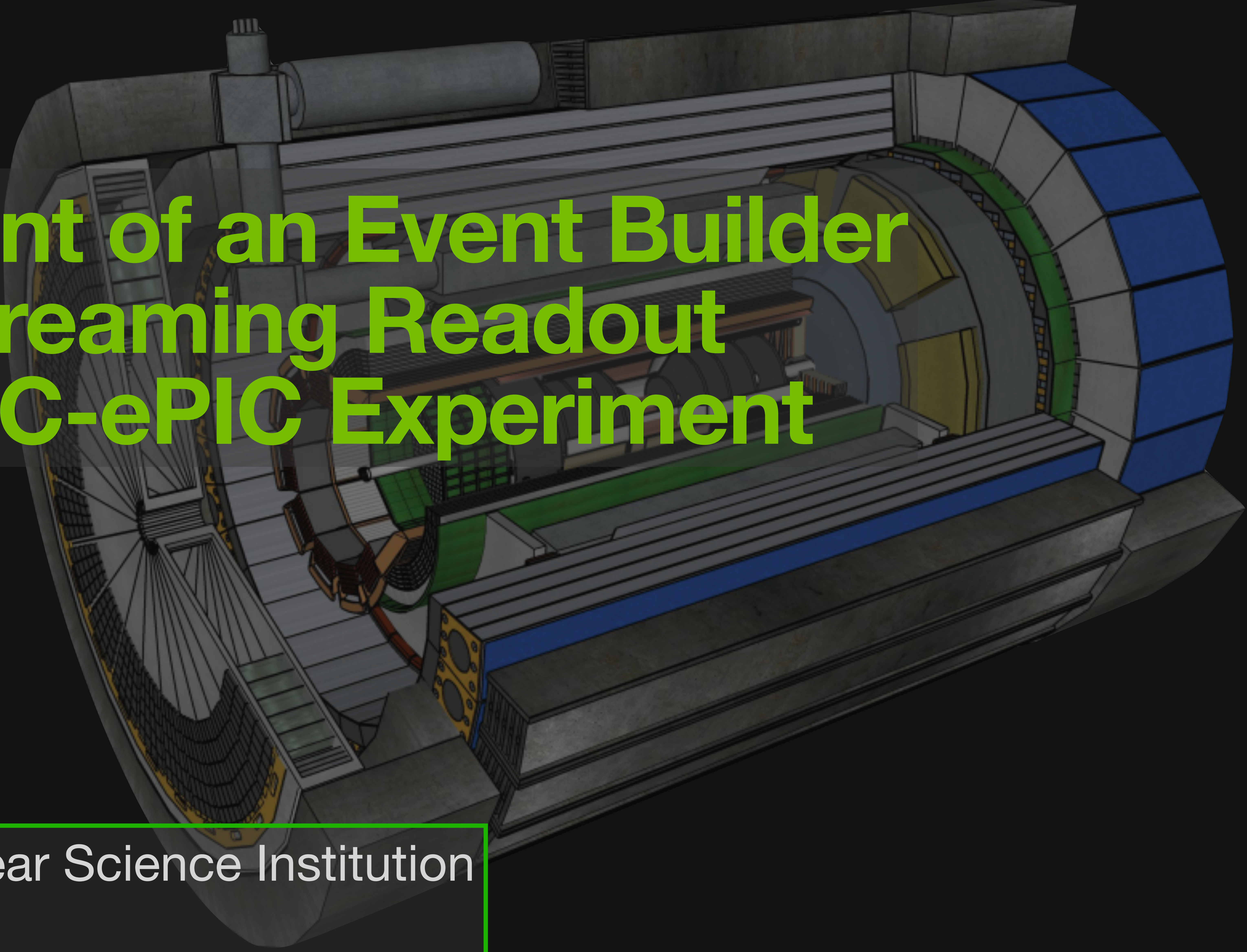


Development of an Event Builder with Streaming Readout for the EIC-ePIC Experiment

The University of Tokyo, Quark Nuclear Science Institution
Takuya Kumaoka

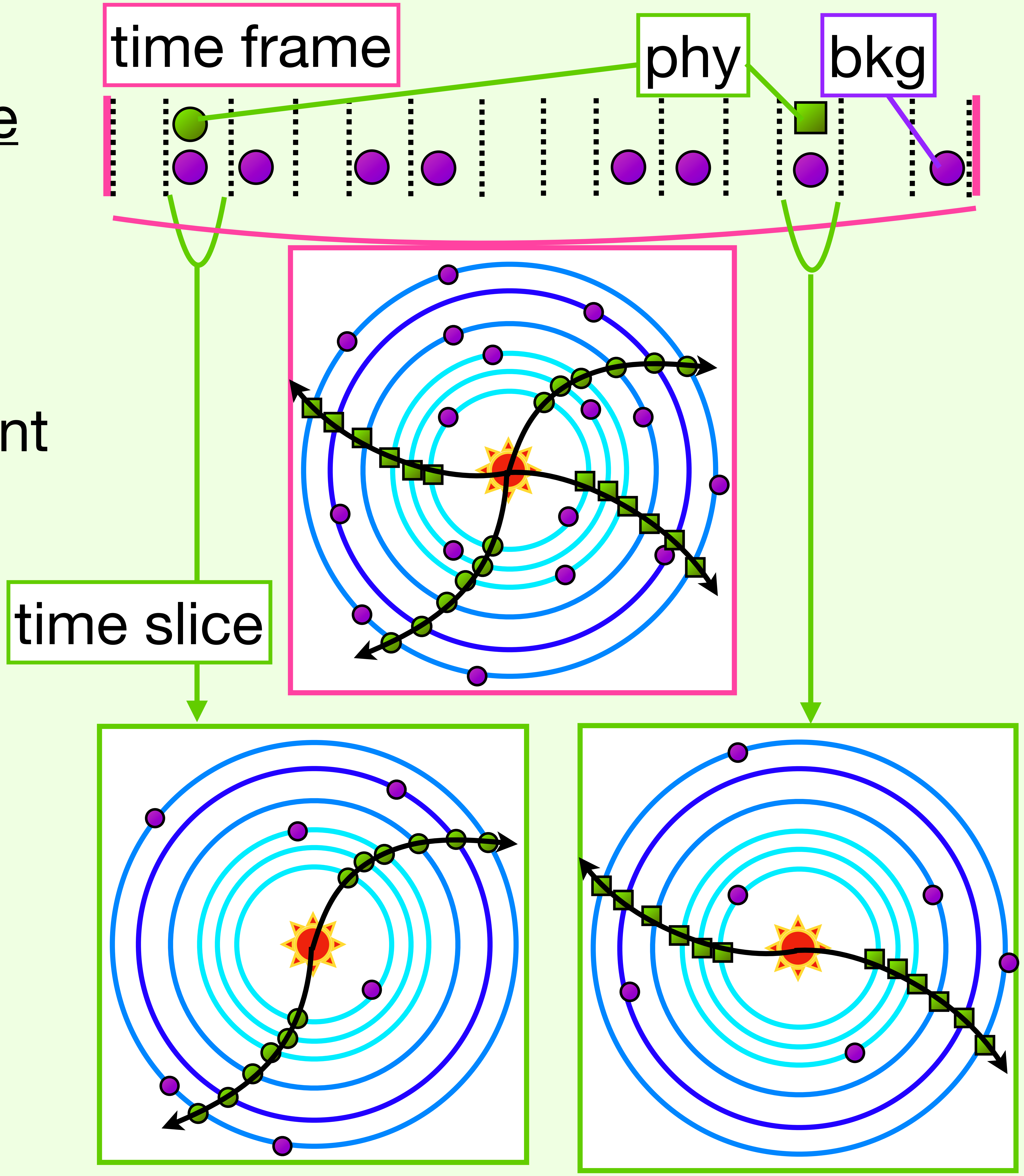
Aim of My Work

My study aim

Develop event extruction&filter/identification module on the streaming using EICrecon.

Requirements

- Zero suppression physics event extraction
- Compress the data to send and storage every event
- Simple and safety to run on the streaming



Current Architecture of My Online Event Filtering System

Map of functions to be developed

Time frame
Timeframe Lv
Source
Process
Unfold
Physics event
Background/ Noise
Event Lv
Process
Write

1. Time alignment for each sub-detector

- Digitization

2. **Time splitting**

- Clustering
- ACTS (track and vertex reconstruction)

3. Terminate processes for time-slices with empty or few hits.

Event identification

4. Terminate processes for time-slices with identified noise events (Using reconstructed tracks).

Progress

1. Create a new factory to time alignment (**Done**)

2. Unfolding trivial test (**Done**)

3. Time splitting test (**Done**)

4. Timing Coincidence (**Done**)

5. Injection of background events and detector noises and evaluation of timing coincidence (time windows vs. rejection, efficiency, primary tracking, etc) (**On going: Mid of August**)

This Presentation Topic

1. Introduction new time splitting algorithm (ver.3)

2. Event filter efficiency

3. Process time

4. Track and Vertex Parameters

5. Track's Purity and Efficiency

New event build algorithm Ver3.0

1. Using the quickest detector (TOF) slices the time-frame.

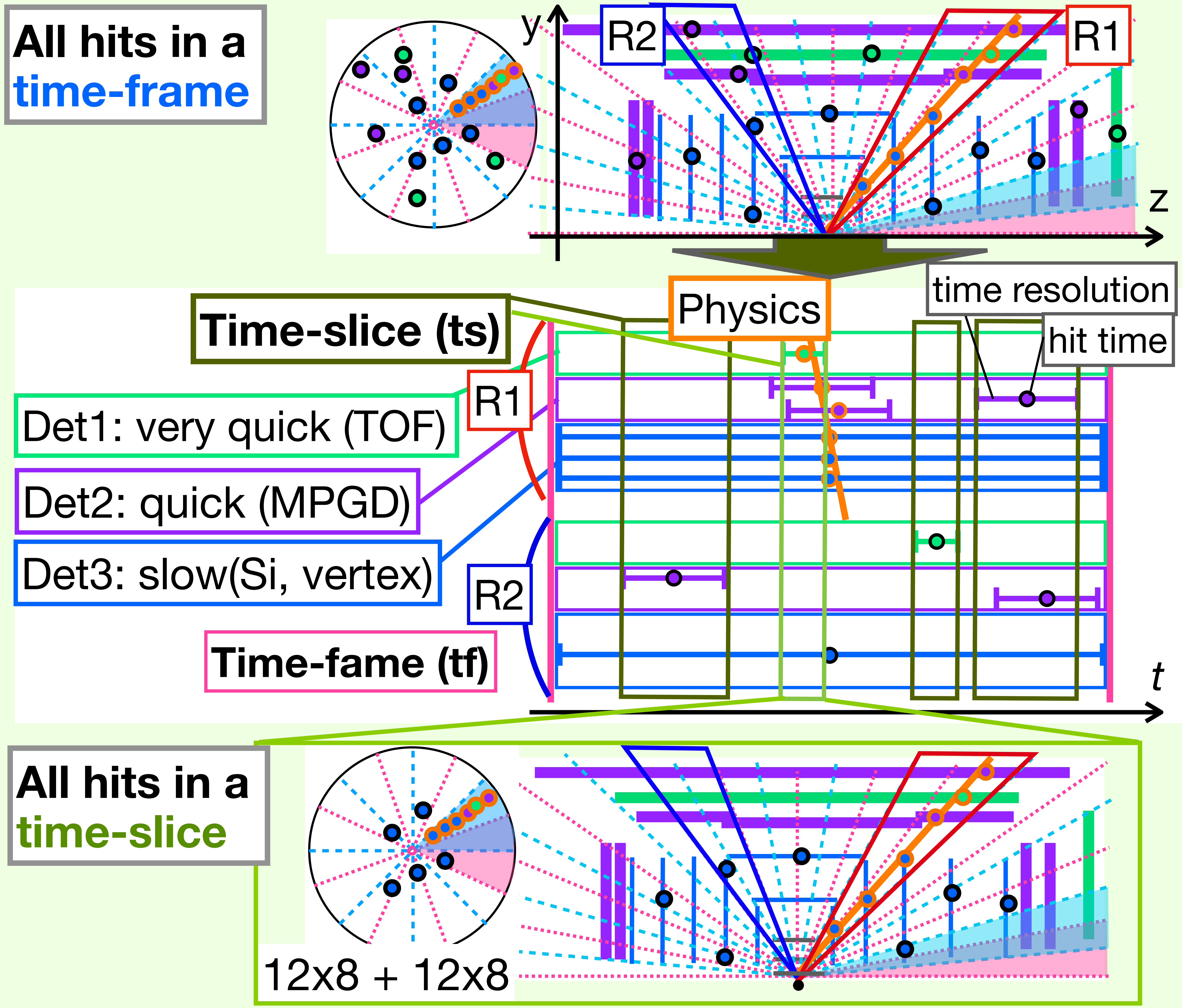
2. Check the topology of the hits.

3. If there are more than three hits in the same region, the time slice is triggered.

4. Repeat steps 1–3 for all TOF hits.

5. If any unused MPGD hits remain, perform the same process using them.

6. If there is no trigger using TOF and MPGD in the time frame, perform the same process based on the Silicon detectors’ hits.



All hits in a time-frame

Time-slice (ts)

Physics

time resolution

hit time

Det1: very quick (TOF)

Det2: quick (MPGD)

Det3: slow(Si, vertex)

Time-fame (tf)

All hits in a time-slice

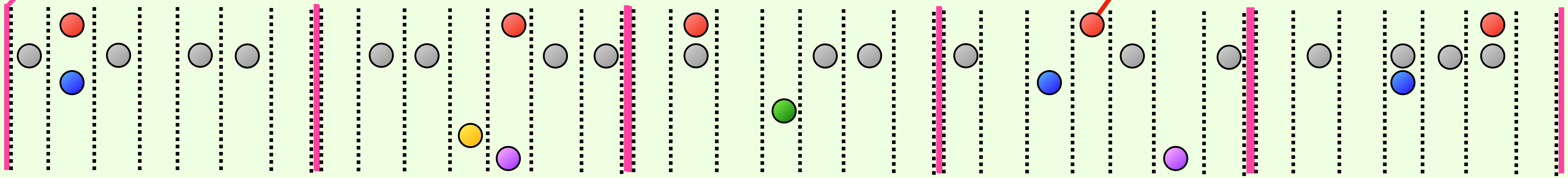
12x8 + 12x8

Simulation Setup

Physics simulation (1000 events)

time-frame (2 μs: 500 kHz)

physics events(500 kHz)

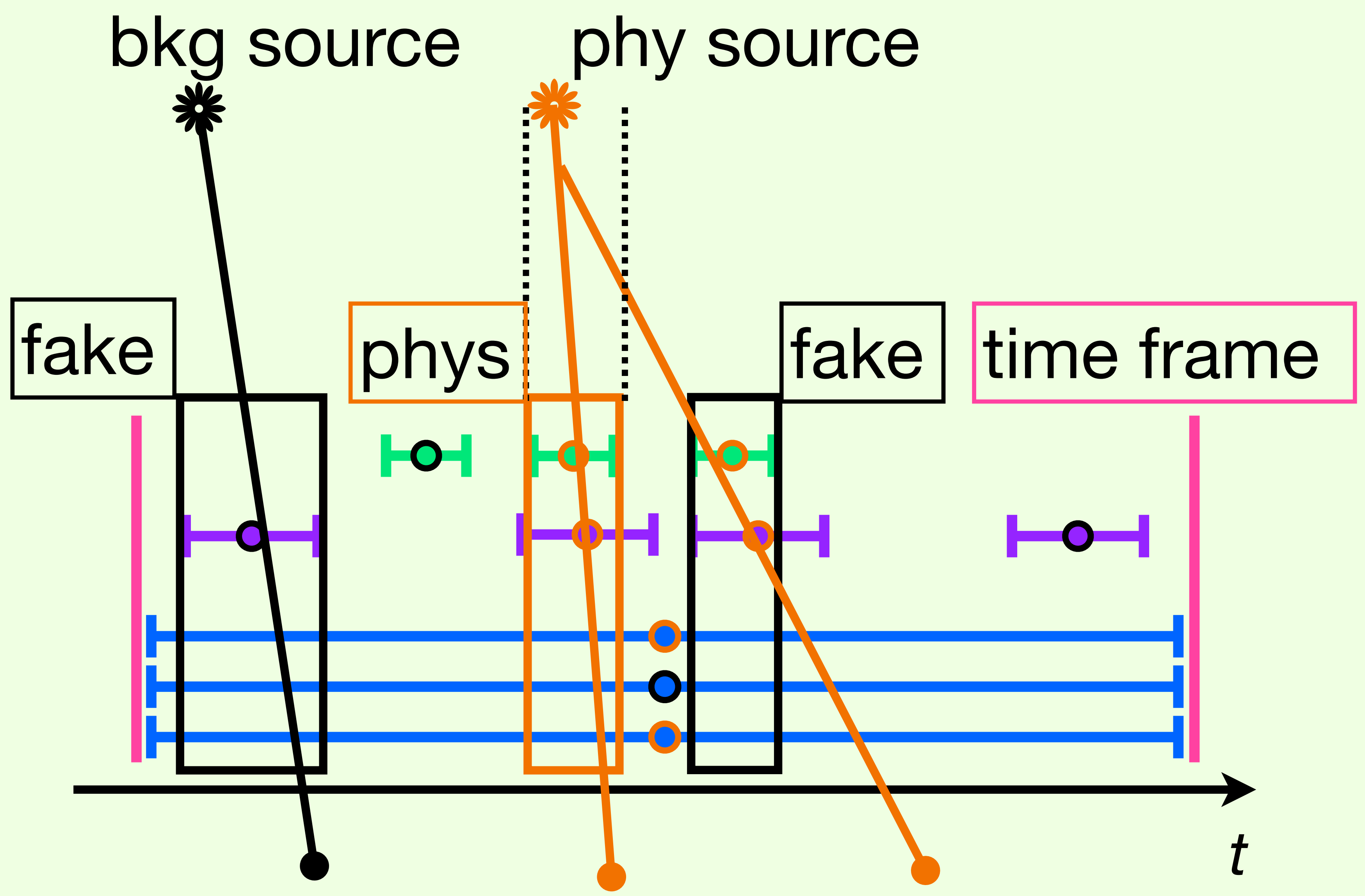


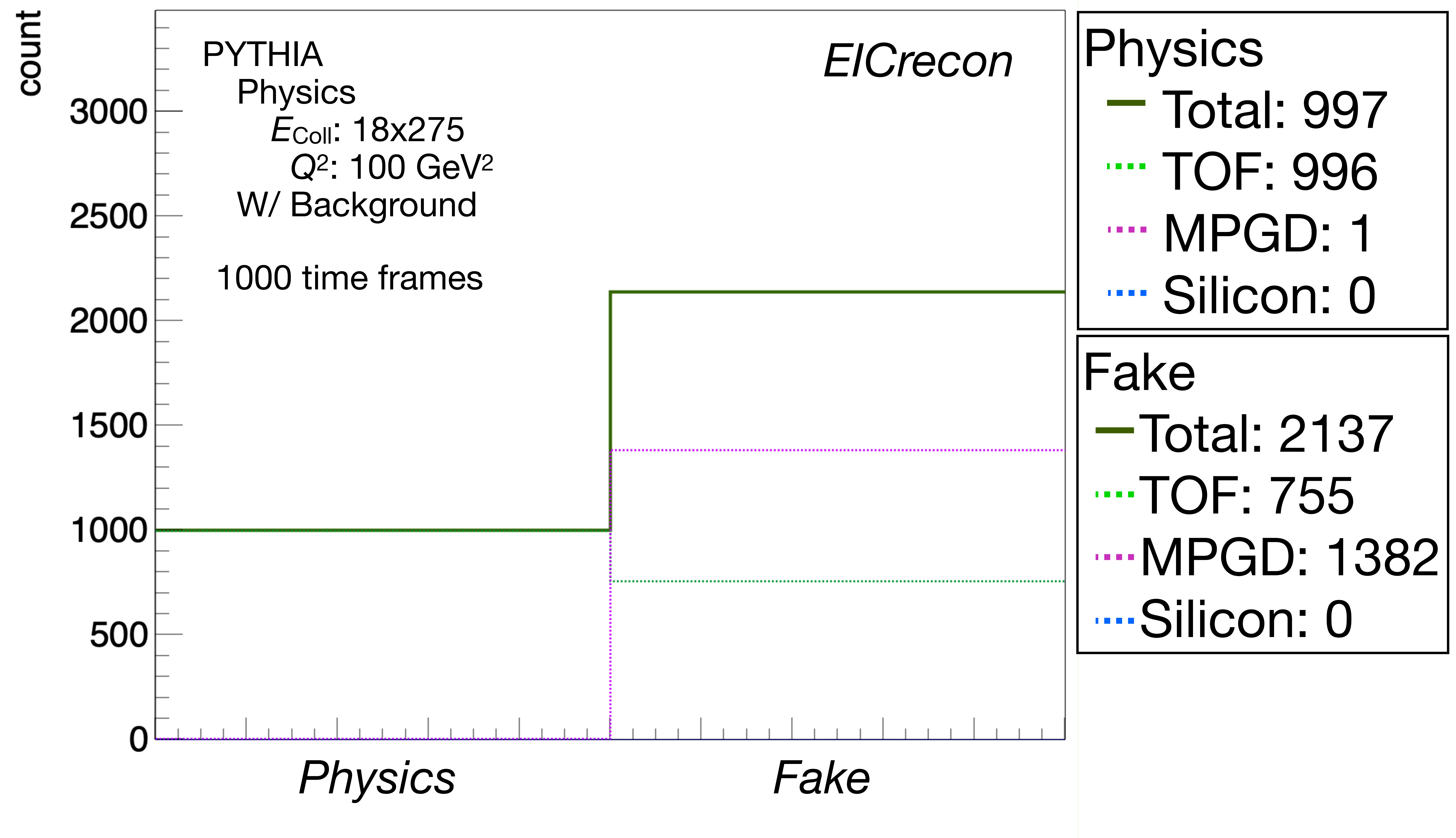
🔴	**DIS NC 18x275 Q^2 > 1 or 100** **(Deep inelastic scattering neutral current)**	500 kHz
⚪	**Synchrontron Radiation**	3.3 GHz
🔵	**Electron bremsstrahlung radiation**	317 kHz
🟢	**Electron Touscheck scattering (intrabeam dcattering)**	1.3 kHz
🟡	**Electron Coulomb scattering processes**	0.72 kHz
🟣	**Proton beam gas interactions**	22.5 kHz

First Event Filter Efficiency

Time slice containing the production time of a physics
→ ✓ **physics trigger**
→ ✗ **fake triggers**

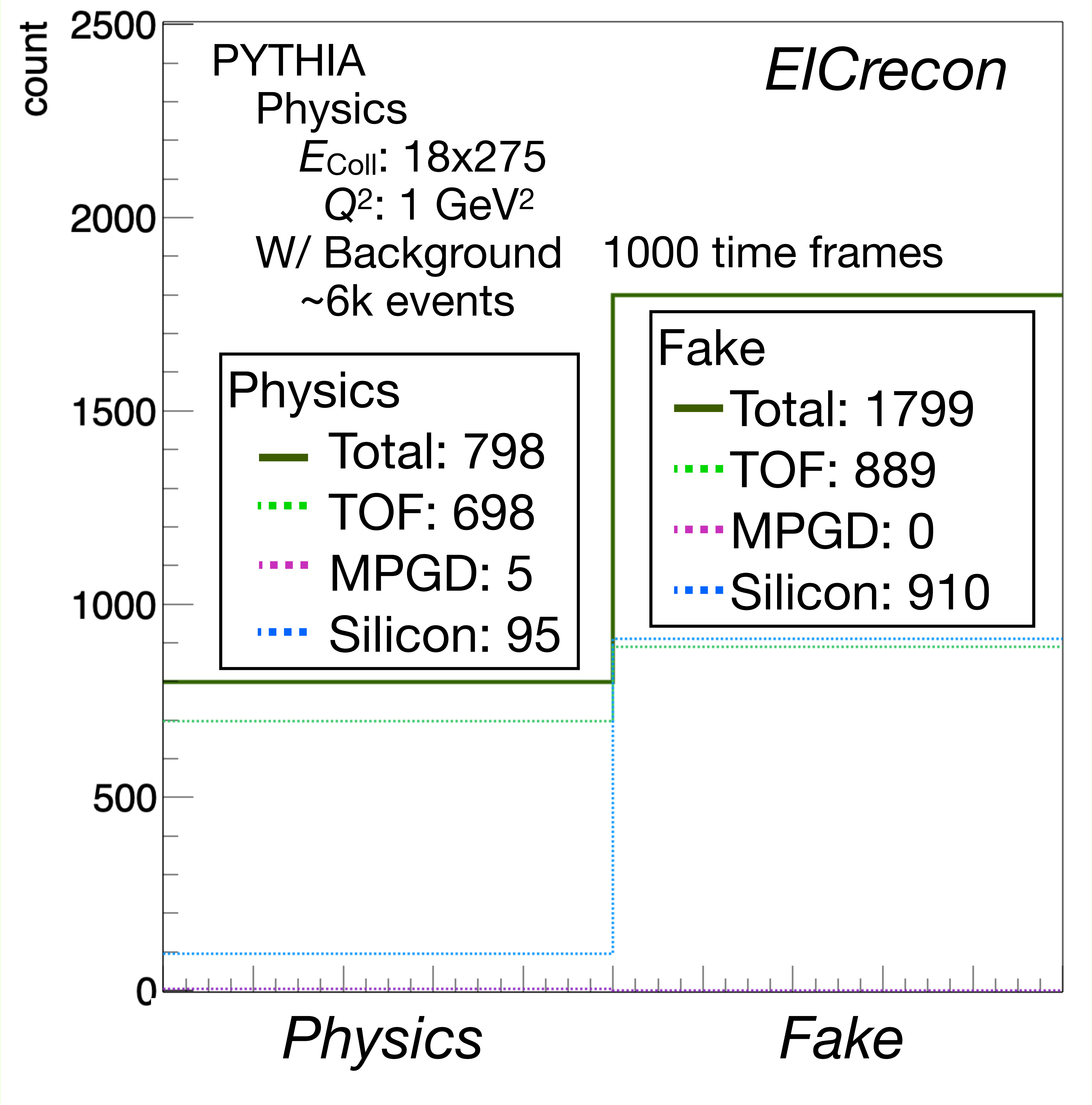
※ Even if a slice contains many hits from a physics event, it is still classified as fake if it does not include the production time.

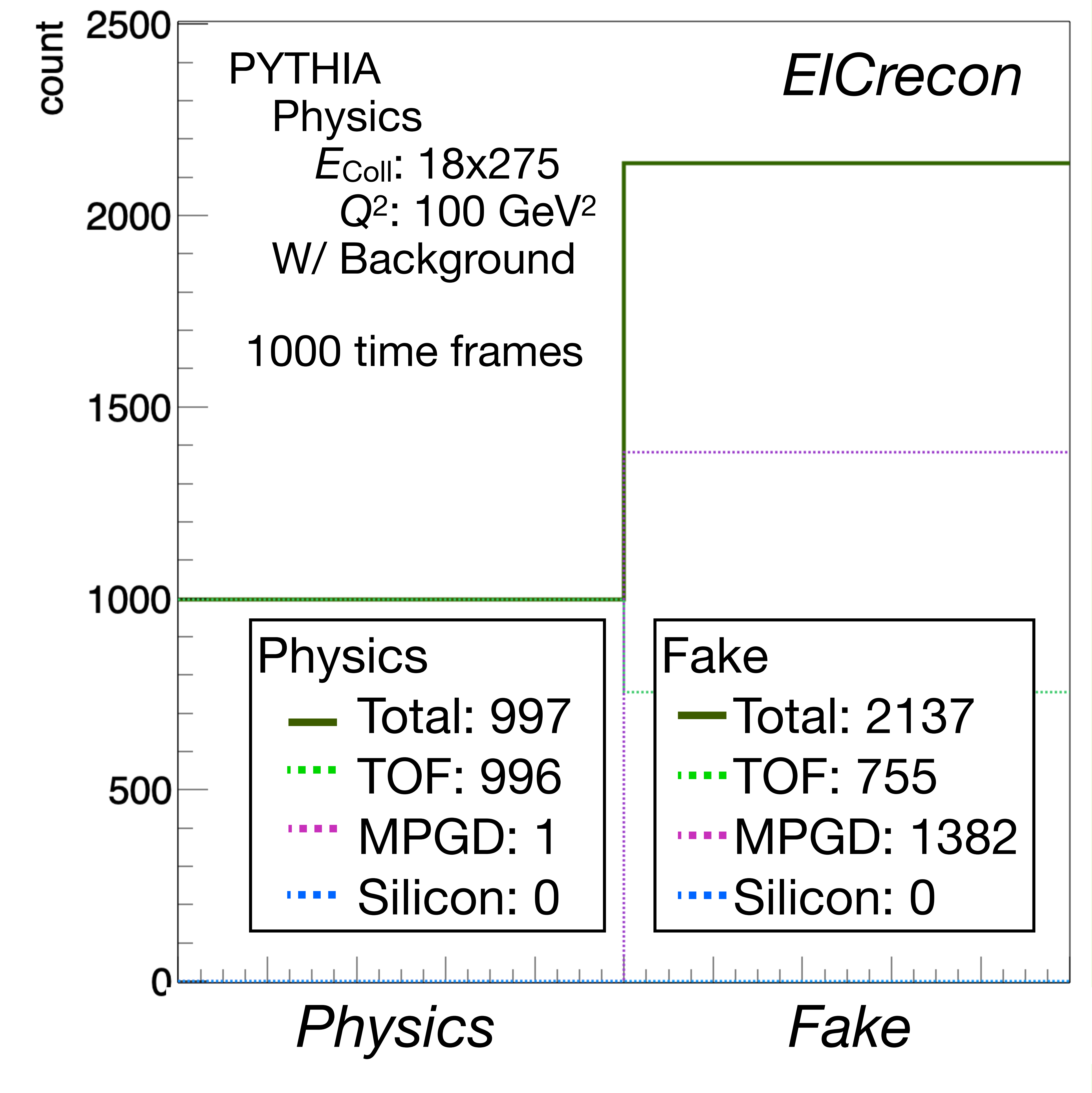




For Q^2 = 100 GeV^2 events, the first event filter efficiency is over **99%**.
→ Need to check what makes these fake triggers.

First Event Filter Efficiency



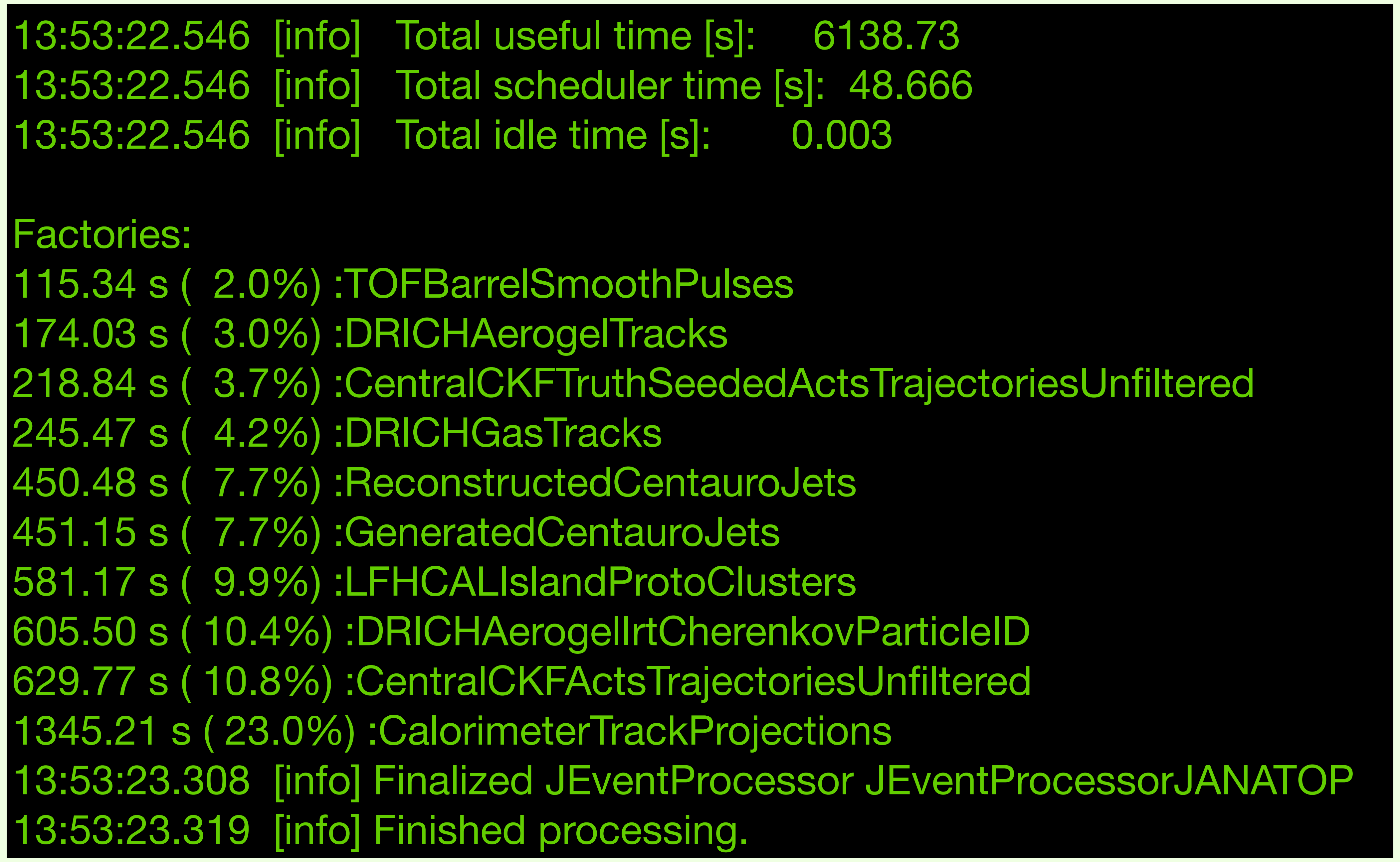


For Q^2 = 1 GeV^2 events, the first event filter efficiency is about 80%.

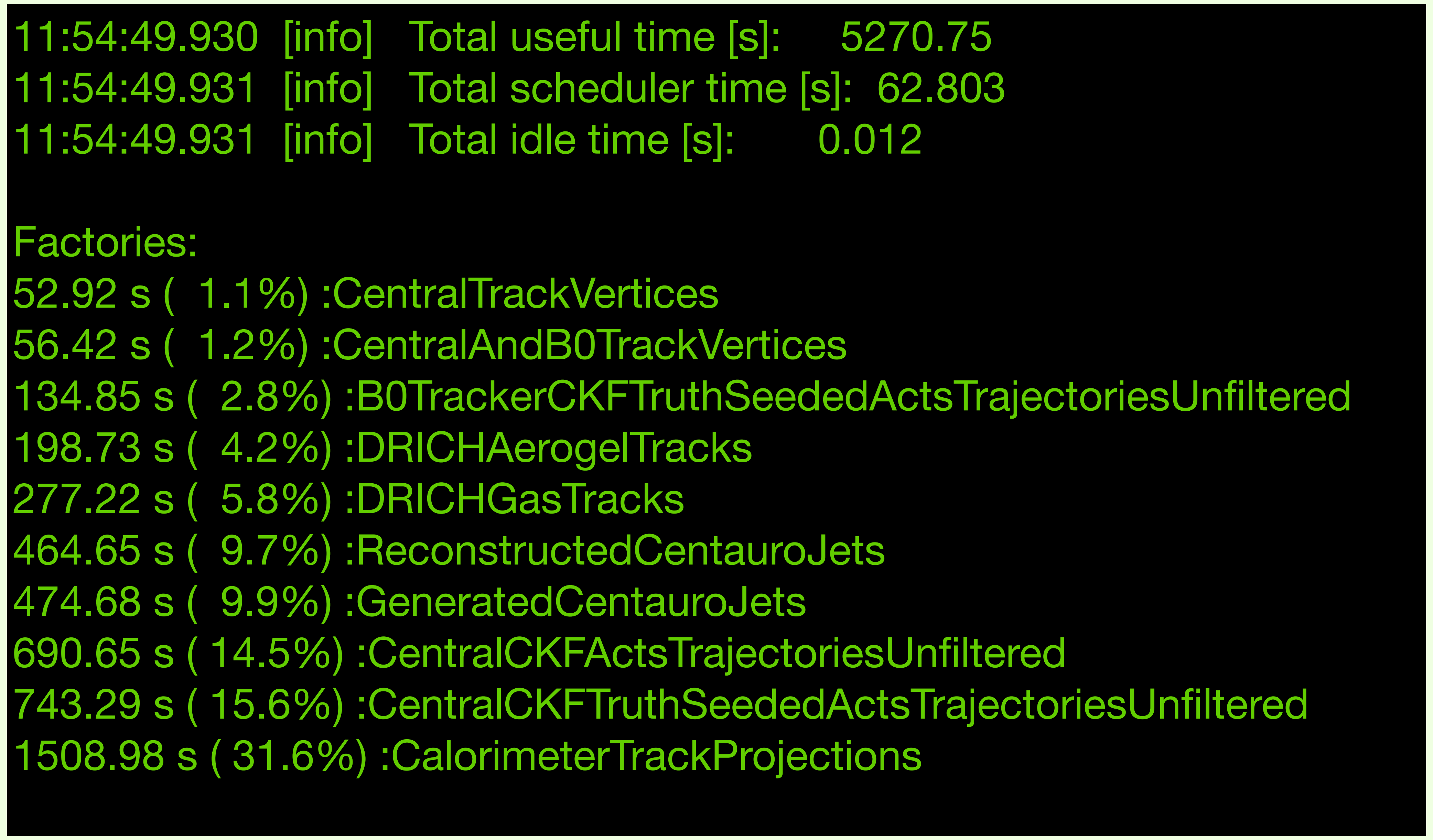
→ This quality is not enough. → What type of events are missed? → Lower than Q^2 = 1 GeV^2?

Total Process Time W/O and W/ TimeSplitter

W/O TimeSplitter: 1.7 h / 1000 time-frame ➡ W TimeSplitter: 1.5 h / 1000 time-frame



```
13:53:22.546  [info]  Total useful time [s]:     6138.73
13:53:22.546  [info]  Total scheduler time [s]:  48.666
13:53:22.546  [info]  Total idle time [s]:       0.003

Factories:
115.34 s (  2.0%) :TOFBarrelSmoothPulses
174.03 s (  3.0%) :DRICHAerogelTracks
218.84 s (  3.7%) :CentralCKFTruthSeededActsTrajectoriesUnfiltered
245.47 s (  4.2%) :DRICHGasTracks
450.48 s (  7.7%) :ReconstructedCentauroJets
451.15 s (  7.7%) :GeneratedCentauroJets
581.17 s (  9.9%) :LFHCALIslandProtoClusters
605.50 s ( 10.4%) :DRICHAerogelIrtCherenkovParticleID
629.77 s ( 10.8%) :CentralCKFActsTrajectoriesUnfiltered
1345.21 s ( 23.0%) :CalorimeterTrackProjections
13:53:23.308  [info] Finalized JEventProcessor JEventProcessorJANATOP
13:53:23.319  [info] Finished processing.
```



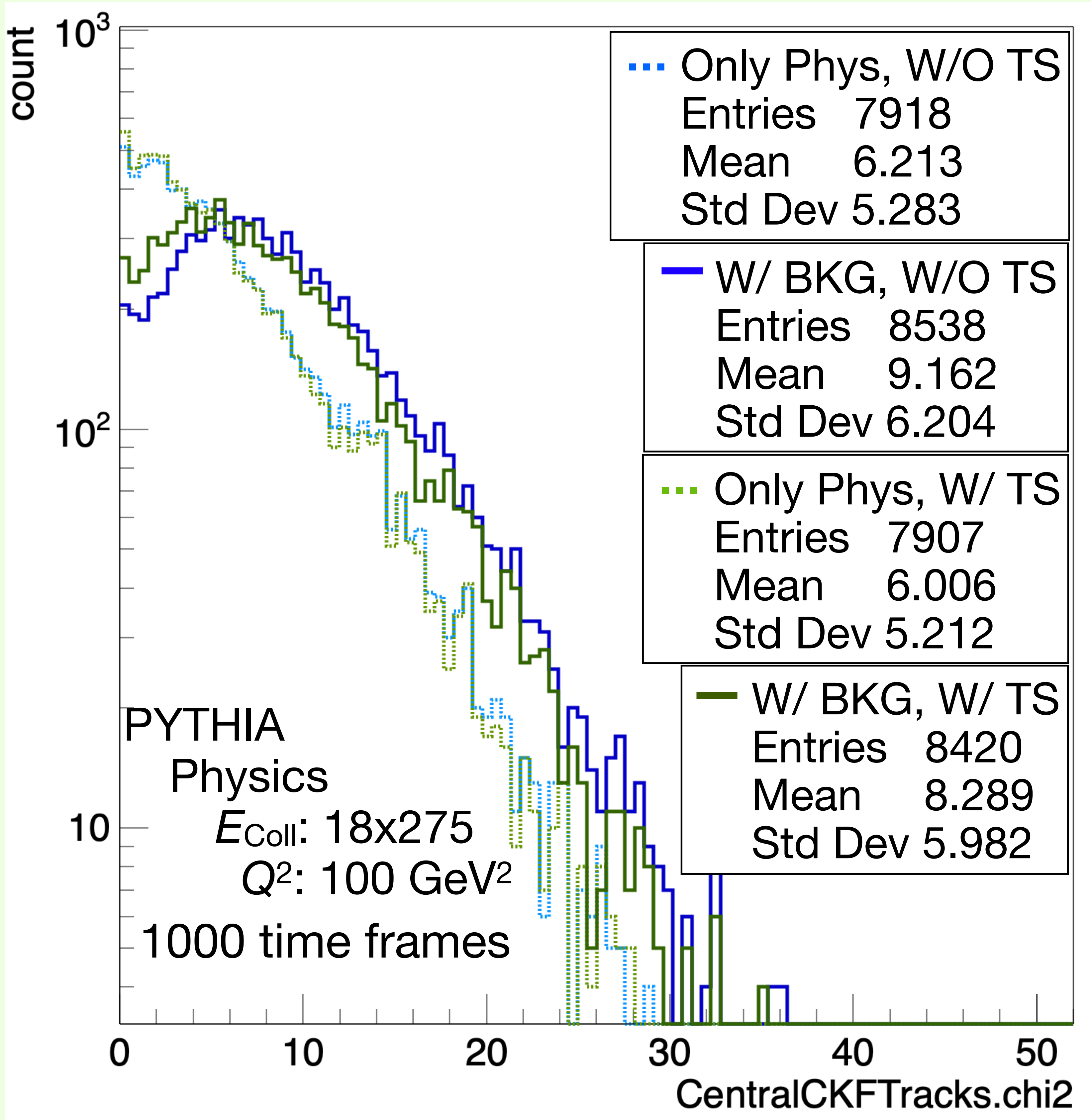
```
11:54:49.930  [info]  Total useful time [s]:     5270.75
11:54:49.931  [info]  Total scheduler time [s]:  62.803
11:54:49.931  [info]  Total idle time [s]:       0.012

Factories:
52.92 s (  1.1%) :CentralTrackVertices
56.42 s (  1.2%) :CentralAndB0TrackVertices
134.85 s (  2.8%) :B0TrackerCKFTruthSeededActsTrajectoriesUnfiltered
198.73 s (  4.2%) :DRICHAerogelTracks
277.22 s (  5.8%) :DRICHGasTracks
464.65 s (  9.7%) :ReconstructedCentauroJets
474.68 s (  9.9%) :GeneratedCentauroJets
690.65 s ( 14.5%) :CentralCKFActsTrajectoriesUnfiltered
743.29 s ( 15.6%) :CentralCKFTruthSeededActsTrajectoriesUnfiltered
1508.98 s ( 31.6%) :CalorimeterTrackProjections
```

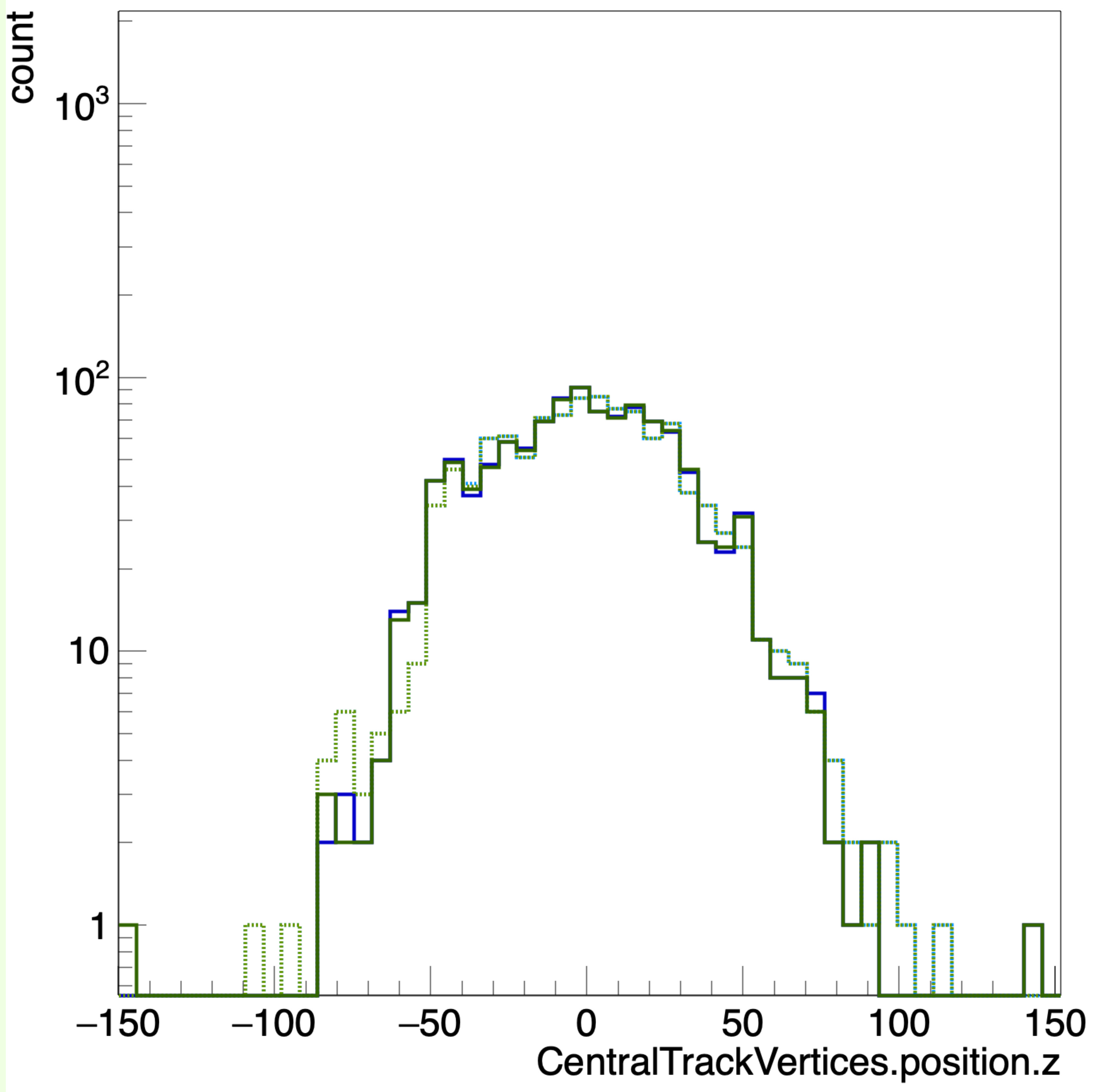
The processing time with the time splitter is **faster** than without it.
→ This is expected because the splitter reduces background hits and helps the tracking.

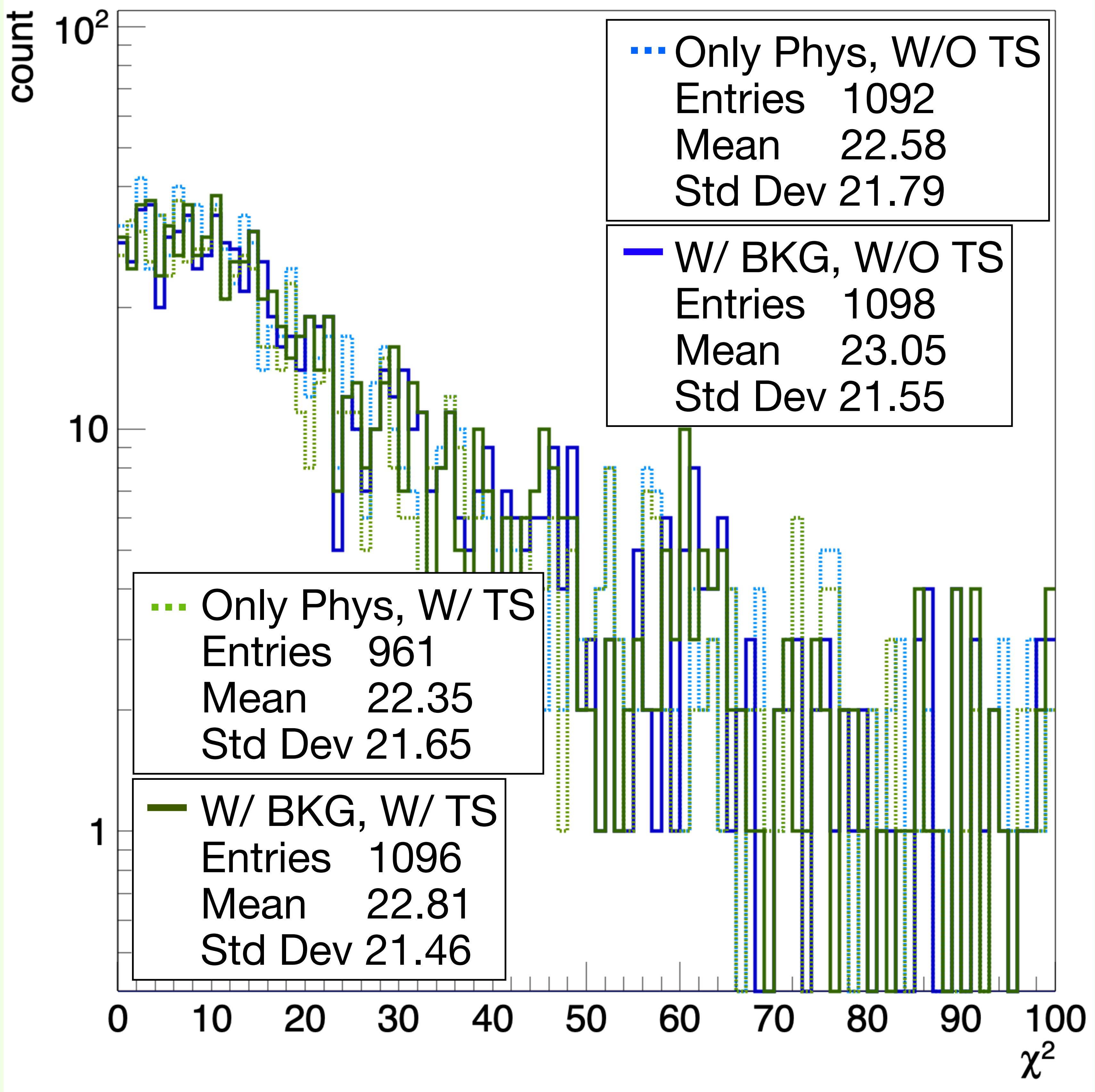
Single event of large number of hits:: W/O TS: 468 s → W/ TS: 242 s
→Particularly, this splitting works well for large number of hits.

Reconstructed Tracks’ and Vertices’ Parameter Comparison

ACTS’s tracking and vertex output comparison



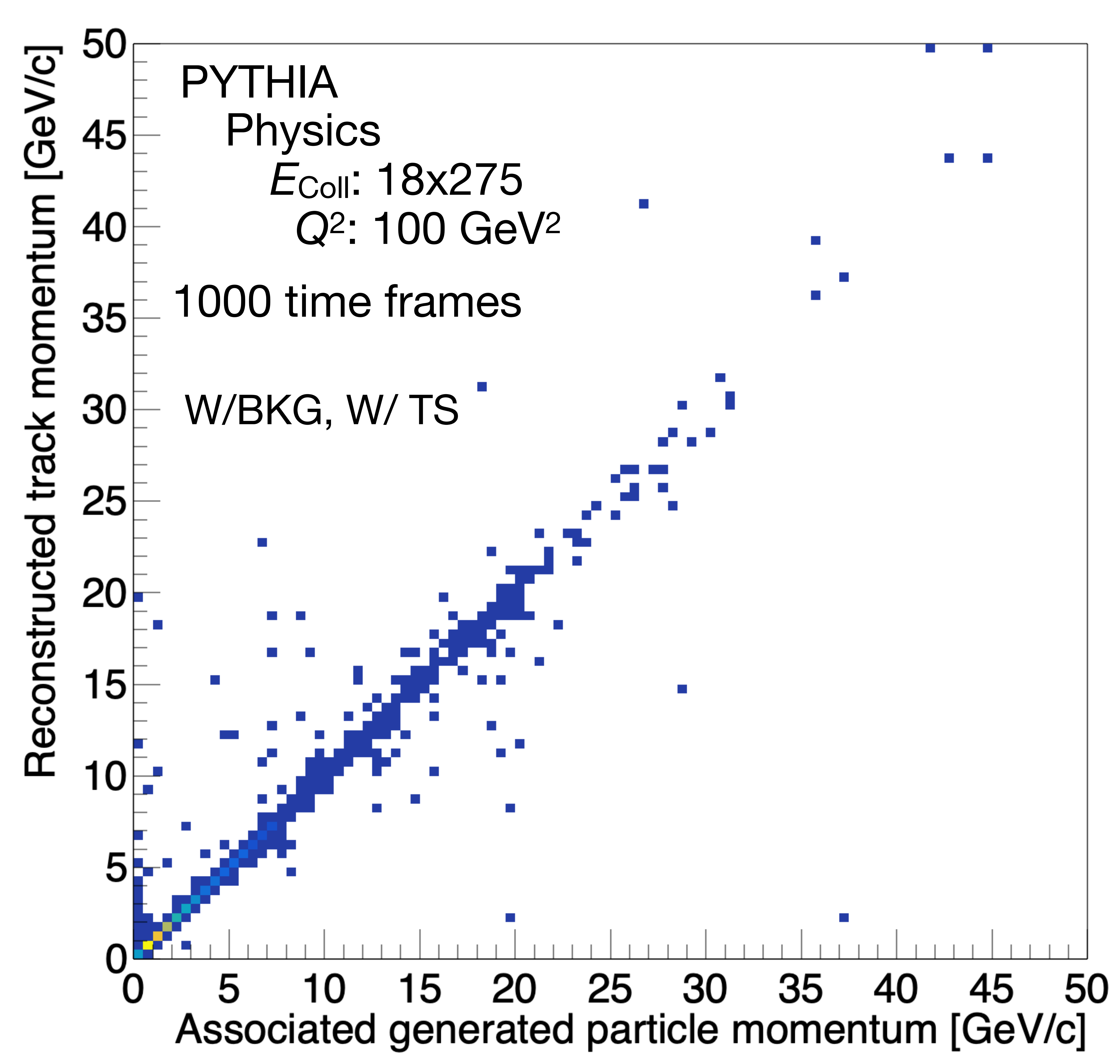


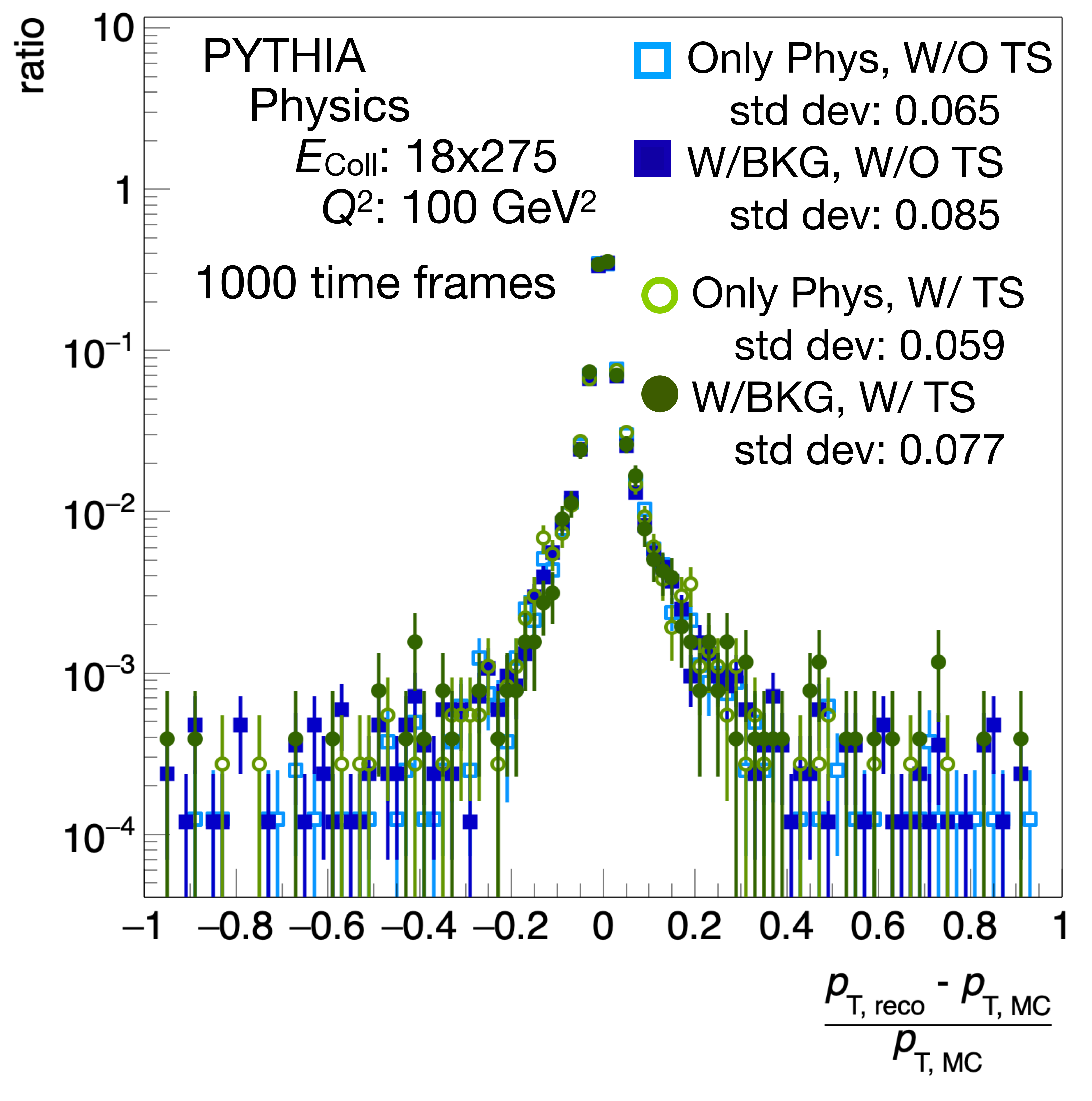


Overall, the results with time splitting are consistent with those without it.

For both the tracking χ^2 and the vertex χ^2, the time-split results show **better** performance than the no-splitting case.

Track p_T Resolution

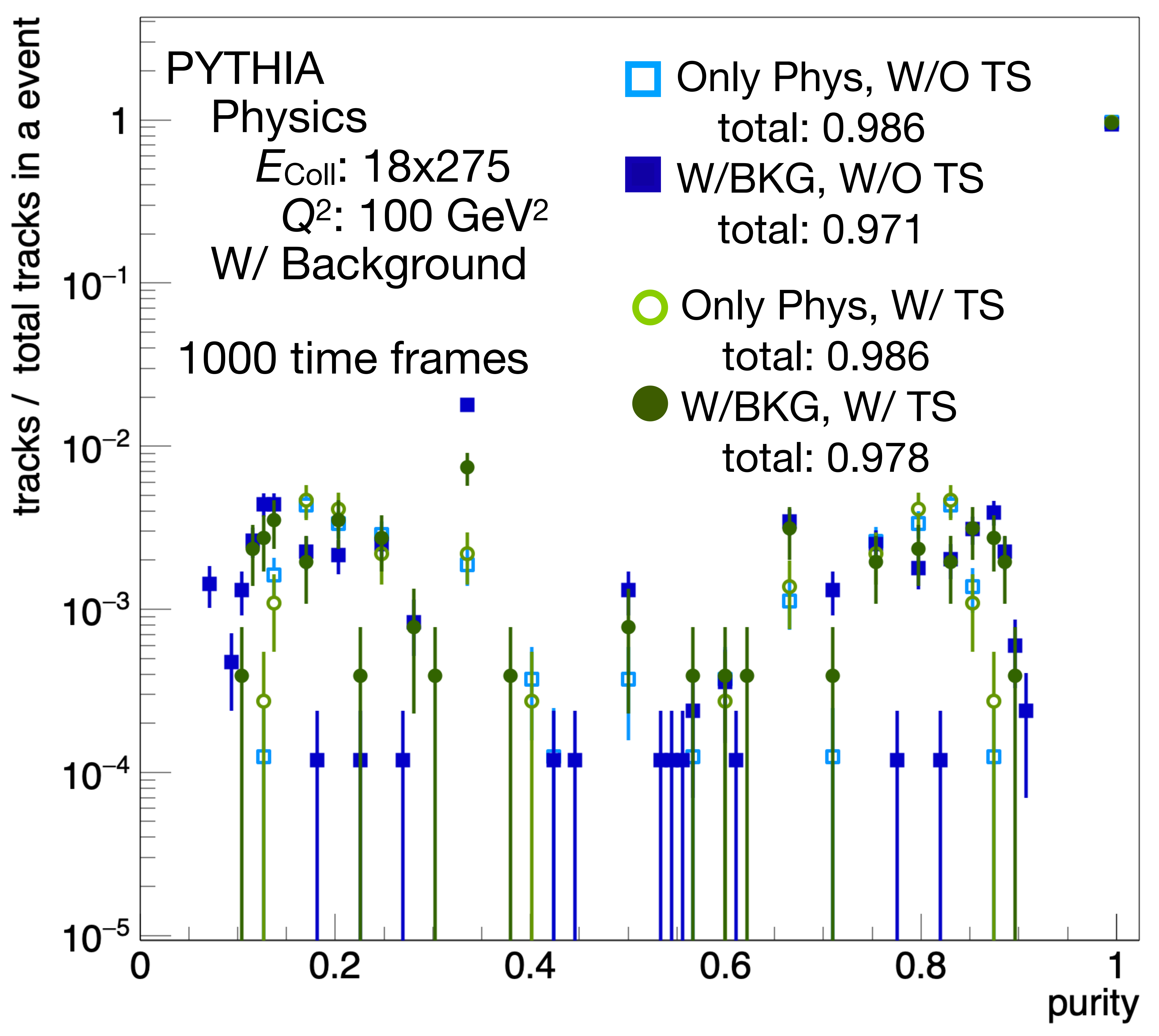


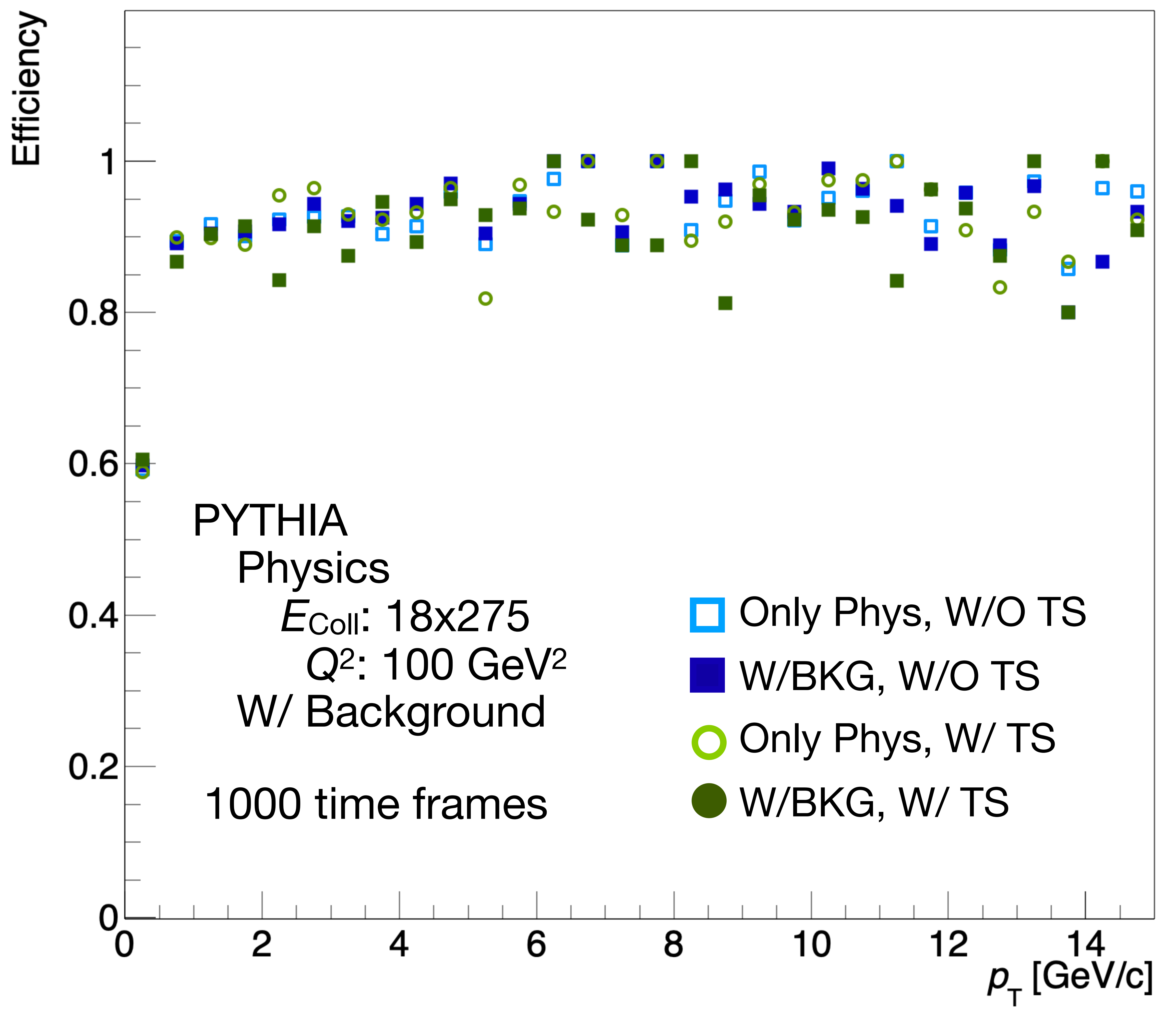


Good linearity between MC and reconstructed-track p_T.

The p_T resolution of the time splitting is a little better than W/O time splitting.

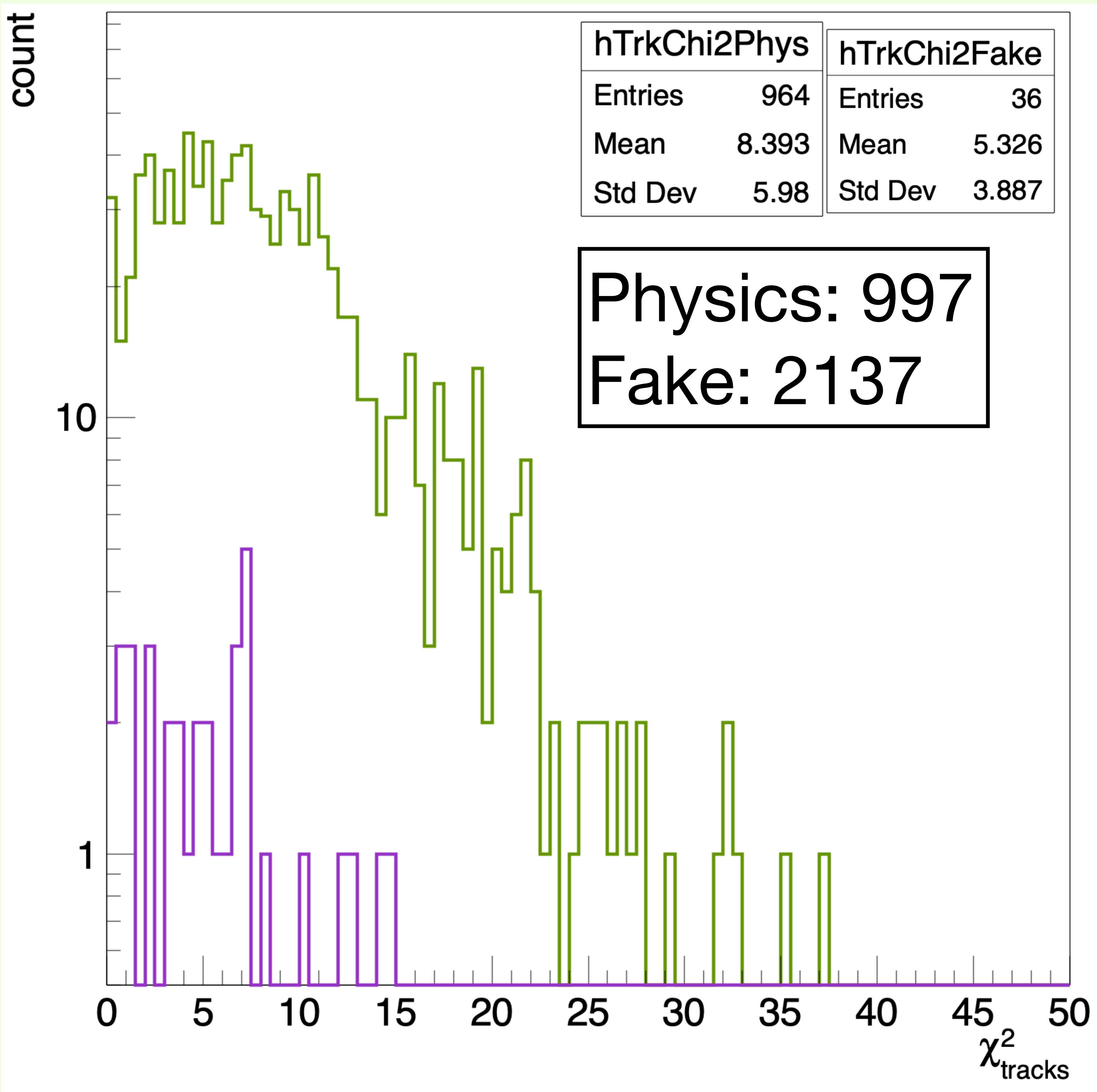
Track Purity and Efficiency Evaluation



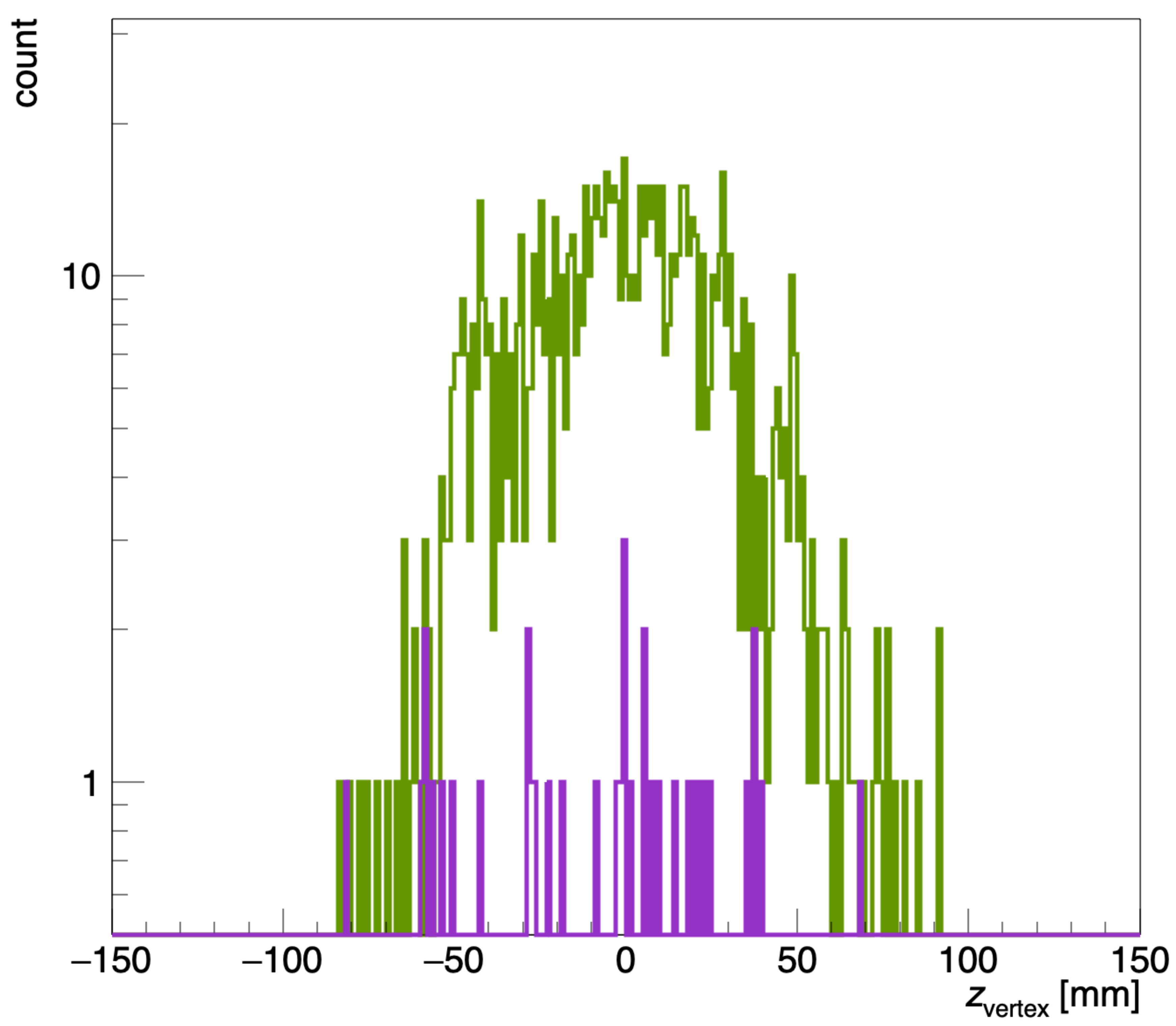


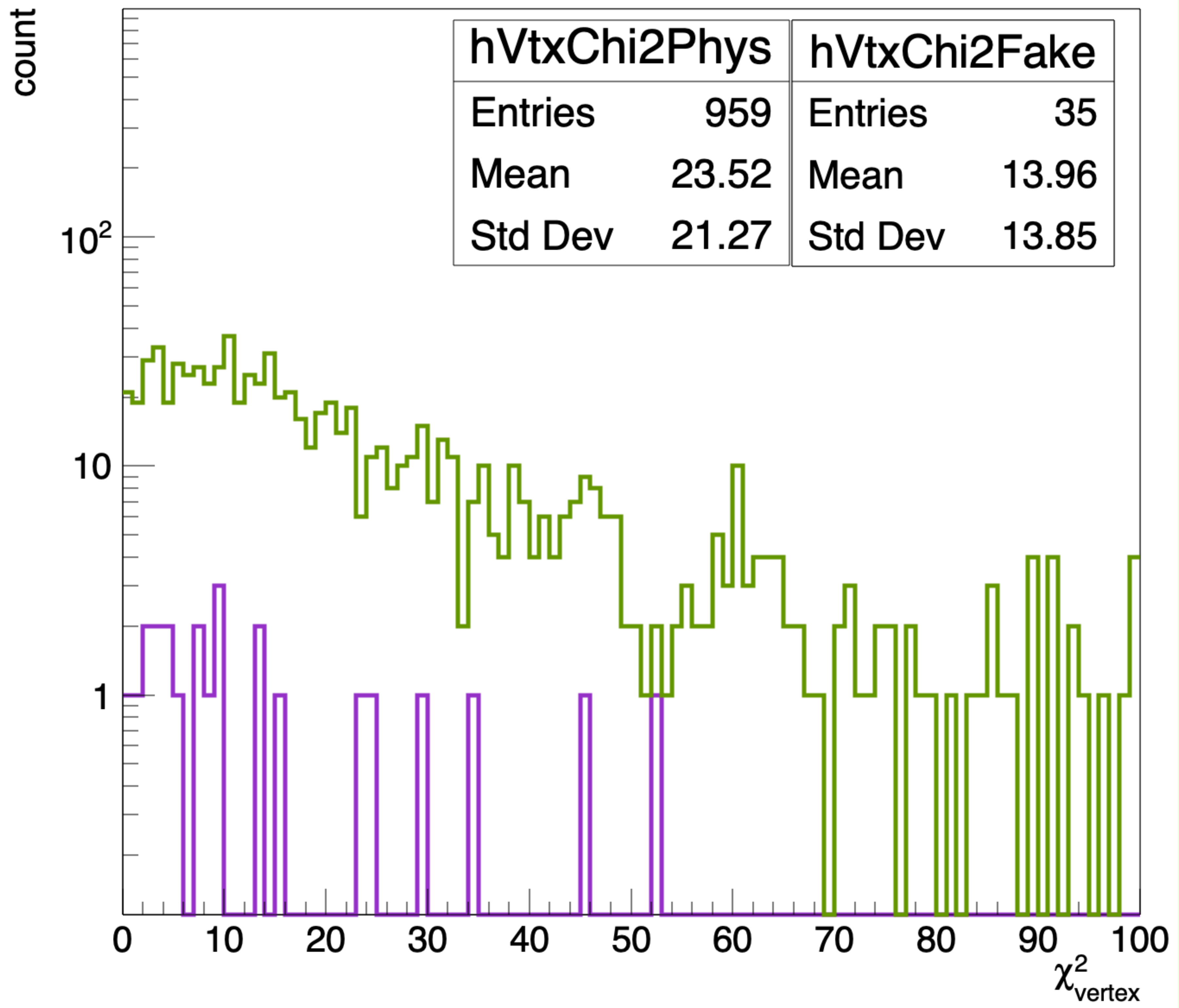
The purities of W/ time slice were a little better than W/O ones.
The efficiency of W/ time slice were mostly consistent with W/O ones, but a little worse.

Difference between physics and background time slice



Physics: 997
Fake: 2137





- The most fake time slices have only a small number of hits.
- The tendencies of values are not an obvious difference.
- Some time slices do not have a vertex despite the physics time slice.

Next Plan

- Understand what type of events are missed and when the fake triggers are generated.

- Test wiring system developed by Nathan.

- Test **multi-threads** run. (GPU usage?)

- Implement a new factory for **strict second filtering** and **event identification**.

- Apply this splitting code to various physics (Particularly, **low Q^2** simulation) and realistic event ratio （Simon’s developments should support this study.）

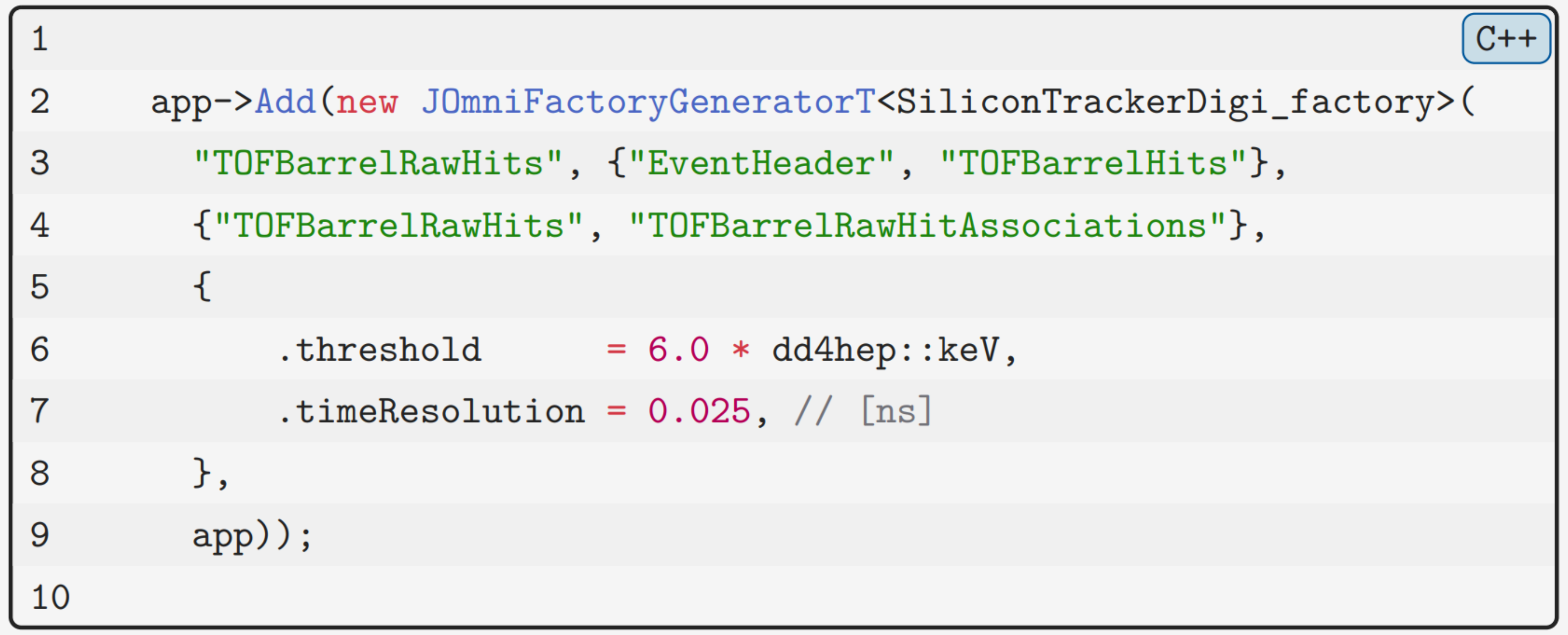
Preparing new system to re-write existing EICrecon codes

When the system changes, we often need to update the code in a consistent way.
e.g. When adding new parameters or renaming global variables.

Nathan Talk
https://indico.bnl.gov/event/30167/contributions/115026/attachments/65786/113020/2025-10-26%20external%20wiring%20update-1.pdf

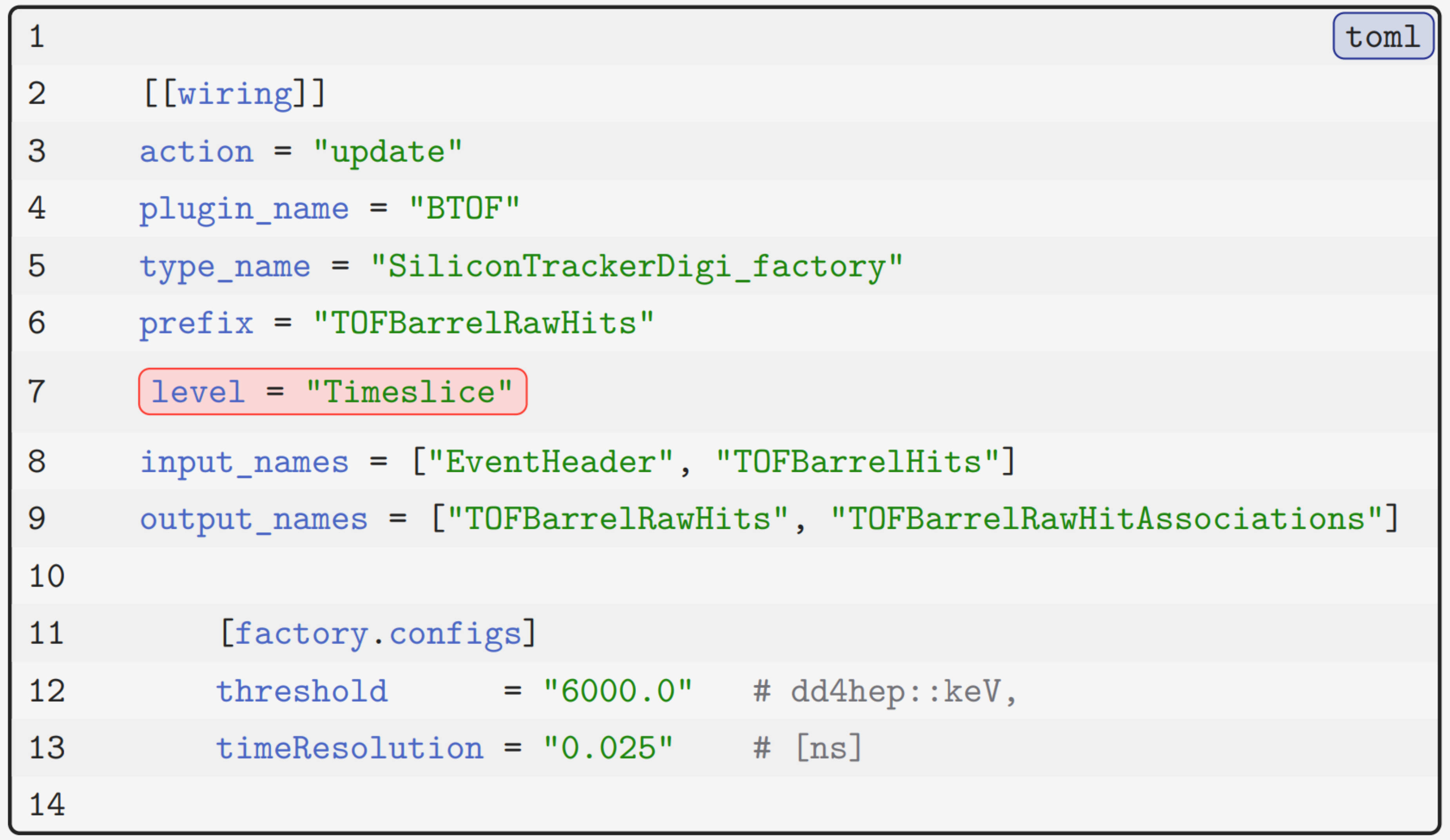
TOML: https://toml.io/en/

Just preparing and test the system with Nathan.



C++

```cpp

app->Add(new JOmniFactoryGeneratorT<SiliconTrackerDigi_factory>(
  "TOFBarrelRawHits", {"EventHeader", "TOFBarrelHits"},
  {"TOFBarrelRawHits", "TOFBarrelRawHitAssociations"},
  {
      .threshold      = 6.0 * dd4hep::keV,
      .timeResolution = 0.025, // [ns]
  },
  app));

```

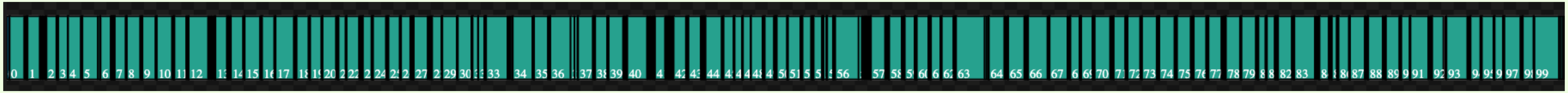


toml

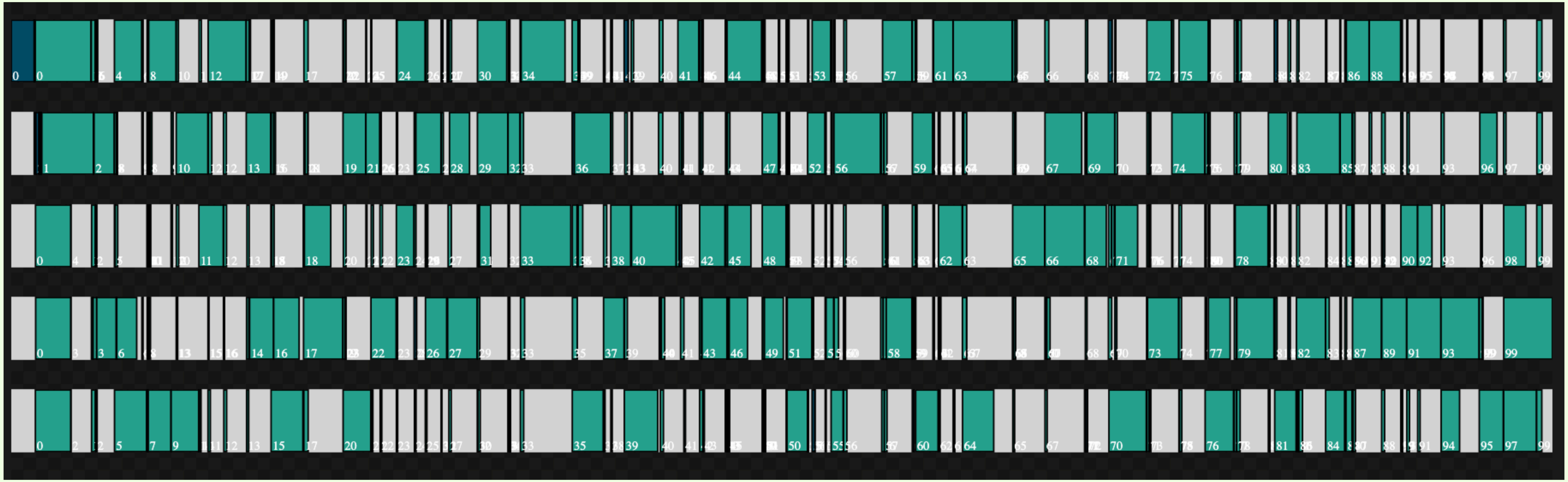
```toml

[[wiring]]
action = "update"
plugin_name = "BTOF"
type_name = "SiliconTrackerDigi_factory"
prefix = "TOFBarrelRawHits"
level = "Timeslice"
input_names = ["EventHeader", "TOFBarrelHits"]
output_names = ["TOFBarrelRawHits", "TOFBarrelRawHitAssociations"]

    [factory.configs]
    threshold      = "6000.0"   # dd4hep::keV,
    timeResolution = "0.025"    # [ns]

```

nthread test

nthread=1: 456.223 s / 100 time frame



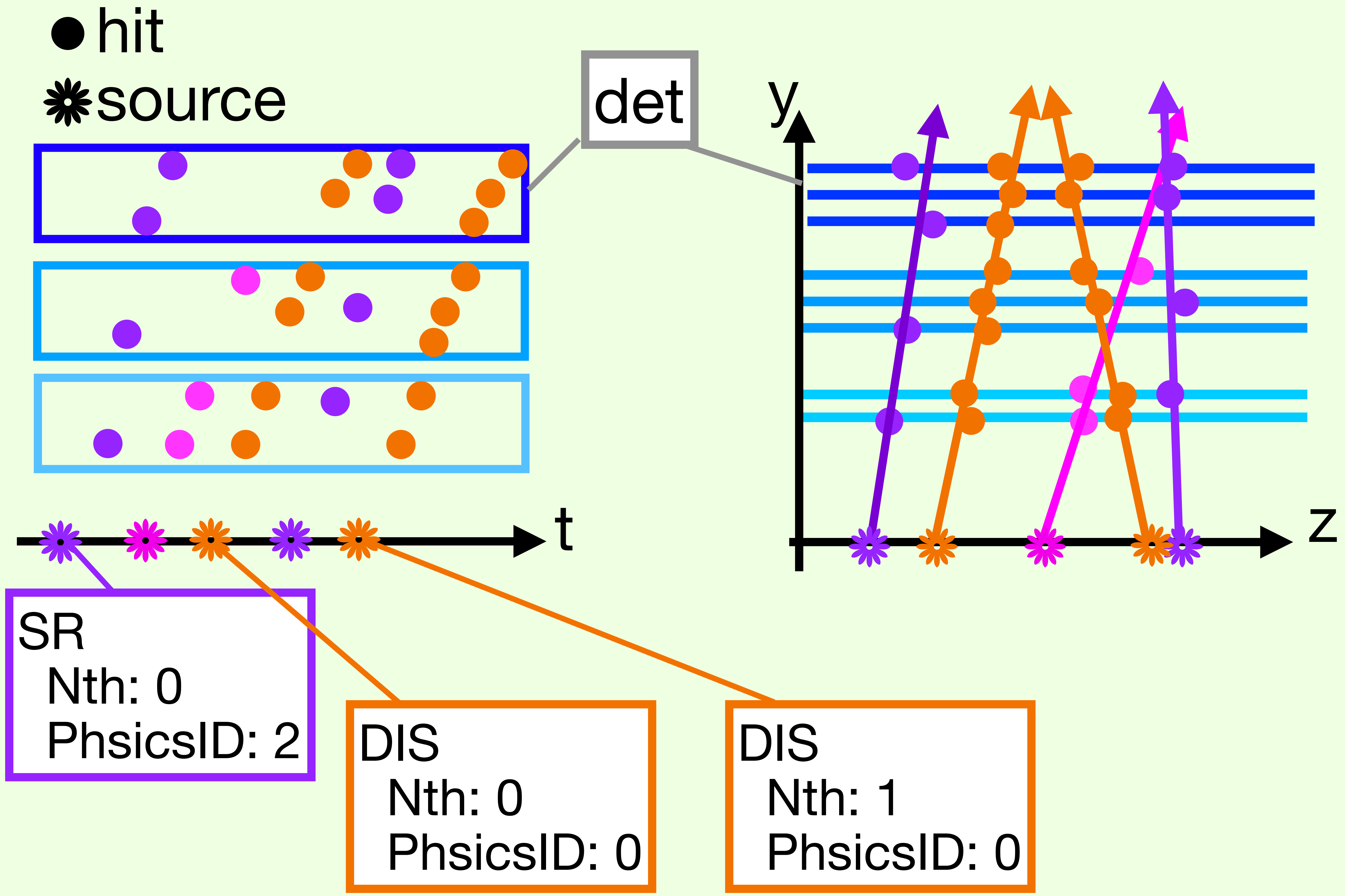
nthread=5: 541.52 s / 100 time frame



Despite increasing the number of threads correctly, the processing time increased.
→ Need to understand the Nathan's evaluator and the reason.

New Source Data Type Proposal

Currently we are missing some event source information (source time, numbering, Q^2…). We have MC particles information and it has generation status to distinguish physics and background. However, this status cannot distinguish different sources of same type source.
→The proposing data type enable to distinguish sources and give valuable information. And it is helpful for both MC and reconstruction (a little duplicate reco vertex).



```
✹ EventInfo ← NEW
  ├Nth
  ├CollisionTiming
  ├CollisionEnergy
  ├CollisionPosition
  ├CollisionAngle
  ├Q2                                background
  └PhysicsID                         ├──────────→
        [ DIS, EXCLUSIVE, SR, etc…   ]
        [ 0    1          2    …     ]

● DetRecHits
  ├ ⋮
  ├EventAssociation ← NEW
```