Hit Carryover

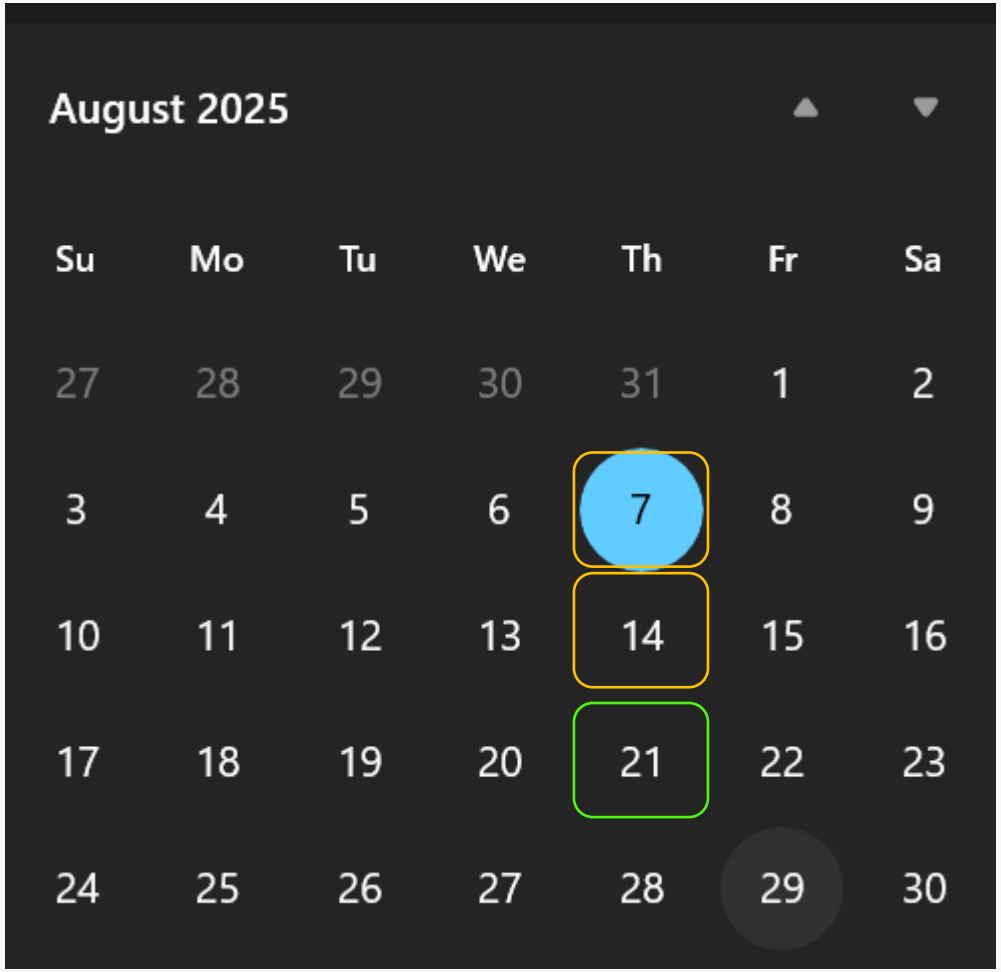
Ryotaro Koike
Kyoto University

Documentation

- I am writing an internal report on hit carryover.
- Link to an overleaf project (view only):
  https://www.overleaf.com/read/gxkgpkxvdchs#11cb64
- Questions, comments, and suggestion are welcome to improve the quality!
  Only half-written for now. This slide is most up-to-date 😐

Toward preliminary

- **I need an approval of performance preliminary for some plots.**
- **I am thinking to request it in the pre-GM on Aug 21.**
  (It's after I came back to Japan …
  I want to do it in a GM, but I need more time)
- **We will have 3 INTT meeting before then, including today.**
- **Unfortunately, I am taking a shift today. I'm happy if you take a look at this slide and discuss.**



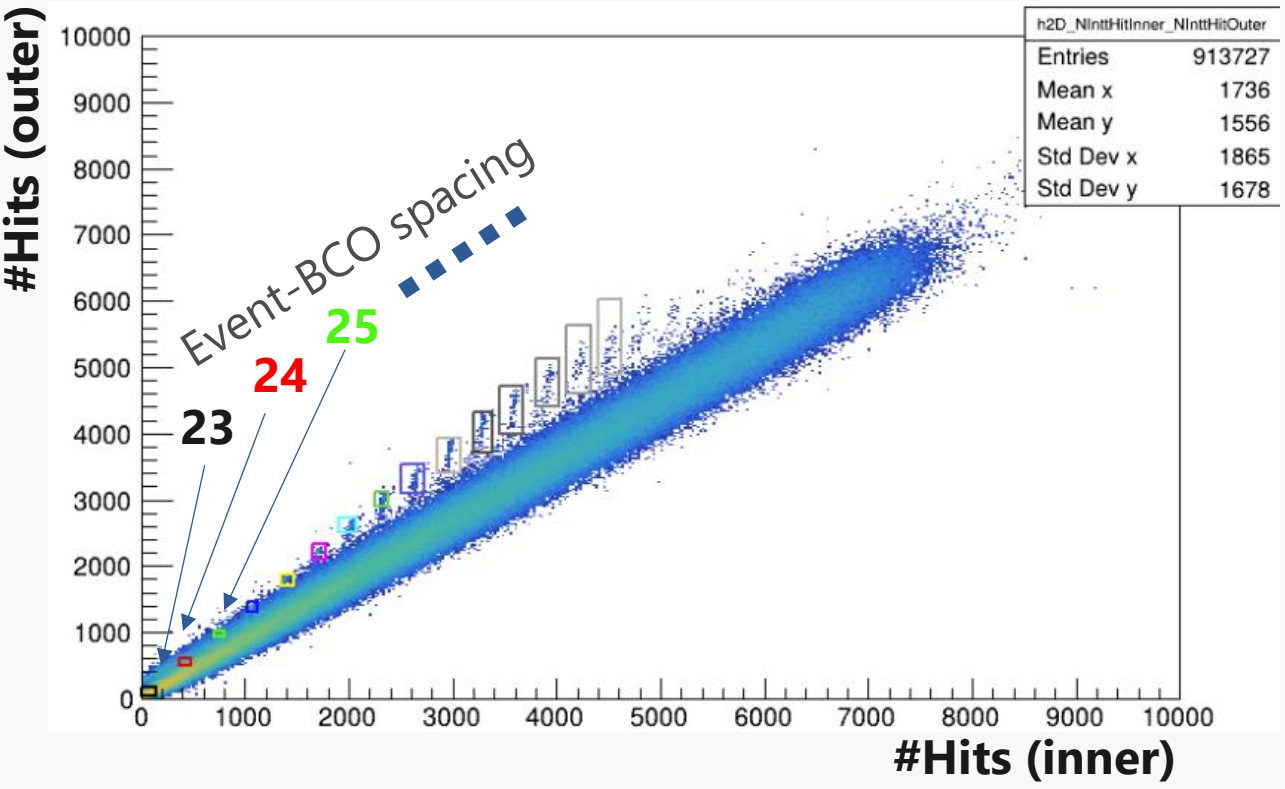
The questions we finally have answers now

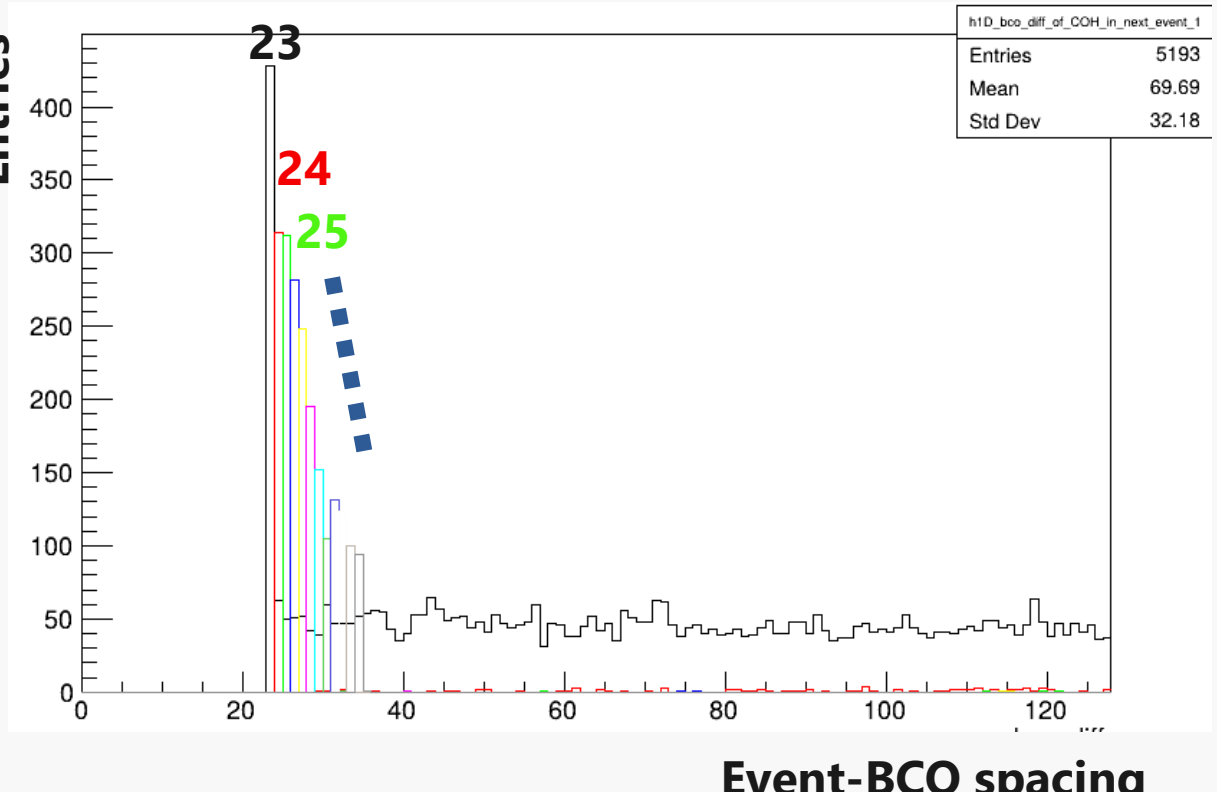
- Why a fish-bone, not a broad distribution?
- Why some fish-bones remained while others are recovered?
- Why there appears the distinction at BCO spacing of 22?
  (only when n_collision=126)

The fish-bone

- **I found the event-bco spacing characterizes each branch of the fish-bone. (Maybe Cheng-Wei has known … ?)**

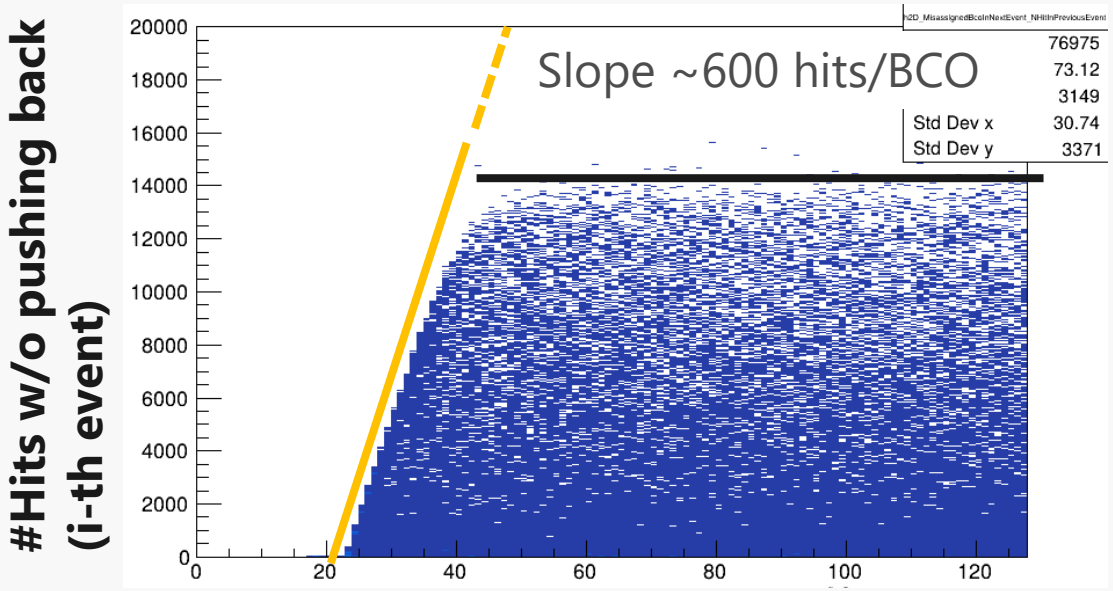
I selected events in each fish-bone and inspected their event-bco spacing.

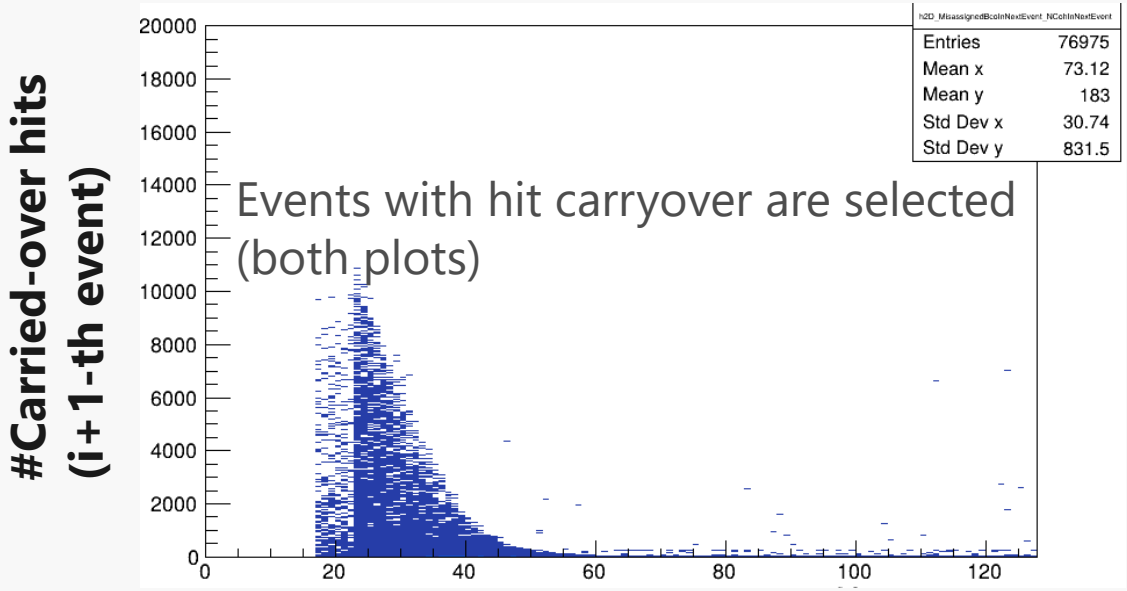




The fish-bone

- **Upper limit for the number of hits that can be processed until the next event comes is visualized in the plots below:**



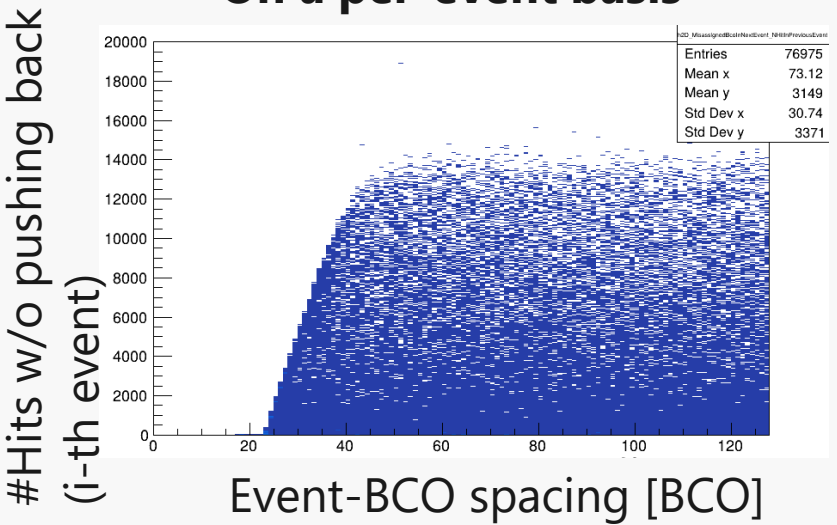


- **If the number of hits in an event exceeds this upper limit, hit carryover occurs.**
  - The upper limit varies depending on how long duration FELIX had until the next event comes.
  - If the event-bco spacing is >=60, the upper limit is large enough for most events, considering the multiplicity distribution.

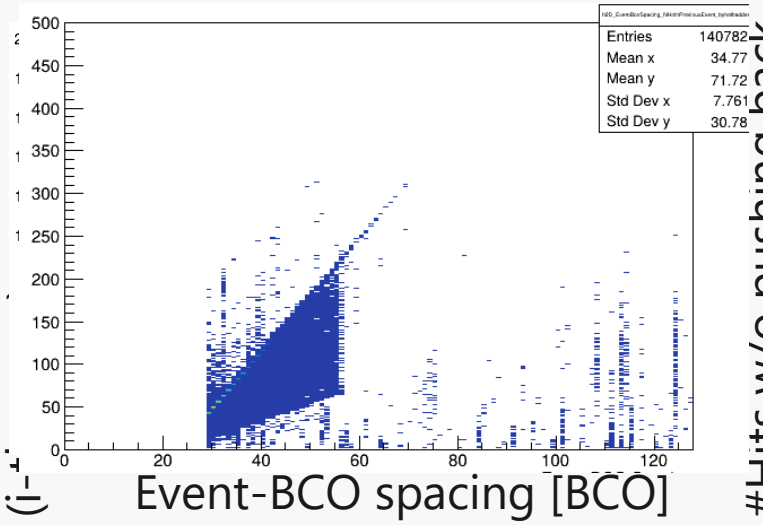
Clear Evidence of the upper limit

- **I made the same plots on a per-event basis, a per-halfladder basis, and a per-chip basis.**
- **The clear straight line-shape and the absence of entries below the line is a strong proof of:**
  - Hit carryover occurs **half-ladder by half-ladder**.
  - A constant processing capability per 1 BCO.
  - Deterministic mechanism of hit carryover.
    (Hit carryover occurs if and only if the number of hits exceeds the processing capability.)

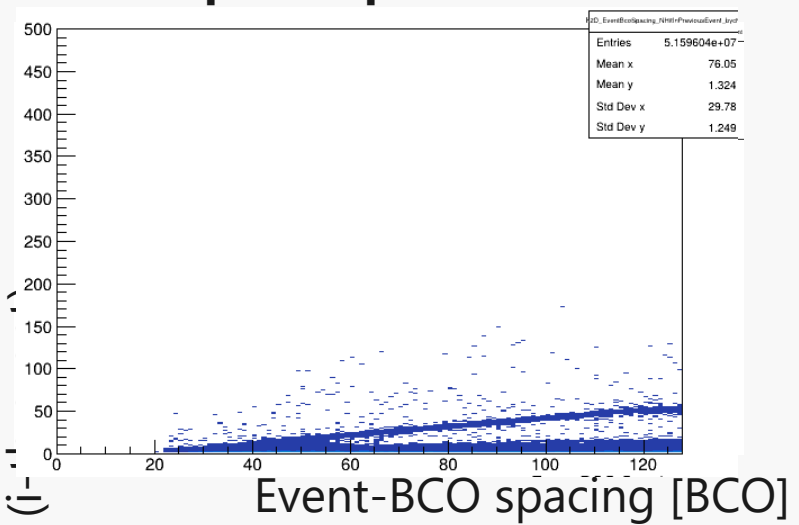
On a per-event basis



On a per-halfladder basis
(N_carried_hits in that halfladder > 6 cut is applied only for this plot)



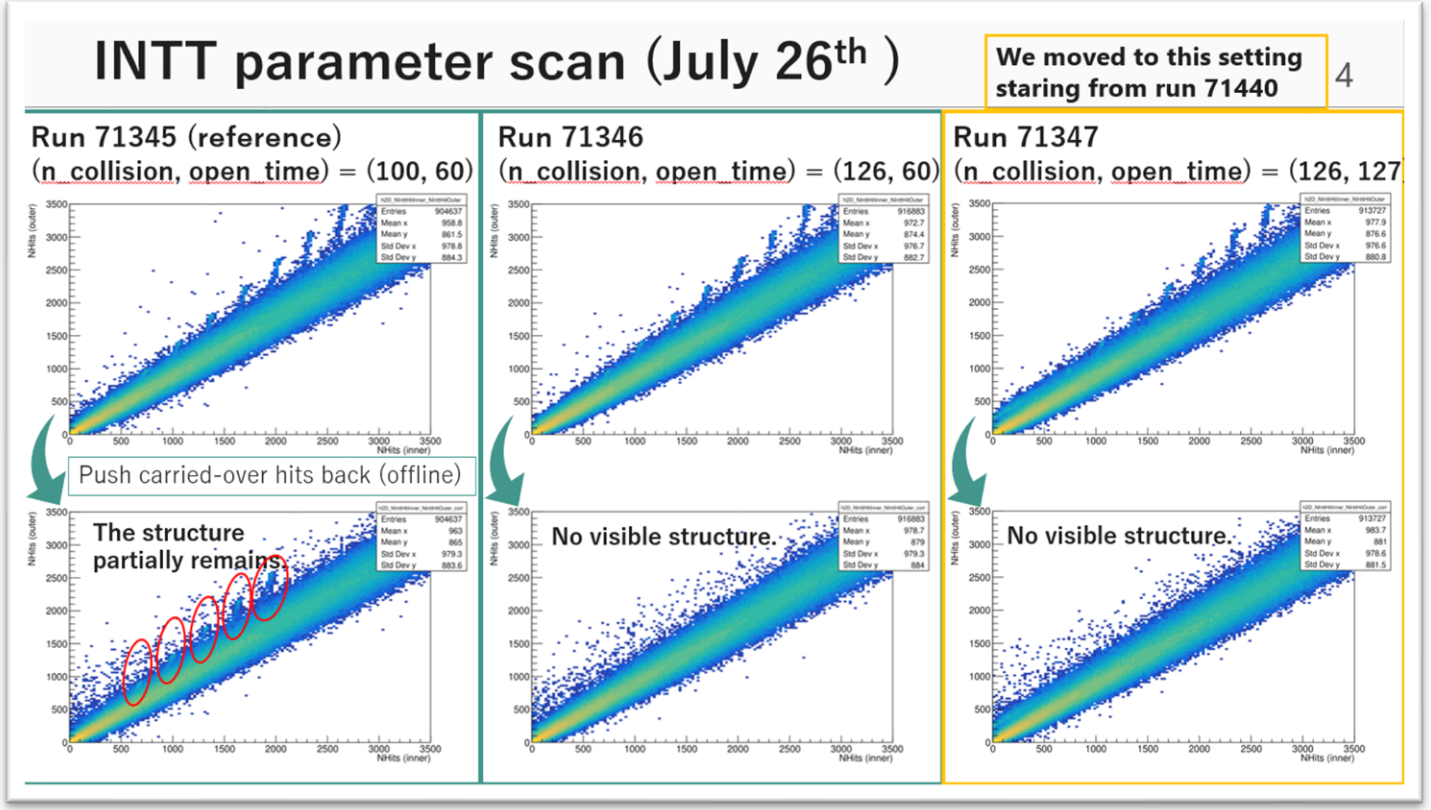
On a per-chip basis



The remaining fish-bones

- Fish-bone branches with small event-bco spacing was left while others were recovered by pushing the detected carried-over hits.
- We had some remaining fish-bones when n_collision=100, but no such things when n_collision=126.

Changing n_collision 100→126 was effective.



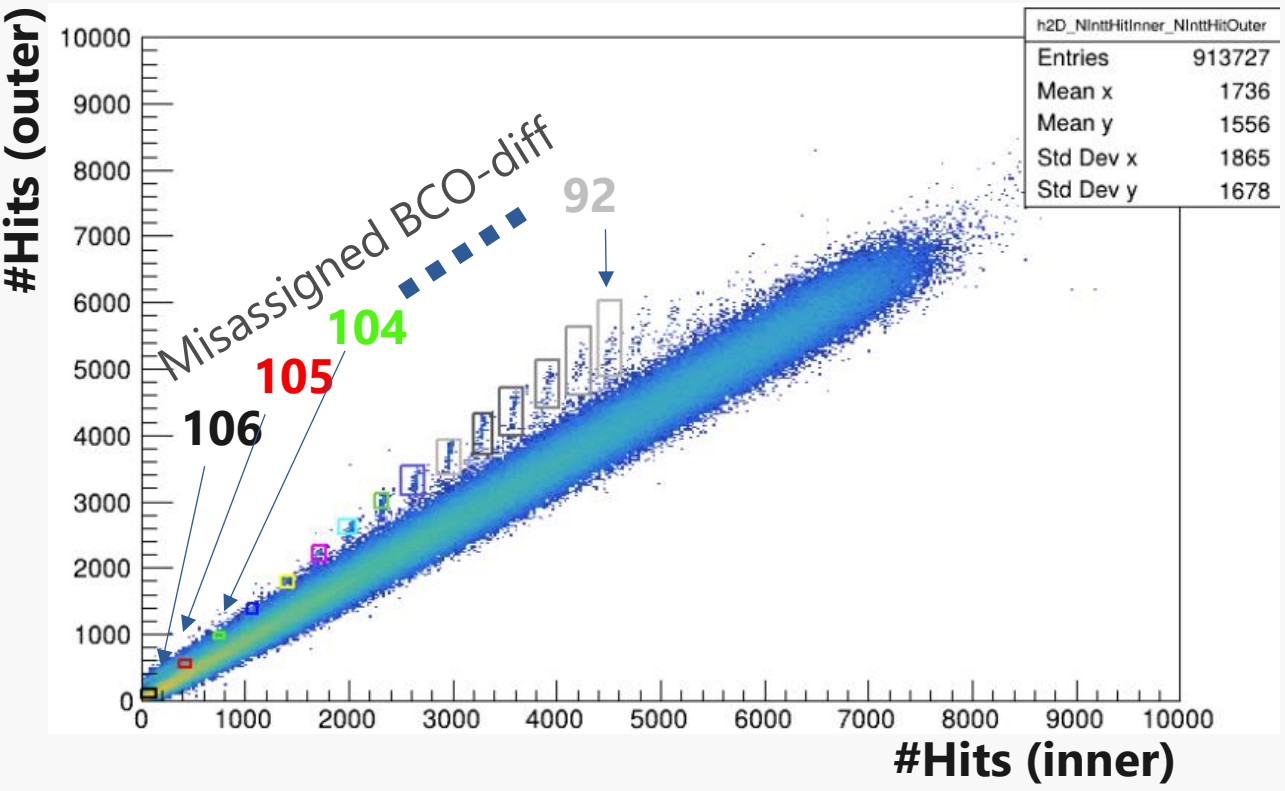
Slide from last week's weekly INTT meeting

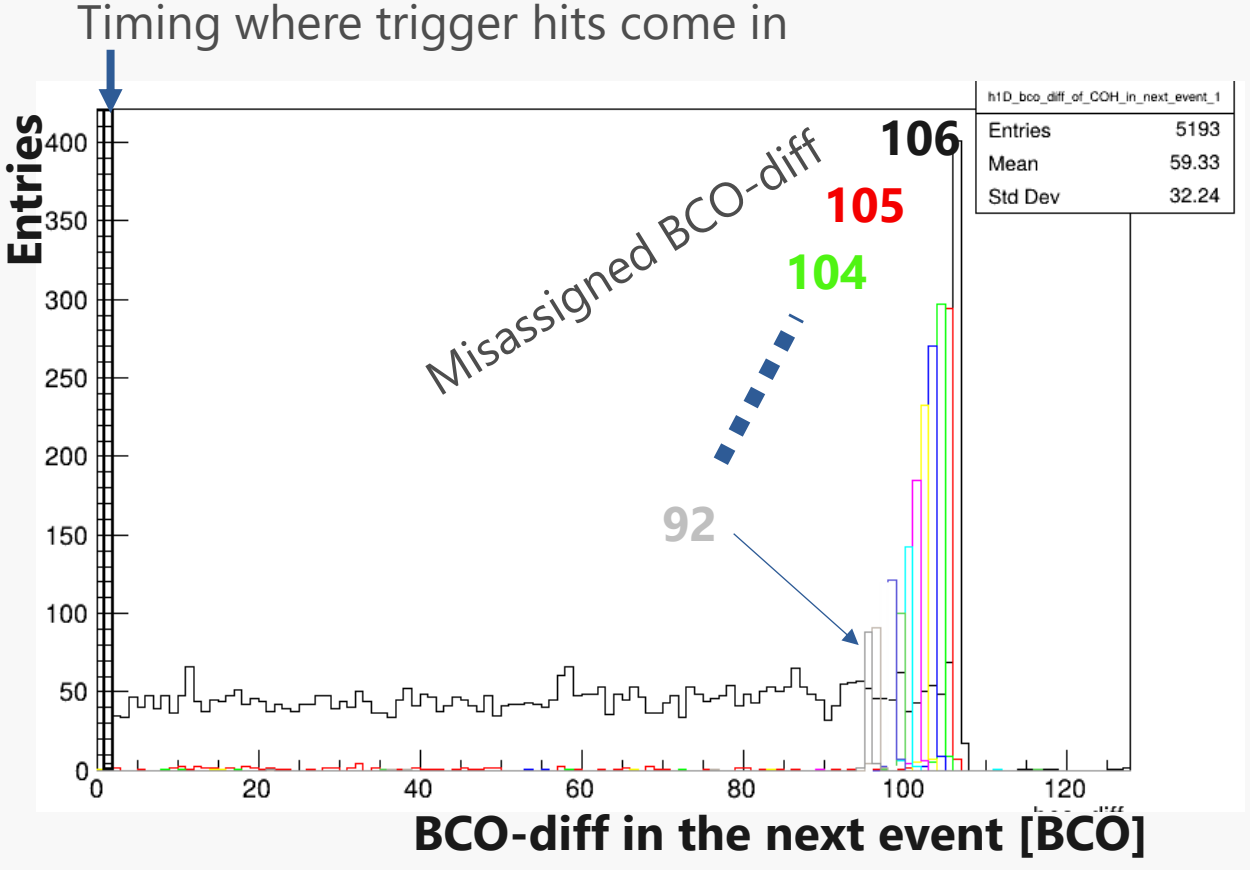
The remaining fish-bones

- We can account for them by “misassigned BCO-diff”, which is a falsely calculated BCO-diff of the carried-over hits.

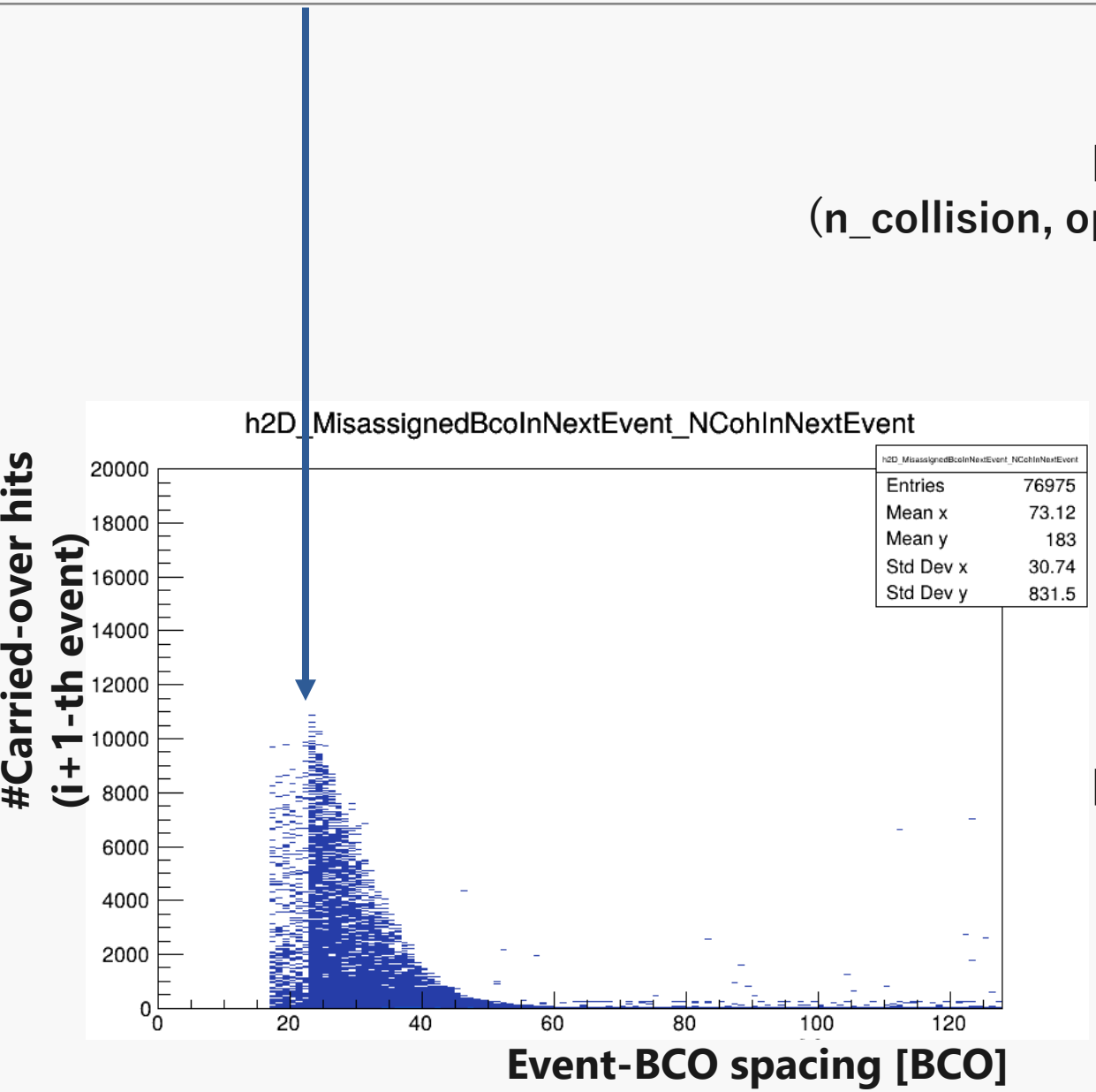
$$(\text{misassigned BCO diff}) = (\text{fphx bco})_{\text{carried-over hits from i th evvent}} - (\text{BCO full})_{i+1 \text{ th event}}$$

- Those missing hits were **rejected according to the n_collision**, based on the “misassigned BCO-diff”.

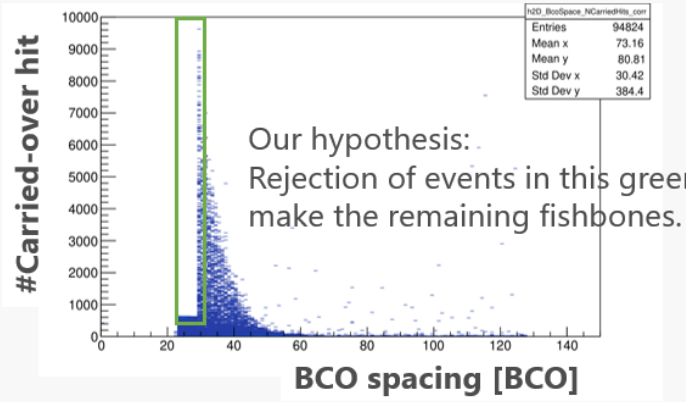




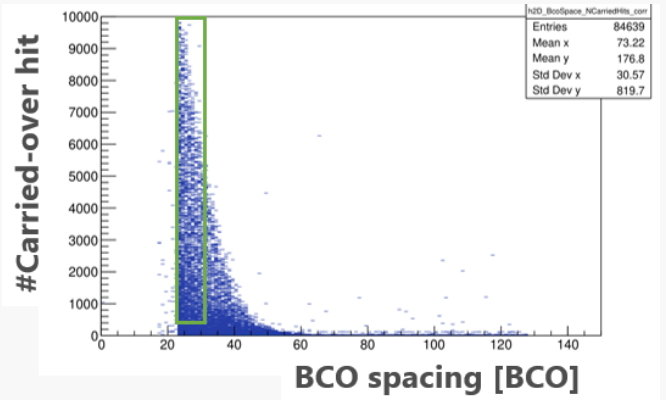
The distinction at BCO spacing of 22



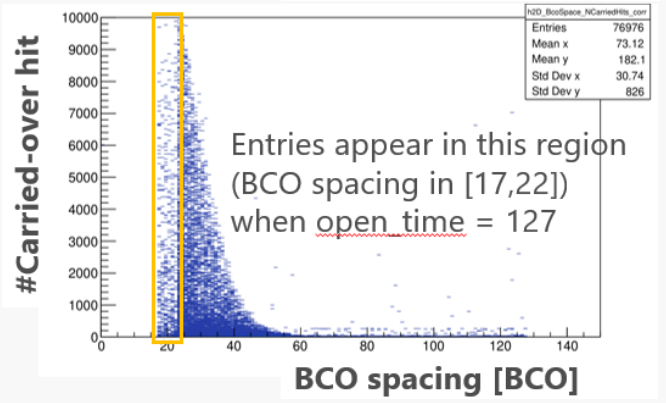
Run 71345
(n_collision, open_time) = (100, 60)



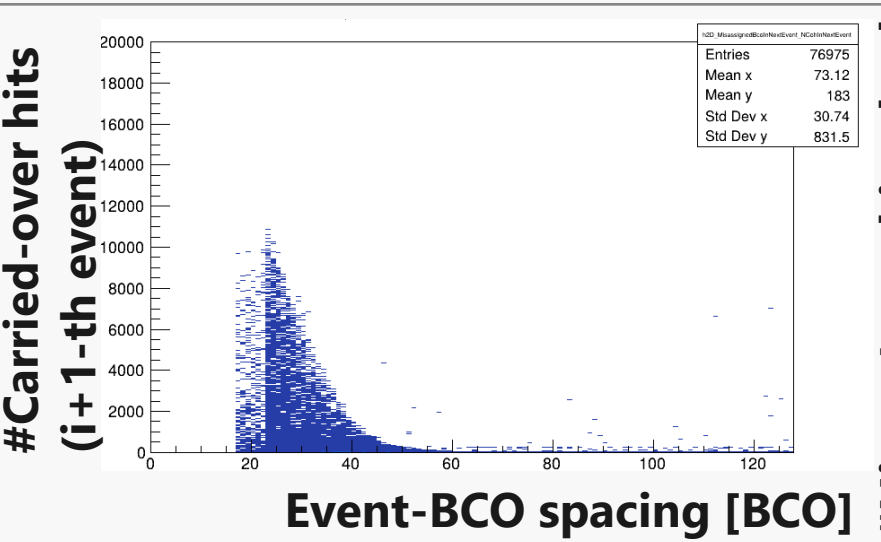
Run 71346 (126, 60)

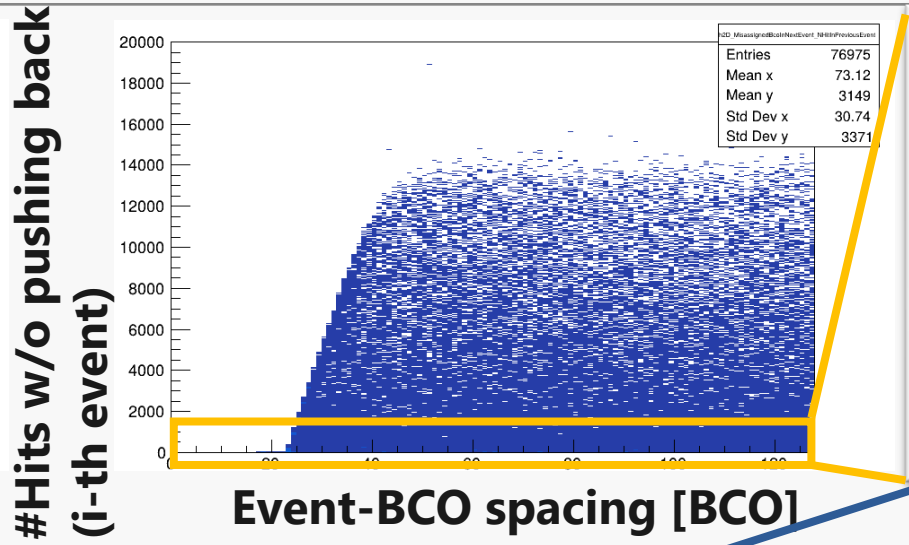


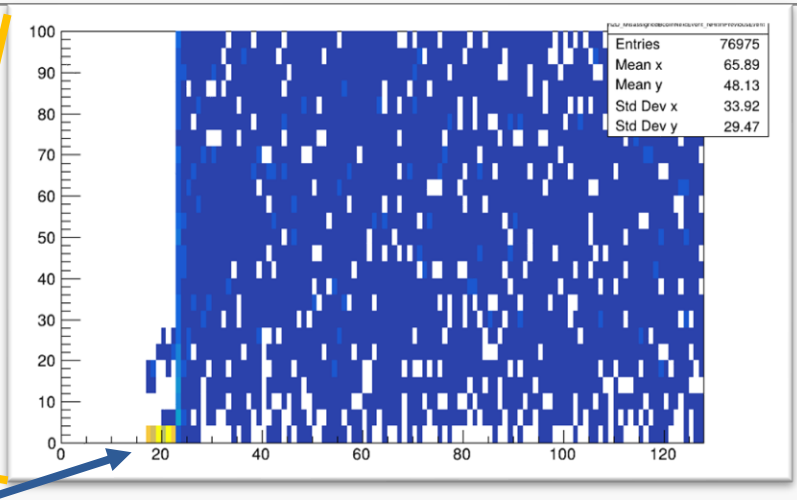
Run 71347 (126, 127)



The distinction at BCO spacing of 22







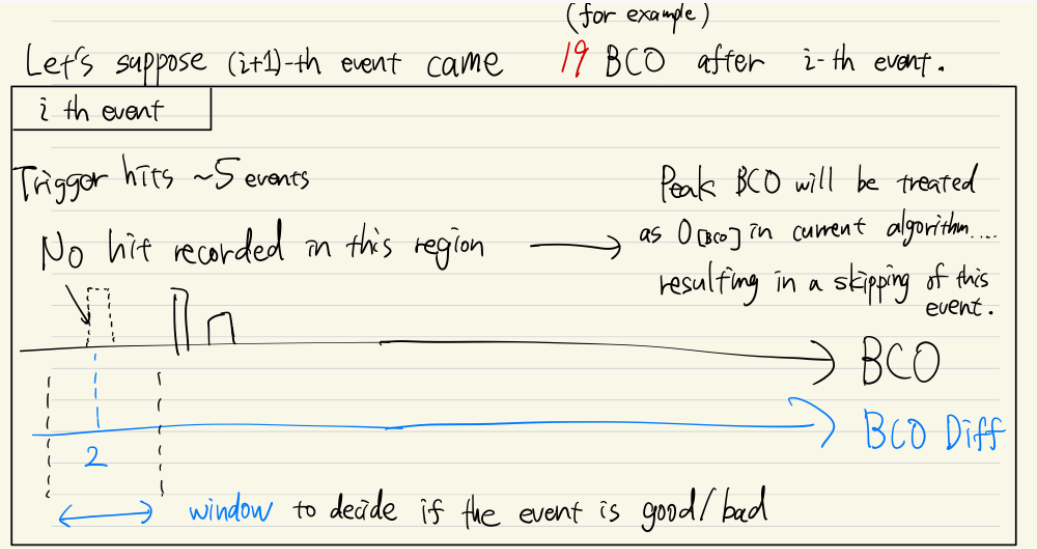
[0,16]: DAQ busy
[17,22]: missing hit carryover
[23,]: identified hit carryover

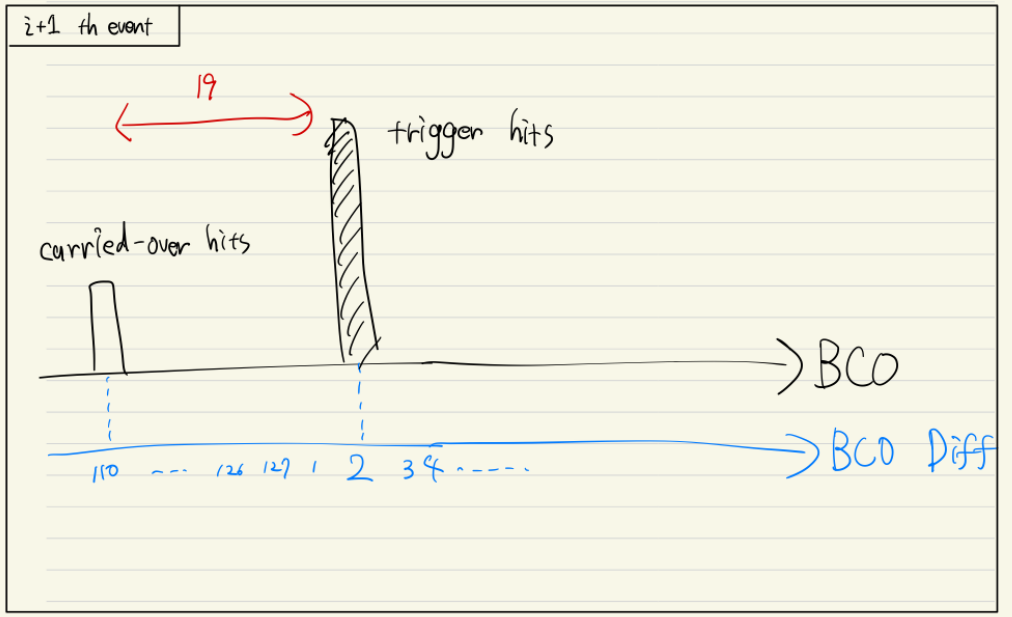
- **FELIX can <u>process only ~5 hits</u> for an event with a small BCO-spacing (17~22)**
  → Those events were probably skipped due to a BCO-diff cut in my macro.
- **Most hits of such an event would be carried-over and it's not identified.**
  - Current algorithms identifies carried-over hits in (i+1)-th event based on hit-bco in the i-th event, which would not be correctly calculated for those events (due to too little statistics).

The hit carryover that we are missing

- Considering the mechanism , there must be a hit carryover we are missing.
- For example, in a situation like this →, i-th event is now considered bad due to a bco-diff window in the algorithm, and it's skipped ...

(This will **not** corrupt/polute the next event, since such carried-over hits have completely different BCO_diff from the trigger hits.)





The hit carryover that we are missing

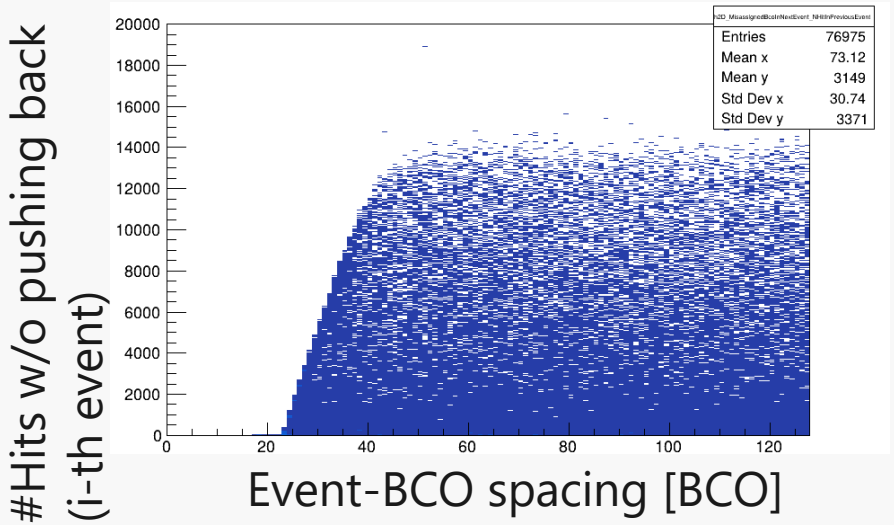
- **Candidate events found in an output log.**

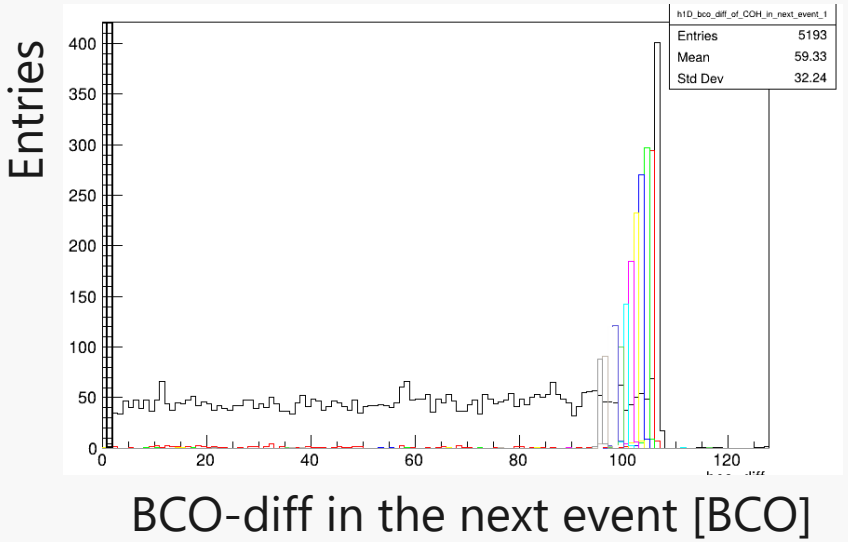
```
770500
-------------- eID: 770521, bad_evt, NInttRawHits: 84, ith_event.bco_diff_peak: 117, ith_event.hitbco_peak: 93, triggered_BcoDiff: 118
-------------- eID: 770557, bad_evt, NInttRawHits: 1, ith_event.bco_diff_peak: 103, ith_event.hitbco_peak: 0, triggered_BcoDiff: 118
event-bco spacing: 17
-------------- eID: 770579, bad_evt, NInttRawHits: 3, ith_event.bco_diff_peak: 98, ith_event.hitbco_peak: 0, triggered_BcoDiff: 118
event-bco spacing: 18
-------------- eID: 770595, bad_evt, NInttRawHits: 89, ith_event.bco_diff_peak: 119, ith_event.hitbco_peak: 83, triggered_BcoDiff: 118
770600
-------------- eID: 770650, bad_evt, NInttRawHits: 69, ith_event.bco_diff_peak: 119, ith_event.hitbco_peak: 73, triggered_BcoDiff: 118
-------------- eID: 770674, bad_evt, NInttRawHits: 519, ith_event.bco_diff_peak: 119, ith_event.hitbco_peak: 21, triggered_BcoDiff: 118
-------------- eID: 770686, bad_evt, NInttRawHits: 110, ith_event.bco_diff_peak: 117, ith_event.hitbco_peak: 14, triggered_BcoDiff: 118
770700
-------------- eID: 770731, bad_evt, NInttRawHits: 77, ith_event.bco_diff_peak: 117, ith_event.hitbco_peak: 14, triggered_BcoDiff: 118
-------------- eID: 770734, bad_evt, NInttRawHits: 8, ith_event.bco_diff_peak: 50, ith_event.hitbco_peak: 0, triggered_BcoDiff: 118
event-bco spacing: 19
-------------- eID: 770750, bad_evt, NInttRawHits: 66, ith_event.bco_diff_peak: 117, ith_event.hitbco_peak: 97, triggered_BcoDiff: 118
-------------- eID: 770759, bad_evt, NInttRawHits: 127, ith_event.bco_diff_peak: 117, ith_event.hitbco_peak: 64, triggered_BcoDiff: 118
-------------- eID: 770782, bad_evt, NInttRawHits: 2996, ith_event.bco_diff_peak: 117, ith_event.hitbco_peak: 69, triggered_BcoDiff: 118
-------------- eID: 770784, bad_evt, NInttRawHits: 155, ith_event.bco_diff_peak: 117, ith_event.hitbco_peak: 46, triggered_BcoDiff: 118
-------------- eID: 770796, bad_evt, NInttRawHits: 132, ith_event.bco_diff_peak: 117, ith_event.hitbco_peak: 58, triggered_BcoDiff: 118
770800
770900
```

Event-bco spacing shown only when ith_event.hitbco_peak == 0 && next_event.n_intt_raw_hits > 0

Conclusions

- **Why a fish-bone, not a broad distribution?**
  - The discrete processing capability makes them, depending on event-bco spacing.





- **Why some fish-bones remained while others are recovered?**
  - Carried-over hits from the remaining events were rejected according to n_collision.

- **Why there appears the distinction at BCO spacing of 22? (only when n_collision=126)**
  - Recording of hits starts 23 bco after the previous event.
  - Current algorithm misses hit carryovers with a spacing in [17,22].

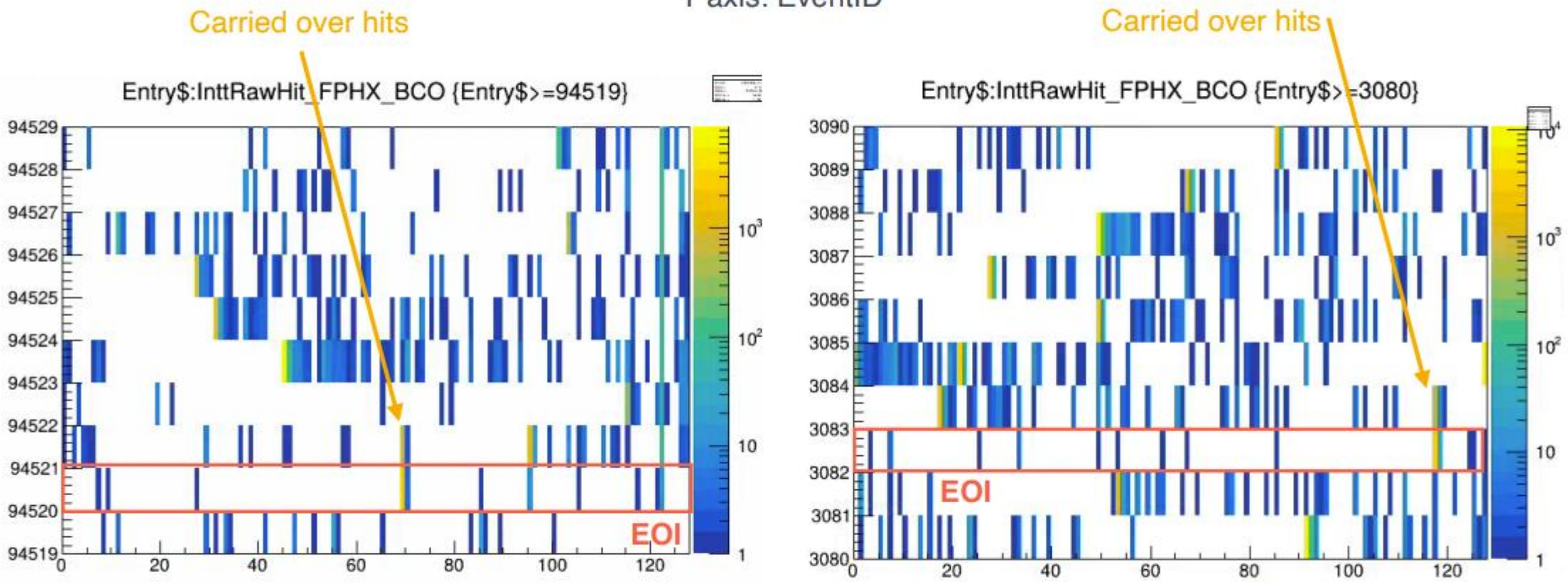
Remaining questions

- **Why the inner layers have more carried-over hits than outer layers?**
  - Inner ladders have a lower threshold?
- **Does a carryover to next-next event occur?**
  - Yes. We would be able to identify it, although the probability and the impact would be negligible.
- **Effects of open_time on hit carryover.**
  - Investigating …
- **How to push back the carried-over hits that we are missing.**
  - I want to apply a different logic for events with an event-bco spacing of 17-22.

Backup

Event display, hit_bco vs EventID

X axis: hit_bco (0 - 127)
Y axis: EventID



Throughout the preceding 10 events, there is no more carried-over-like hit

We have events with carried-over hit issue that cannot be fully recovered. It seems that the hits are just not in the file

https://indico.bnl.gov/event/27318/

The remaining fish-bones

- **Carried-over hits with misassigned BCO-diff ≤ 100: fully recovered.**
- **> 100: shift to next**

Why?
If they don't move, it's simple…