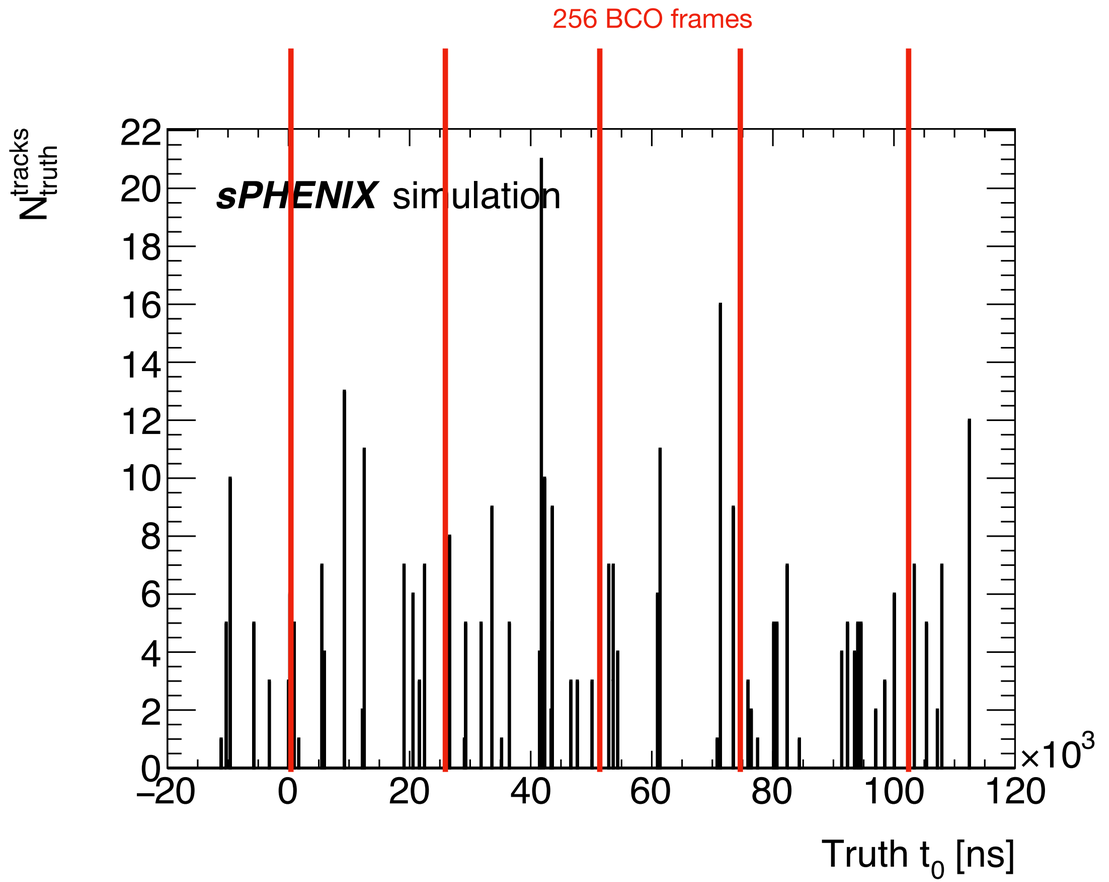
# 4D Tracking with Streaming Readout

Joe Osborn
BNL
February 6, 2024

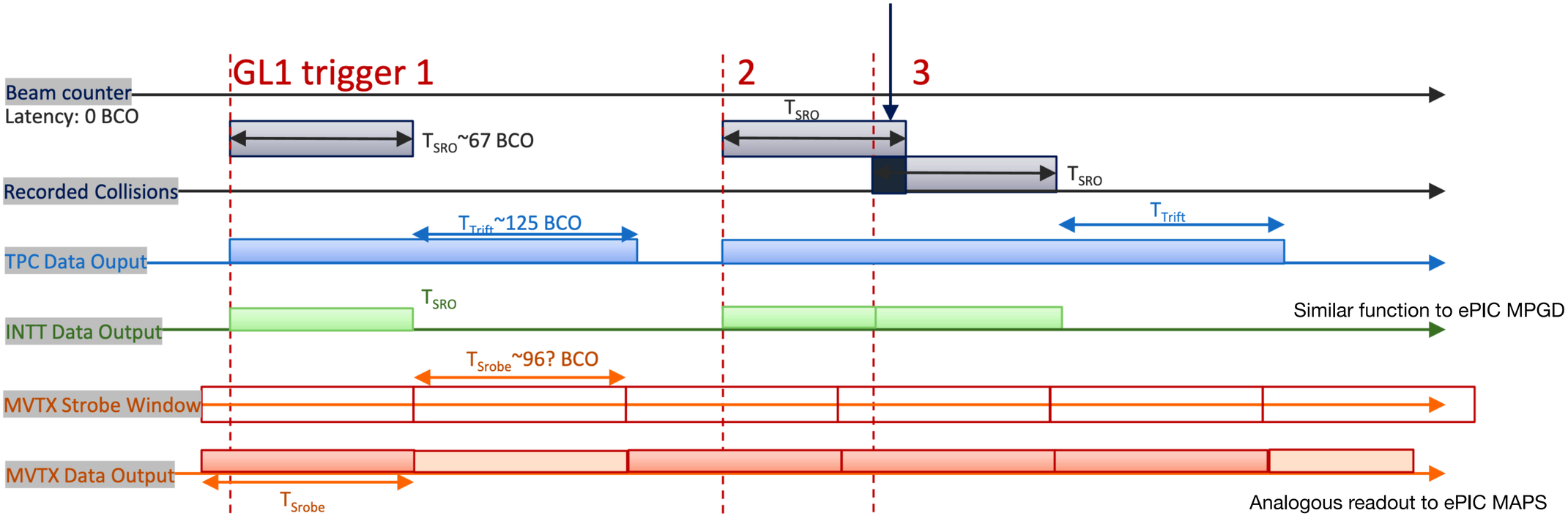
# Streaming Tracking

- Redefine terms:
  - Event == a physics event corresponding to a bunch crossing where an interaction happened, producing particles
  - Trigger/time frame == a chunk of streamed data in time where reconstruction is performed
- Time frame building and reconstruction has to process data in time frames, considering data at time frame boundaries and any duplicated data
- Output of reconstruction is tracks assigned to vertices



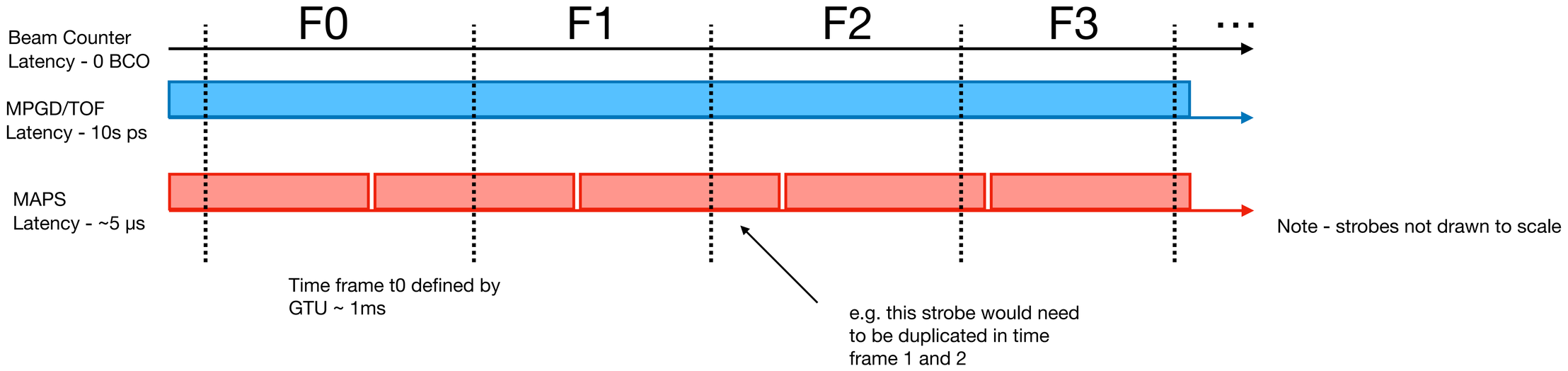

- Example of 100 μs free streaming in sPHENIX environment, to be divided into ~26 μs time frames
- ePIC SRO plans for building ~1 ms time frames sorted in data files

# Streaming Tracking




- Necessary to synchronize all subsystems in time
- Reconstruction is presented with all tracks and vertices from a time/trigger frame
- Time/trigger frame is defined by various subsystem readouts, data rates, storage sizes, etc.

# Streaming Tracking




- ePIC will be 100% streaming, so we won't have a hardware trigger + extended readout
  - Simpler case, because every frame can be treated the same
- Chunk data into ~1ms time frames, where readout windows will in general not coincide with time frames
- Requires duplication of some data at frame boundaries
  - Small effect for ePIC - 2 μs strobe with 1 ms frame gives ~1/500 strobes need to be duplicated

# Integrating Timing to Tracking

- Upon hit creation, construct unique surface identifier that carries both spatial (channel) and timing (crossing) information
- Examples:
- MVTX - determine strobe window relative to triggered crossing
- INTT - determine crossing value based on rate and time G4hit was created

```
uint64_t bcodiff = gl1bco - strobe;
double timeElapsed = bcodiff * 0.106;  // 106 ns rhic clock
int index = std::floor(timeElapsed / m_strobeWidth);

if (Verbosity() >= VERBOSITY_A_LOT) mvtx_hit->identify();

const TrkrDefs::hitsetkey hitsetkey = MvtxDefs::genHitSetKey(layer, stave, chip, index);
```

Data MVTX hit reco

```
for (unsigned int i_rep = 0; i_rep < n_replica; i_rep++)
{
  int strobe = t0_strobe_frame + i_rep;
  // to fit in a 5 bit field in the hitsetkey [-16,15]
  if (strobe < -16) strobe = -16;
  if (strobe >= 16) strobe = 15;

  // We need to create the TrkrHitSet if not already made - each TrkrHitSet should correspond to a chip for the Mvtx
  TrkrDefs::hitsetkey hitsetkey = MvtxDefs::genHitSetKey(layer, stave_number, chip_number, strobe);
```

MVTX simulation digitization

```
ofl = InttNameSpace::ToOffline(raw);
hit_key = InttDefs::genHitKey(ofl.strip_y, ofl.strip_x);  // col, row <trackbase/InttDefs.h>
hit_set_key = InttDefs::genHitSetKey(ofl.layer, ofl.ladder_z, ofl.ladder_phi, intthit->get_bco() - gl1bco);
```

Data INTT hit reco

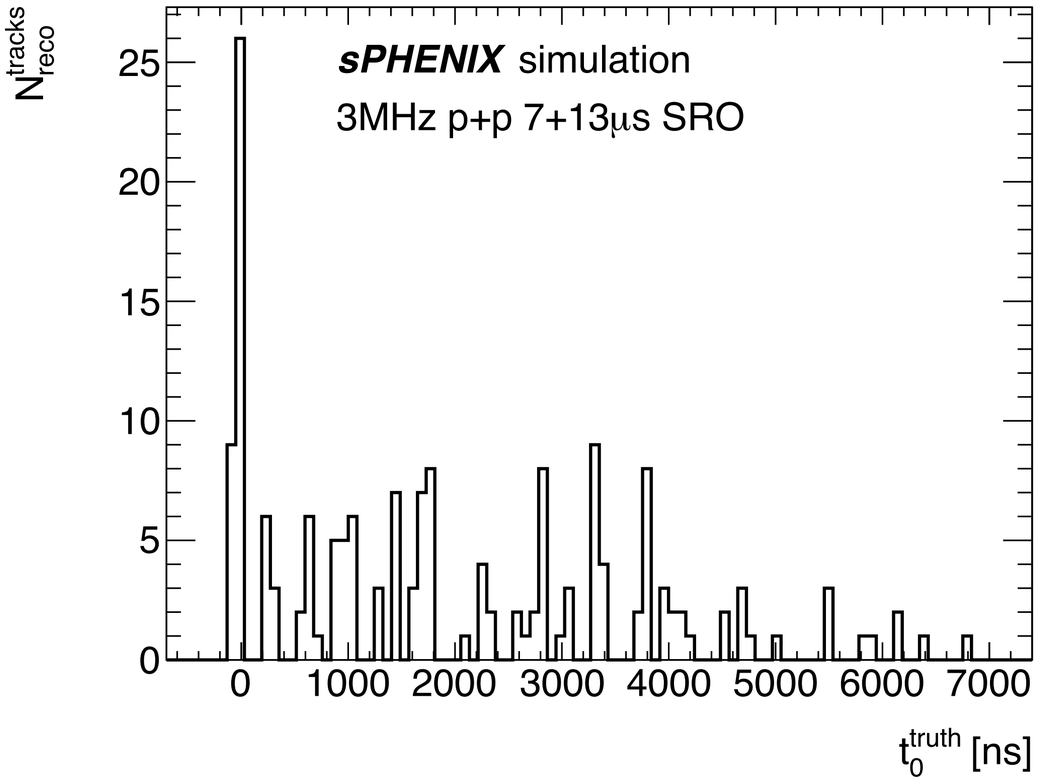
```
// Get the hit crossing
int crossing = (int) (round( time / m_crossingPeriod) );
// crossing has to fit into 5 bits
if(crossing < -512) crossing = -512;
if(crossing > 511) crossing = 511;
// We need to create the TrkrHitSet if not already made - each TrkrHitSet should correspond to a sensor for the Intt ?
// The hitset key includes the layer, the ladder_z_index (sensors numbered 0-3) and  ladder_phi_index (azimuthal location of ladder) for this hit
TrkrDefs::hitsetkey hitsetkey = InttDefs::genHitSetKey(sphxlayer, ladder_z_index, ladder_phi_index, crossing);
```

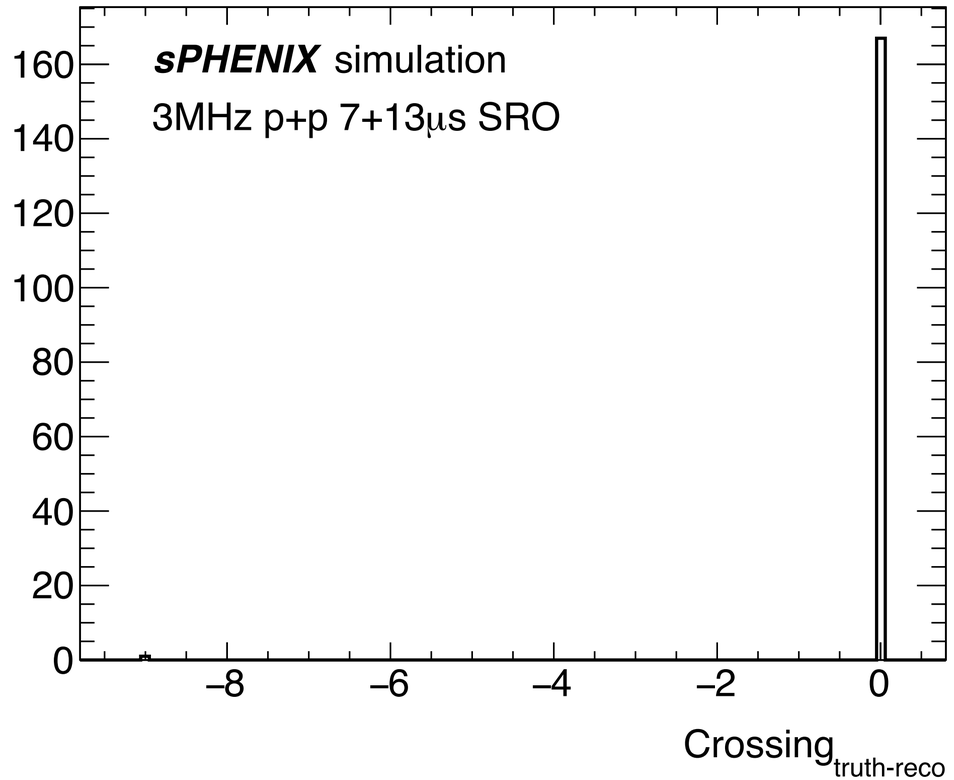
INTT simulation digitization

# Track Reconstruction with Timing

- Need some global timing with which to correlate hits to each other
- For ePIC this will likely be the start of the time frame given by the GTU
  - Can assign MAPS hits to relative strobe with respect to GTU GL1
  - Can assign MPGD/TOF hits to relative crossing number with respect to GTU GL1
- Assign tracks a determined crossing number based on their cluster composition

# Example




- 10 extended readout events in sPHENIX environment
- Reconstruct tracks from all 7 μs of streamed data
- Correctly assign bunch crossing relative to $t_0$ given by GL1 trigger to 167/168 tracks

# Track Reconstruction with Timing

- In sPHENIX we match TPC and silicon tracklets in eta/phi/$PCA_{xy}$
- From those matches, determine whether or not they match in crossing and $PCA_z$ based on the TPC drift velocity
- In ePIC could imagine doing something similar with the MPGD/TOF
  - It may be enough to look at $\chi^2$ contribution given by MPGD hits to overall track. For out of time silicon+MPGD matches, the $\chi^2$ contribution should be nominally much larger for the MPGD clusters than the silicon clusters (if out of time match)
  - The Acts determined $\chi^2$ is a very good discriminator for identifying which track match is “self-consistent”
  - e.g. $\chi^2$ in some dummy tests increases by factor of 5-10 for +/-2 crossings from the correctly matched crossing

# Track Reconstruction with Timing

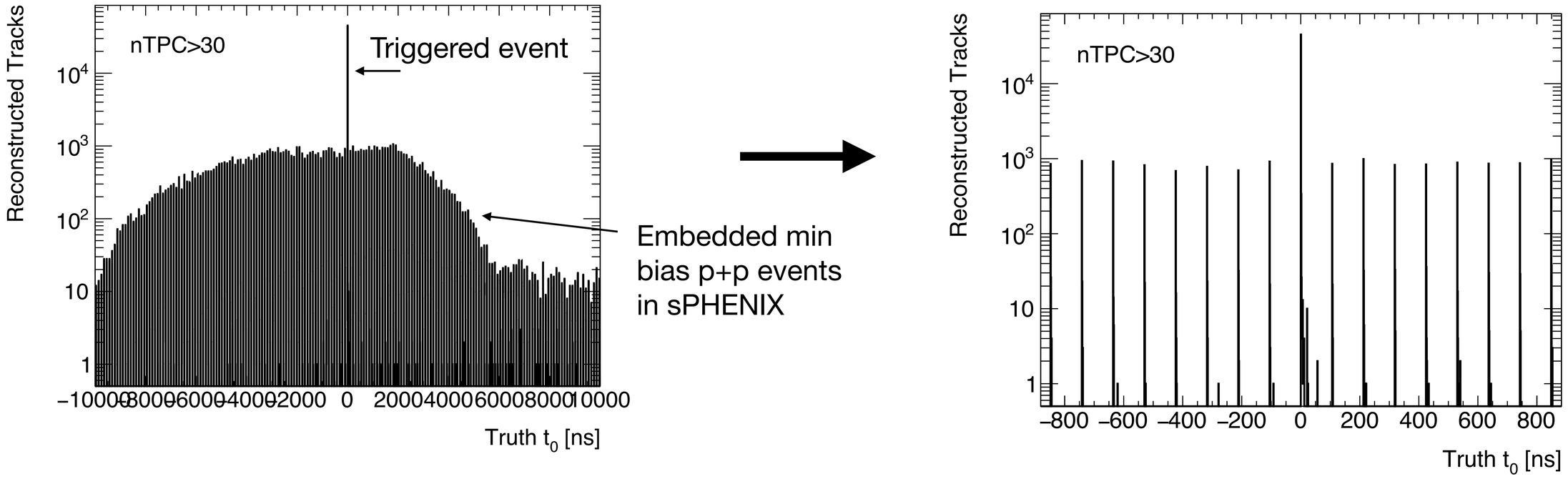
- In ePIC we require the timing information within the tracking clusters because the MPGD
  - sPHENIX TPC gives us this “for free” because drift time == TPC cluster z position
- MPGDs have a local 2D measurement + time
- Track reconstruction uses Combinatorial Kalman Filter, which performs both track finding and fitting in the same step
  - Time has to be used as a discriminator in both track finding and track fitting - therefore must be integrated into our hits and calls to Acts

# Final Thoughts

- Not possible to separate reconstruction from time frame
- Not uniformly possible to build real physics events, depending on what type of analysis you want to do
  - Example - is some track a highly displaced track from in time primary vertex or a primary track from a primary vertex 3 beam crossings away?
    - Needs to be dealt with at analysis level and not necessarily reconstruction - depends on whether or not looking for some HF decay or not
  - Going from reconstruction to analysis will require some DST “filtering” e.g. Turbo at LHCb, skimming at CMS, DAOD at ATLAS, <insert name> at <your favorite experiment>
- Streaming with MAPS involves some data duplication at frame boundaries - result of MAPS readout on LHC clock
  - In hit reconstruction, need to sync MAPS strobe with some reference clock (e.g. time frame $t_0$)
- For current track reconstruction workflow, I suspect we will need measurement time to be explicitly included in track finding + fitting

# Back Up

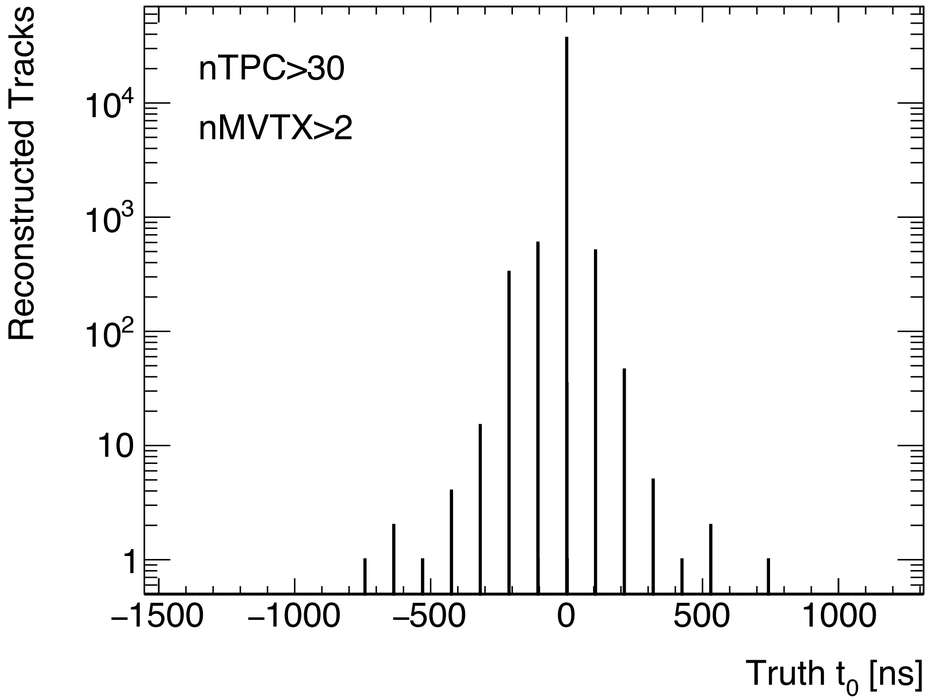
# Zeroth Order Tracking with Timing

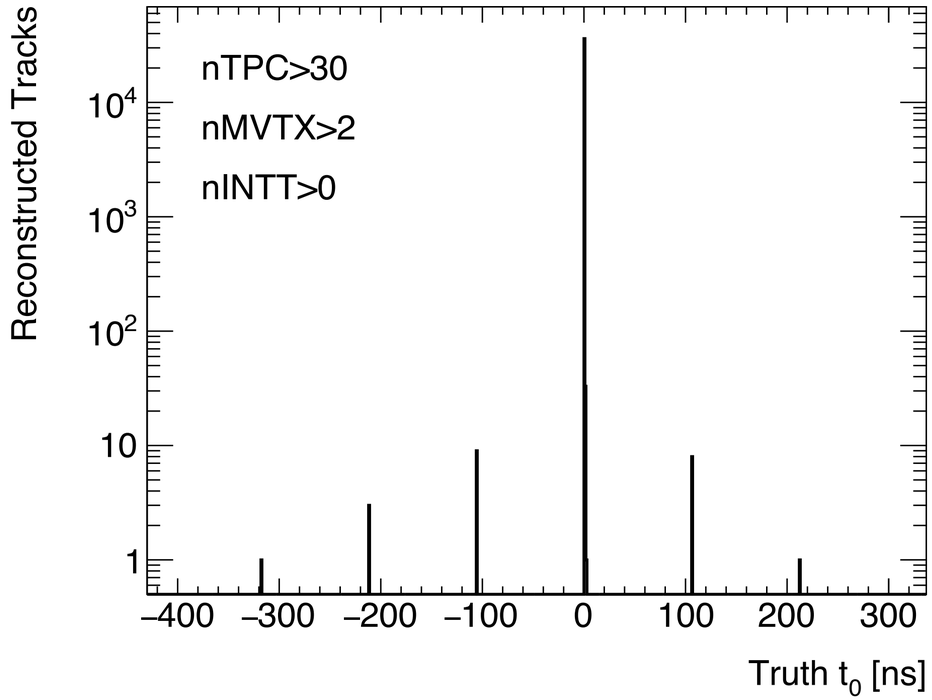



- Looking at tracks with only TPC clusters is sensitive to full RHIC 106 ns bunch structure

- ePIC should see similar behavior except 1/98.5MHz ~ 10 ns

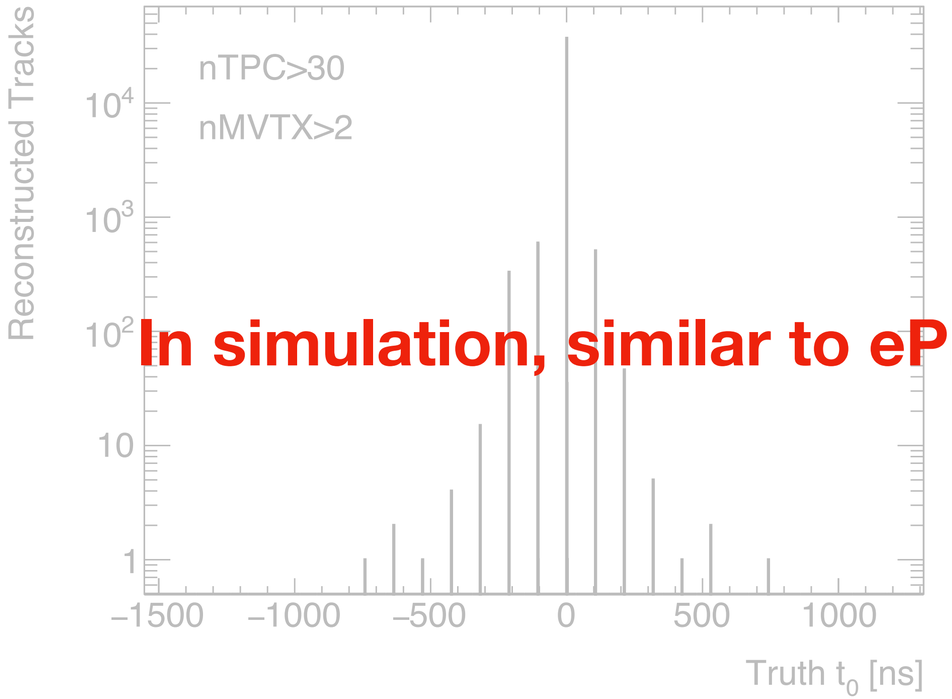
# Zeroth Order Tracking with Timing

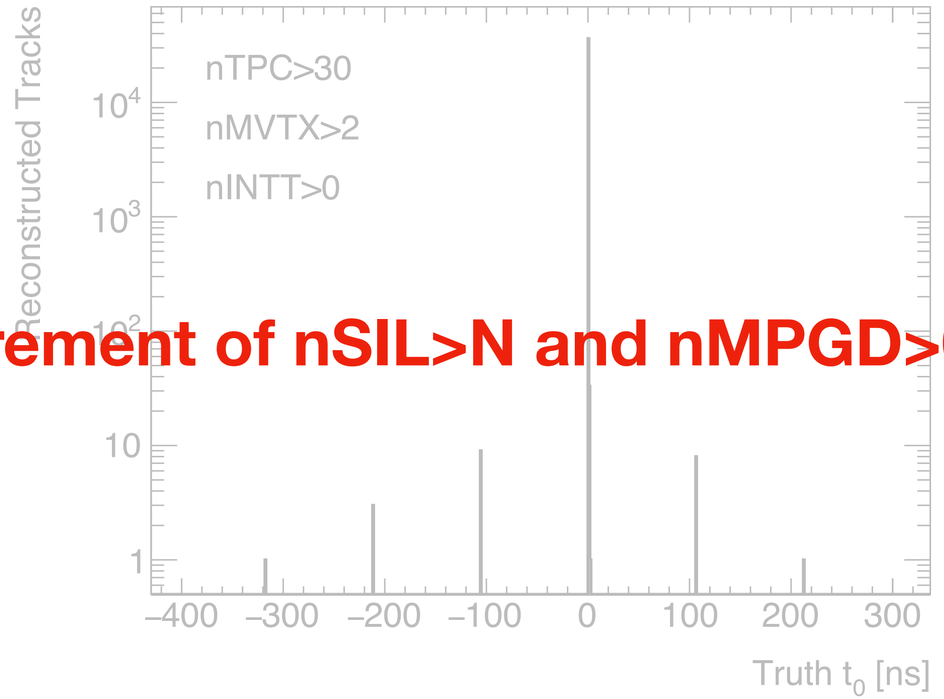



- Requiring clusters on track with smaller timing window limits to the triggered bunch crossing
- Simply a result of “artificially” matching timing windows of various subsystems
- This only gets you so far - e.g. doesn’t account for dead/hot areas (so timing in the hits/tracking is ultimately necessary). But in a simulation, it can already show pile up rejection power

# Zeroth Order Tracking with Timing




**In simulation, similar to ePIC requirement of nSIL>N and nMPGD>0**

- Requiring clusters on track with smaller timing window limits to the triggered bunch crossing
- Simply a result of “artificially” matching timing windows of various subsystems
- This only gets you so far - e.g. doesn’t account for dead/hot areas (so timing in the hits/tracking is ultimately necessary). But in a simulation, it can already show pile up rejection power