TPC compression most critical.**

  - Assumed factor 20x in TDR.
    (Factors badly comparable, as raw format changed ☾ compare rates.)

TPC data rejection alternatives

A. Reject only clusters of identified background / tracks (loopers).
Rejects: 12.5% - 39.1%

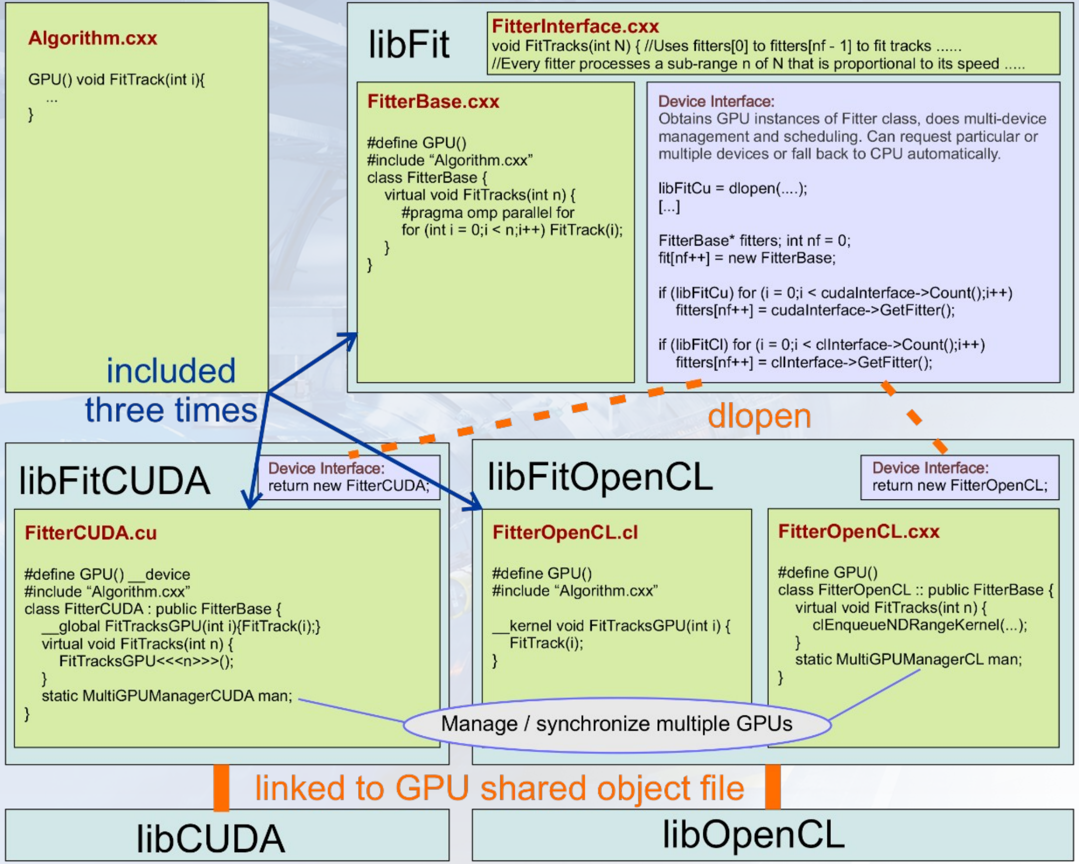
B. Keep only clusters attached or in proximity of identified signal tracks.
Rejects: 37.3% - 52.5%



Detailed layout of GPU software (track fit example)

- **Detailed example of the track fit code.**
  - Majority of the code is in Algorithm.cxx, which is shared between CPU and all GPU versions.
  - libFit can be loaded on all compute nodes (no dependency on CUDA / OpenCL).
    - The cuda and OpenCL tracking libraries (libFitCUDA and libFitOpenCL) can be loaded when the respective runtime (libCUDA or libOpenCL) is present.

- **Common source code for CPU / GPU.**
  - Supporting CUDA
  - HIP (AMD)
  - OpenCL (2.2 or clang 9)



Algorithm.cxx

```
GPU() void FitTrack(int i){
    ...
}
```

FitterInterface.cxx

```
void FitTracks(int N) { //Uses fitters[0] to fitters[nf - 1] to fit tracks ......
//Every fitter processes a sub-range n of N that is proportional to its speed .....
```

FitterBase.cxx

```
#define GPU()
#include “Algorithm.cxx”
class FitterBase {
   virtual void FitTracks(int n) {
      #pragma omp parallel for
      for (int i = 0;i < n;i++) FitTrack(i);
   }
}
```

Device Interface:
Obtains GPU instances of Fitter class, does multi-device management and scheduling. Can request particular or multiple devices or fall back to CPU automatically.

```
libFitCu = dlopen(....);
[...]

FitterBase* fitters; int nf = 0;
fit[nf++] = new FitterBase;

if (libFitCu) for (i = 0;i < cudaInterface->Count();i++)
   fitters[nf++] = cudaInterface->GetFitter();

if (libFitCl) for (i = 0;i < clInterface->Count();i++)
   fitters[nf++] = clInterface->GetFitter();
```

Device Interface:

```
return new FitterCUDA;
```

FitterCUDA.cu

```
#define GPU() __device
#include “Algorithm.cxx”
class FitterCUDA : public FitterBase {
   __global FitTracksGPU(int i){FitTrack(i);}
   virtual void FitTracks(int n) {
      FitTracksGPU<<<n>>>();
   }
   static MultiGPUManagerCUDA man;
}
```

Device Interface:

```
return new FitterOpenCL;
```

FitterOpenCL.cl

```
#define GPU()
#include “Algorithm.cxx”

__kernel void FitTracksGPU(int i) {
   FitTrack(i);
}
```

FitterOpenCL.cxx

```
#define GPU()
class FitterOpenCL :: public FitterBase {
   virtual void FitTracks(int n) {
      clEnqueueNDRangeKernel(...);
   }
   static MultiGPUManagerCL man;
}
```

Summary

- ALICE will take **50 kHz of minimum bias Pb-Pb** data in Run 3.
- There will be **no triggers** but data is compressed online in software.
- High data compression factors require online reconstruction, in particular TPC tracking.
- Full tracking for the TPC in the synchronous phase, tracking for ITS and TRD for few percent of the events.
- Full reconstruction with final calibration in the asynchronous phase.
- The majority of the synchronous phase will run on GPUs, asynchronous phase can run on GPU but also in the GRID.
- In an optimistic scenario, we can offload almost full barrel tracking to GPU.
- TPC Reconstruction more challenging than today due to space charge distortions.
- TPC Data compression still big issue:
  - Entropy compression factor of **9.1x** (10% short wrt. TDR).
  - **Cluster rejection turns out to be difficult.**
    - Random high-p_T tracks fake-protect junk clusters.
    - Incomplete track merging reduces the number of looping legs to be removed.
    - Still significant fraction of unattached clusters.
    - 13% of clusters not accessible by tracking (very low p_T, charge cloud from low p_T protons, noisy pads).
  - **Strategy B** could increase rejection ratio at the risk of loosing physics tracks in case of issue with calibration.
- Total data rate still in agreement with the TDR since we can save at other places.
- GPU code implemented in shared code also for CPU, algorithm speed-up ca **20x**, 1 GPU replaces ca **40 CPU cores**.