July 6th, 2023

MuCol software training

Beam Induced Background

for realistic events at Muon Collider

Wiki page

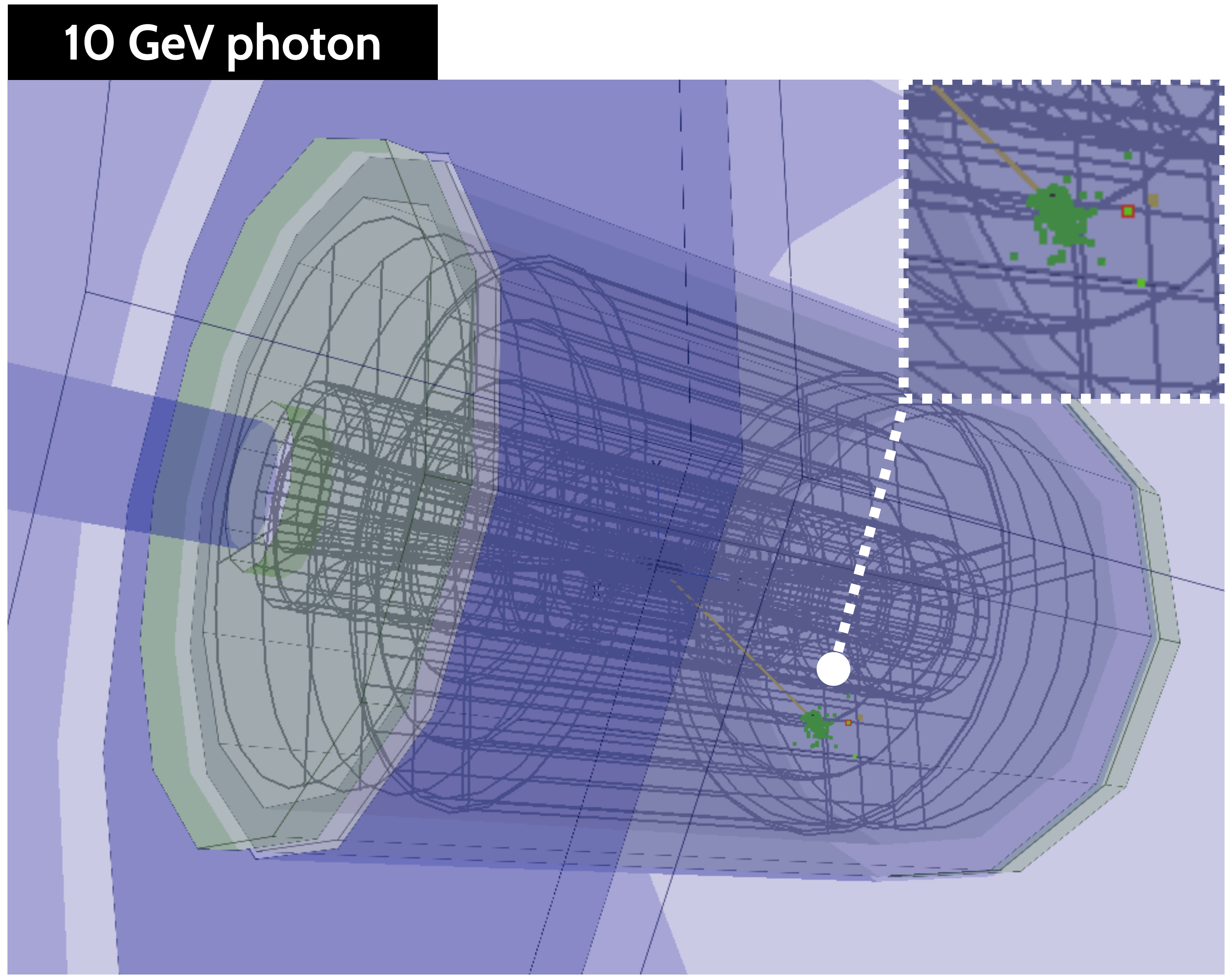
N. Bartosik [a, b]

for the **Muon Collider Physics and Detector Group**

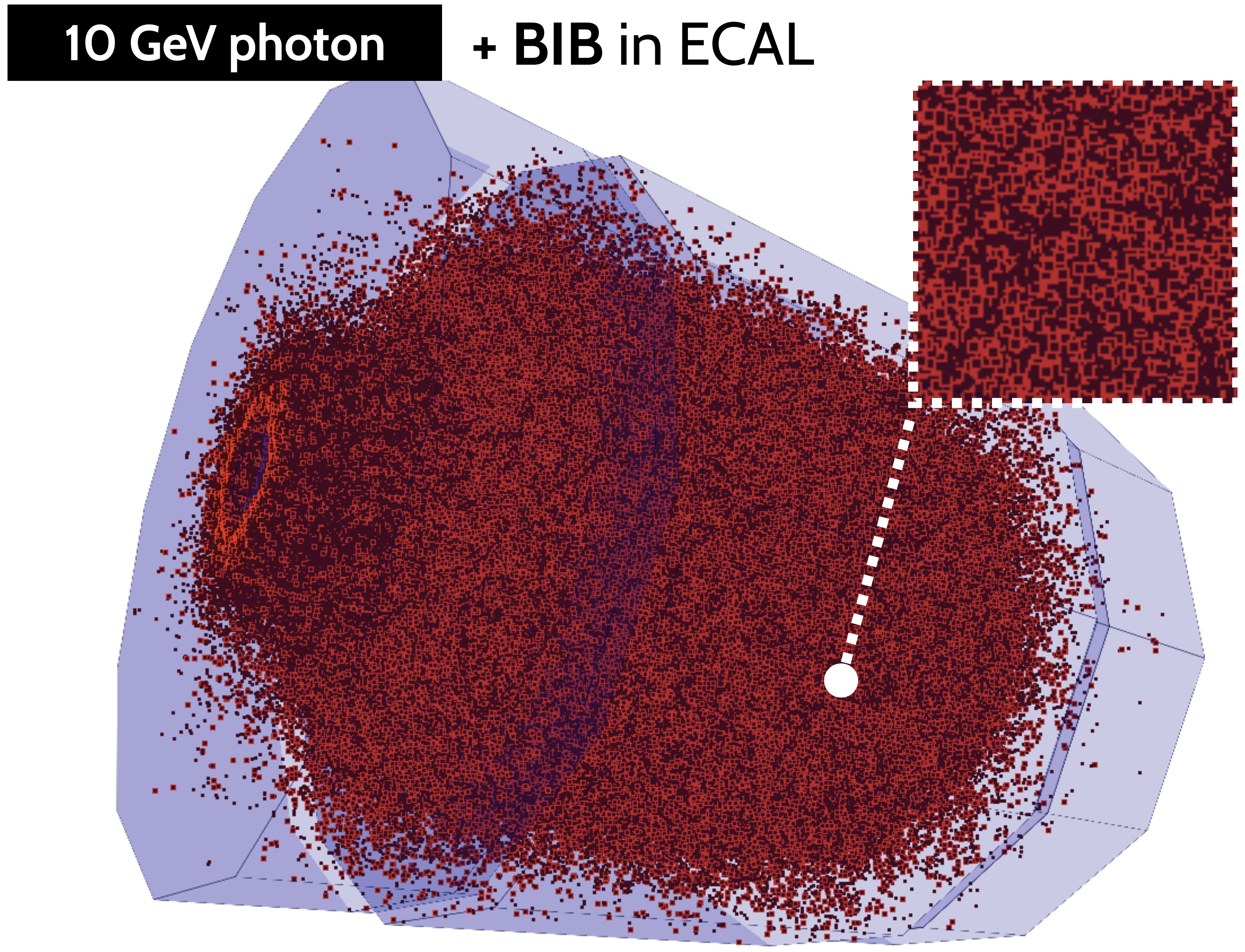
[a] INFN Torino *(Italy)* [b] CERN *(Switzerland)*

We want our simulation studies to be representative of what it will look like in the actual experiment

↳ all BIB effects have to be included in the most realistic way possible



physics analysis will be done on this kind of events



Our event-reconstruction algorithms must be optimised for reconstructing particles from the hard interaction hidden among the sea of BIB particles

Source of BIB

Assuming the beam density of 2×10¹² muons/bunch → large number of decays in the collider ring
e.g. for $\sqrt{s}$ = 1.5 TeV: 4.1×10^5 decays per meter of lattice

Secondary/tertiary particles interact with the accelerator lattice → Beam Induced Background (BIB)

- depends on the beam parameters *(energy, size, rate)*
- depends on the accelerator layout

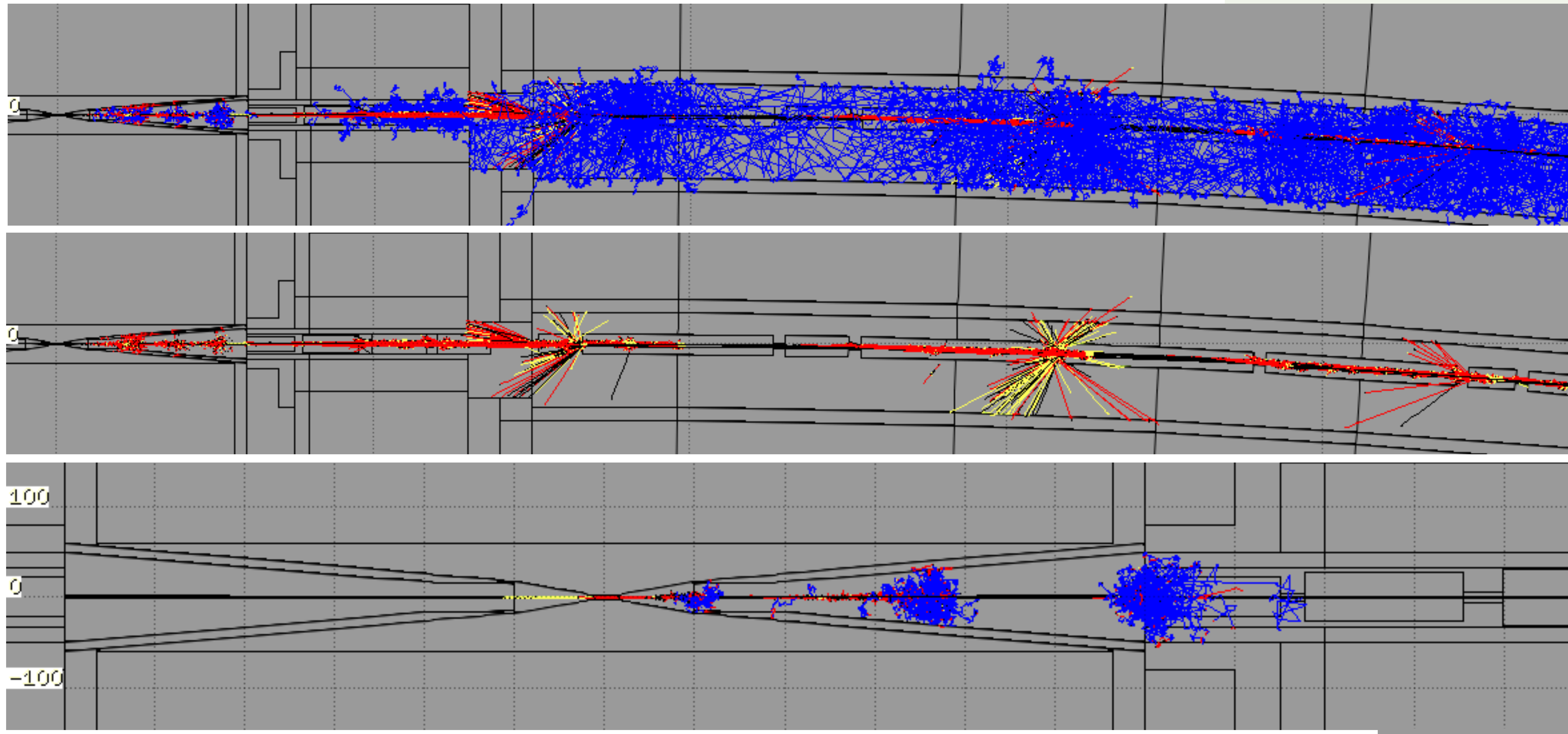
Machine-Detector Interface (MDI) is critical

- must be optimised for a specific collider design
- must be consistently included in the simulation

Current MDI designed for the $\sqrt{s}$ = 1.5 TeV Muon Collider by the Muon Accelerator Program (MAP) ▶

- tungsten nozzles with BCH cladding
- 10° opening angle *(defining the forward acceptance)*

arXiv:2203.07964



FLUKA simulation

Decays of the muons + interactions with the detector lattice and MDI simulated in FLUKA
↳ produces a list of stable particles at the MDI surface → valid only for a specific accelerator lattice + MDI + beam

BIB properties

BIB has several characteristic features to be exploited in the detector design

1. Predominantly very soft particles (~10 MeV) except for neutrons

fairly uniform spatial distribution → no isolated signal-like energy deposits

↳ conceptually different from pile-up contributions at the LHC

2. Significant spread in time (few ns + long tails up to a few μs)

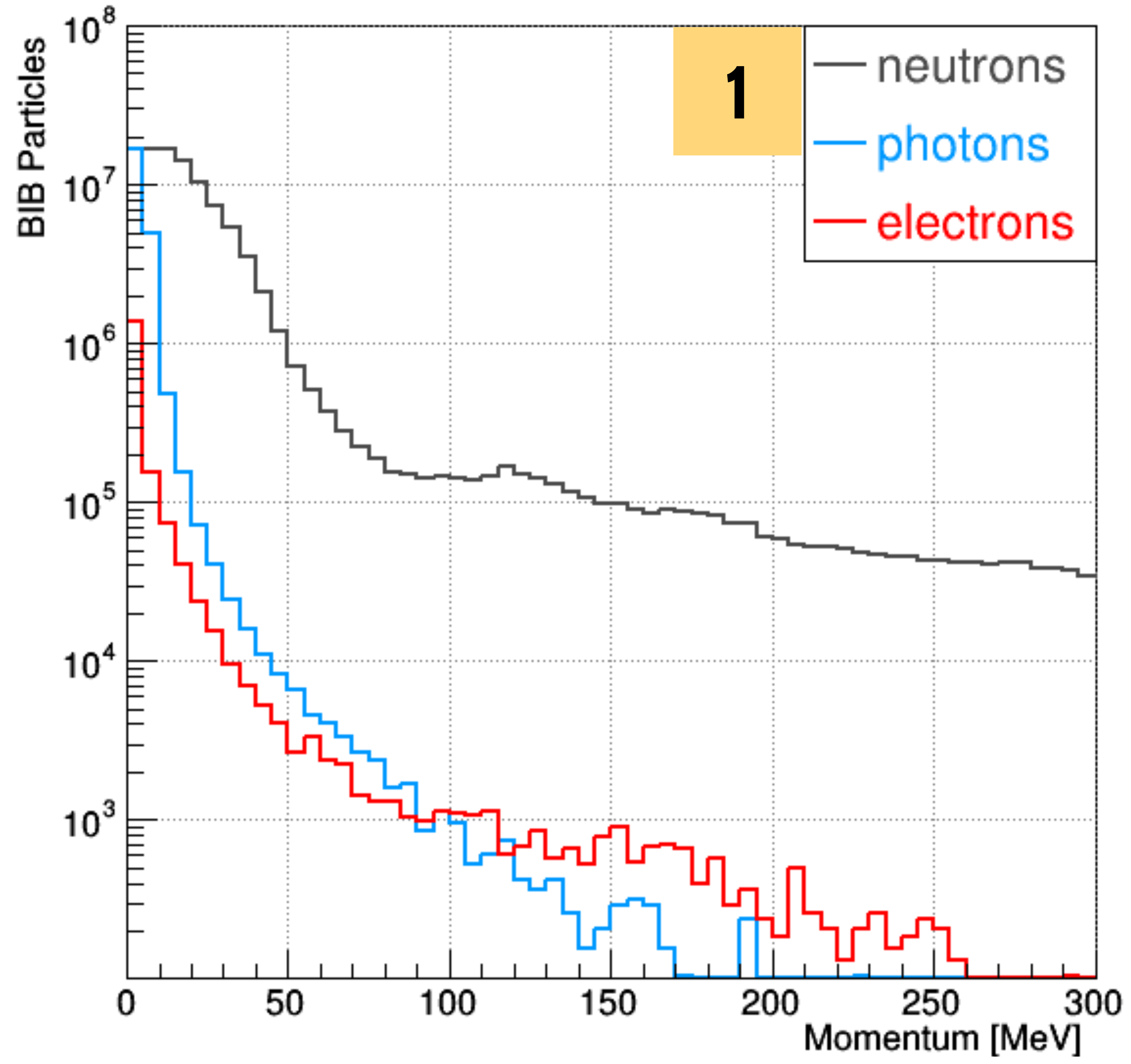
$\mu^+\mu^-$ collision time spread: 30ps at $\sqrt{s}$ = 1.5 TeV | ≤20ps at $\sqrt{s}$ = 3 TeV

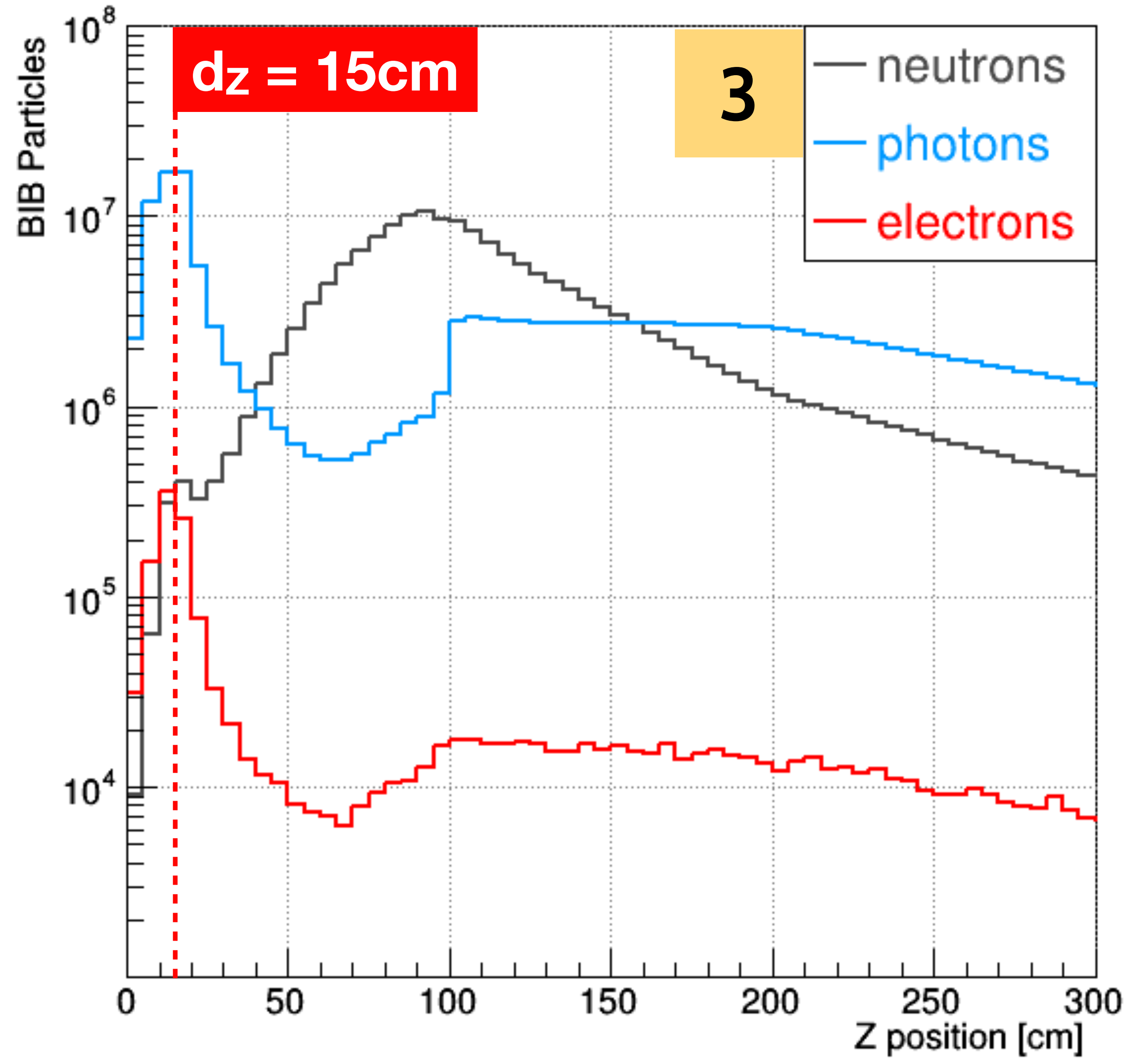
↳ strong handle on the BIB → requires state-of-the-art timing detectors

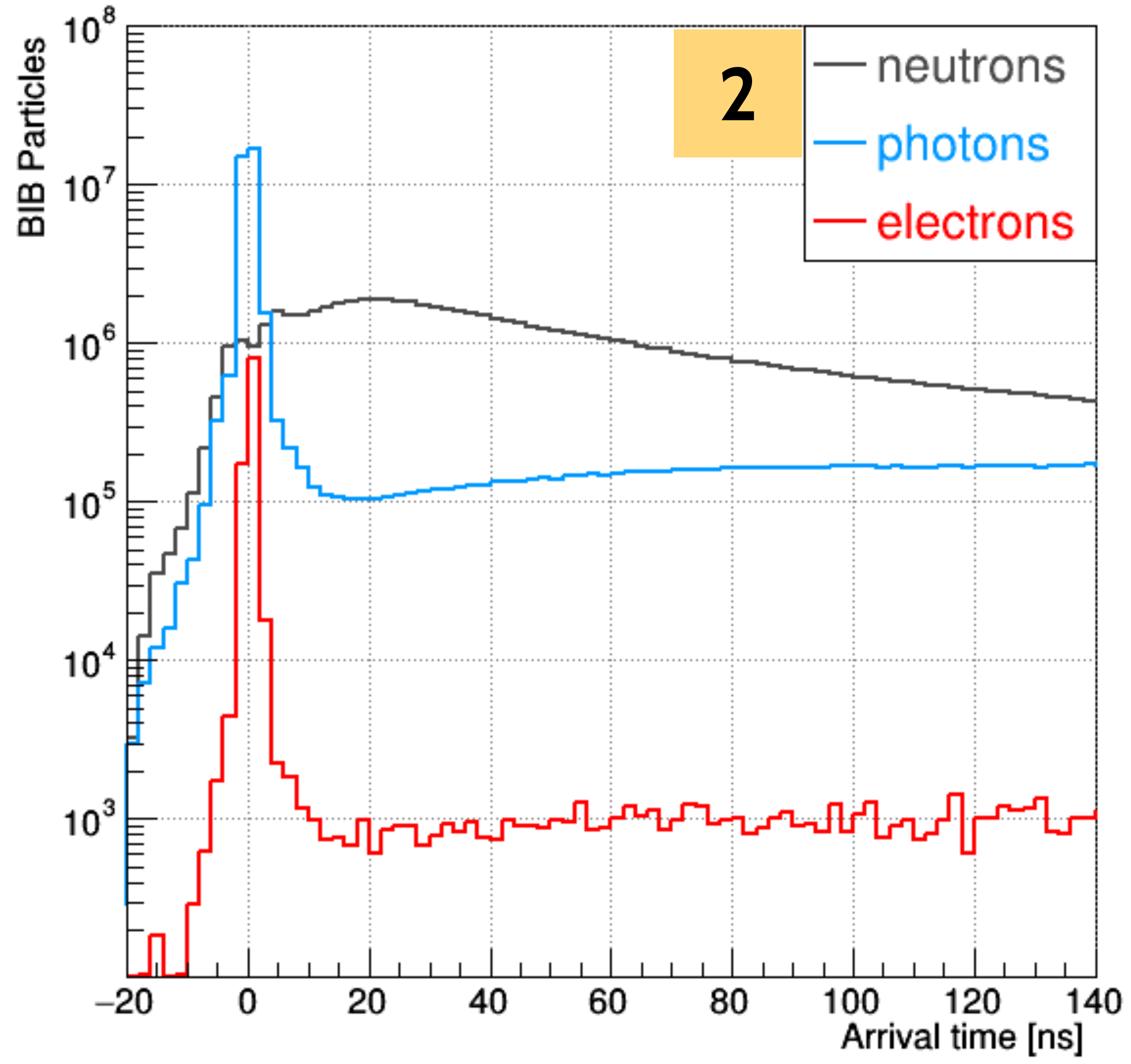
3. Strongly displaced origin along the beam

crossing detector surface at a shallow angle

↳ affects charge distribution + time of flight







To validate our BIB-rejection methods we have to include all these particles in the signal event

Hit classes

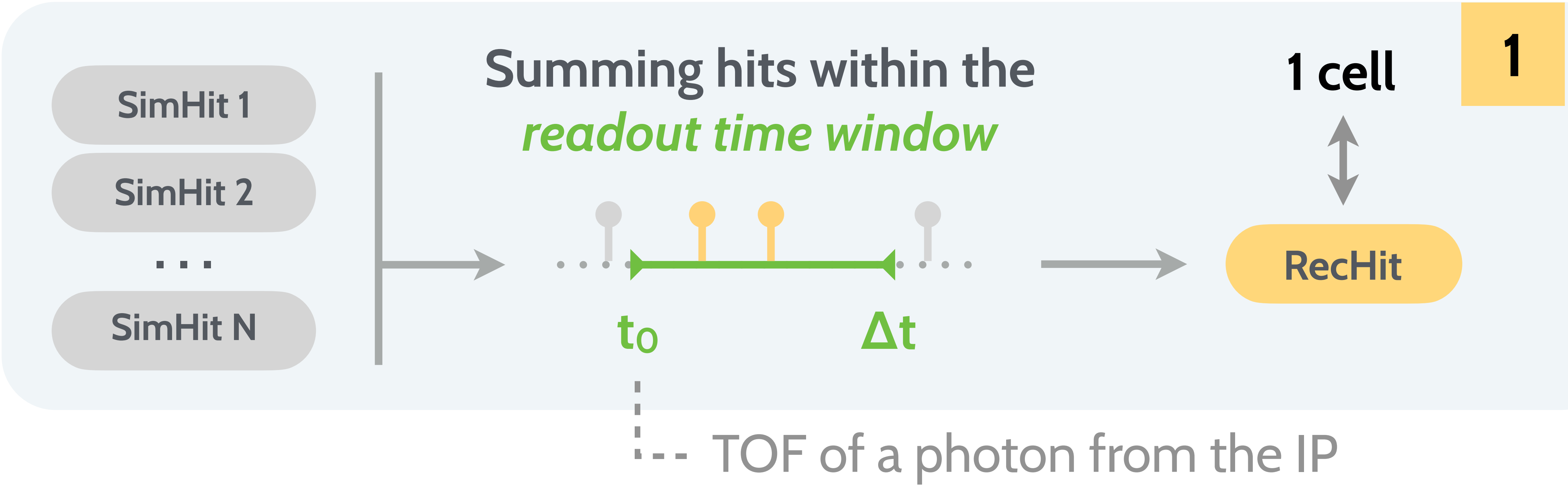
We use two conceptually different LCIO classes to represent digitised detector hits

↳ **CalorimeterHit** (ECAL, HCAL, Muon detector) + **TrackerHit** (Vertex + Tracking detector)

1. Calorimeter hits: cell ID + E_{dep} + timestamp

large cells (0.5×0.5 - 3×3 cm) → manageable # of cells

↳ energy deposits merged within a fixed readout time window



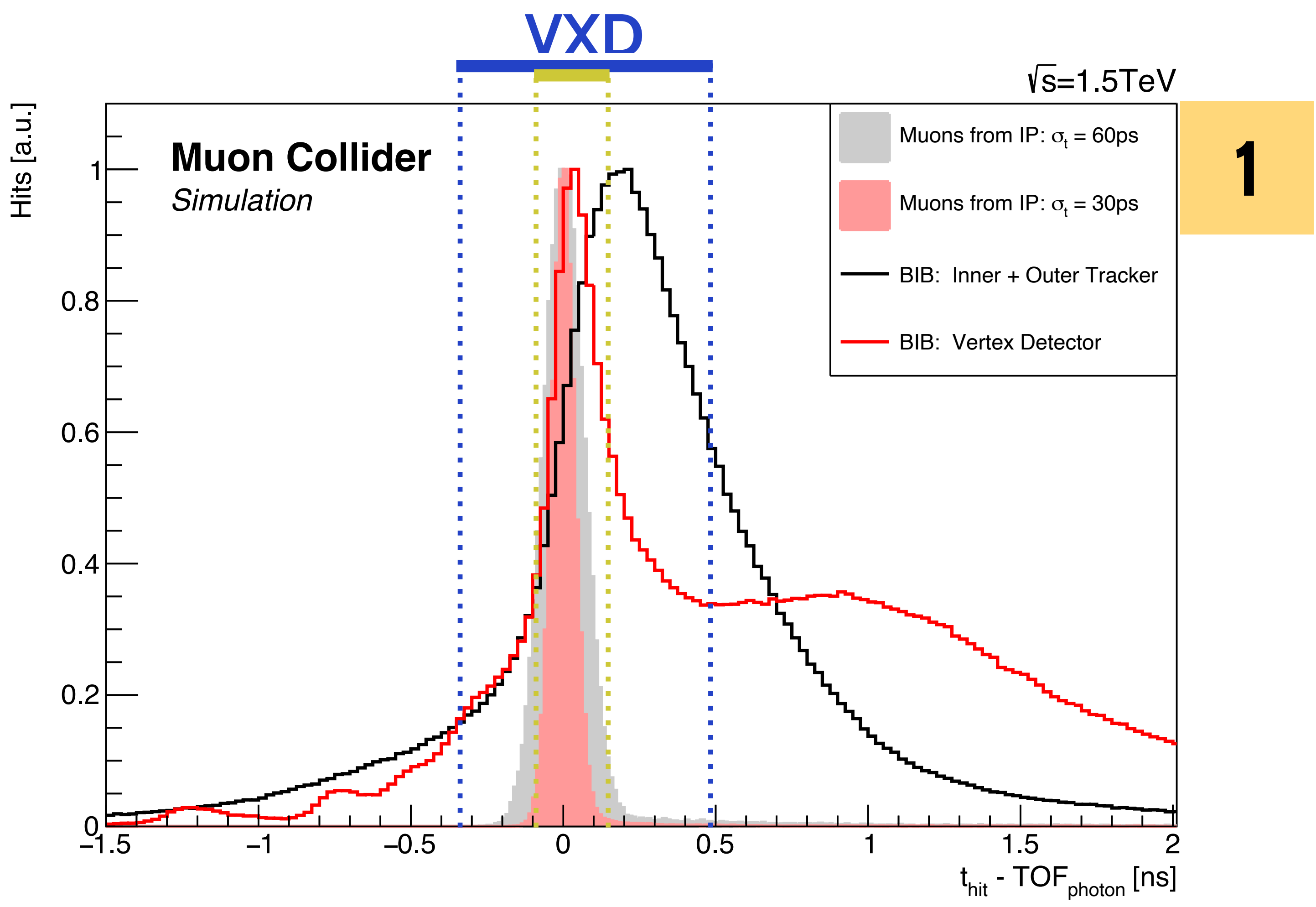
2. TrackerHits: sensor ID + 2D position + time *and more*

small pixels (50×50 μm) to macro-pixels (0.05×10 mm)

↳ only hits within a fixed readout window are kept

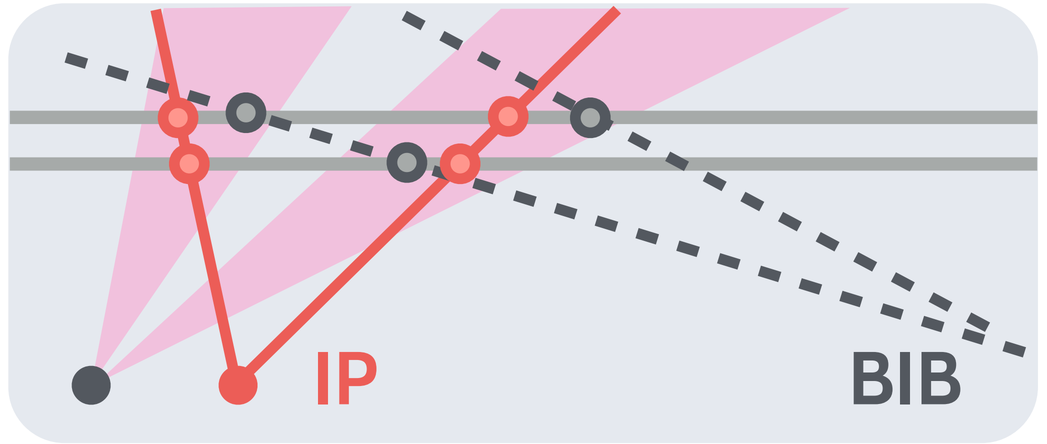
Time-based hit selection allows to significantly reduce the complexity of event reconstruction

Reconstruction of tracks suffers from large combinatorial background
↳ **need suppression of BIB hits + efficient tracking strategies/algorithms**



1. Selection of hits in the narrow time window tailored to the sensor position
↳ limited by the time resolution + beamspot time spread + slow-particle TOF

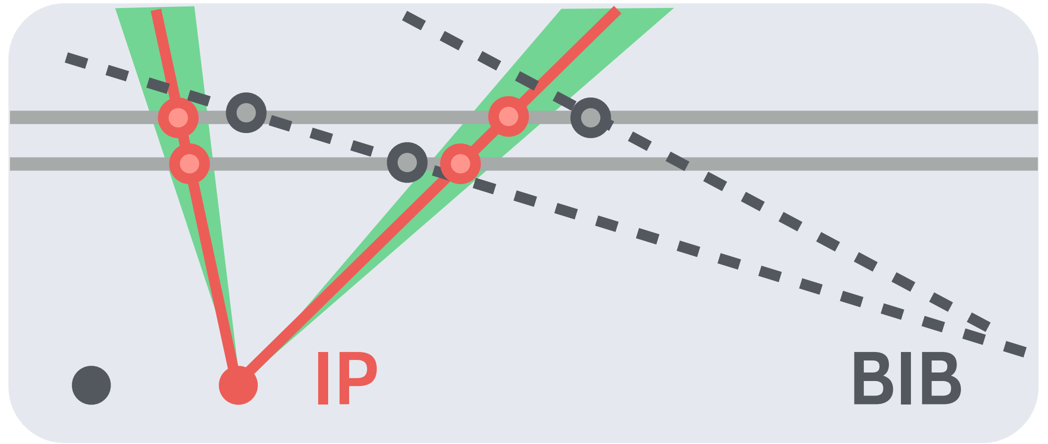
2. Selection of hit doublets aligned with the IP (double layers in the Vertex Detector)
↳ limited by the IP position resolution + actually displaced vertices



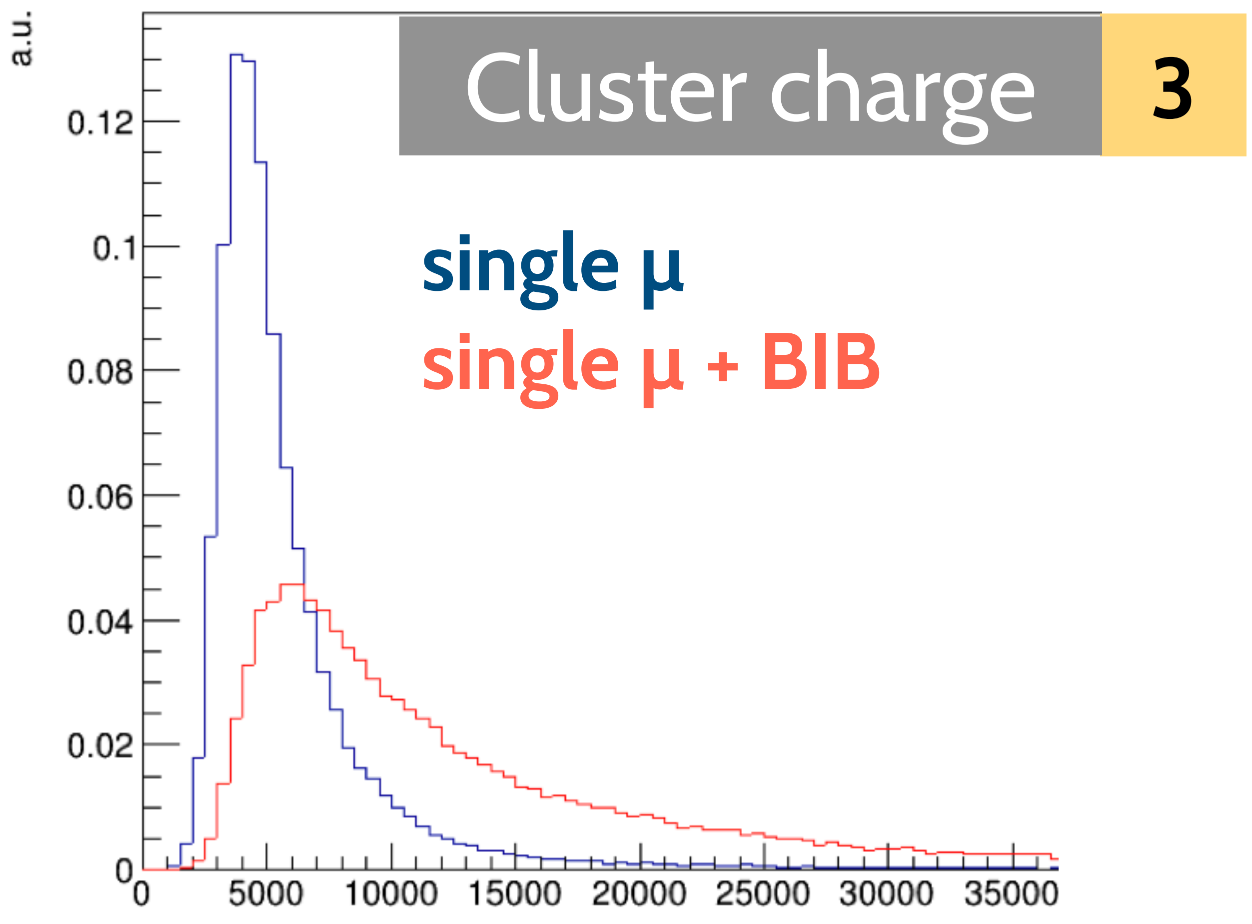
→

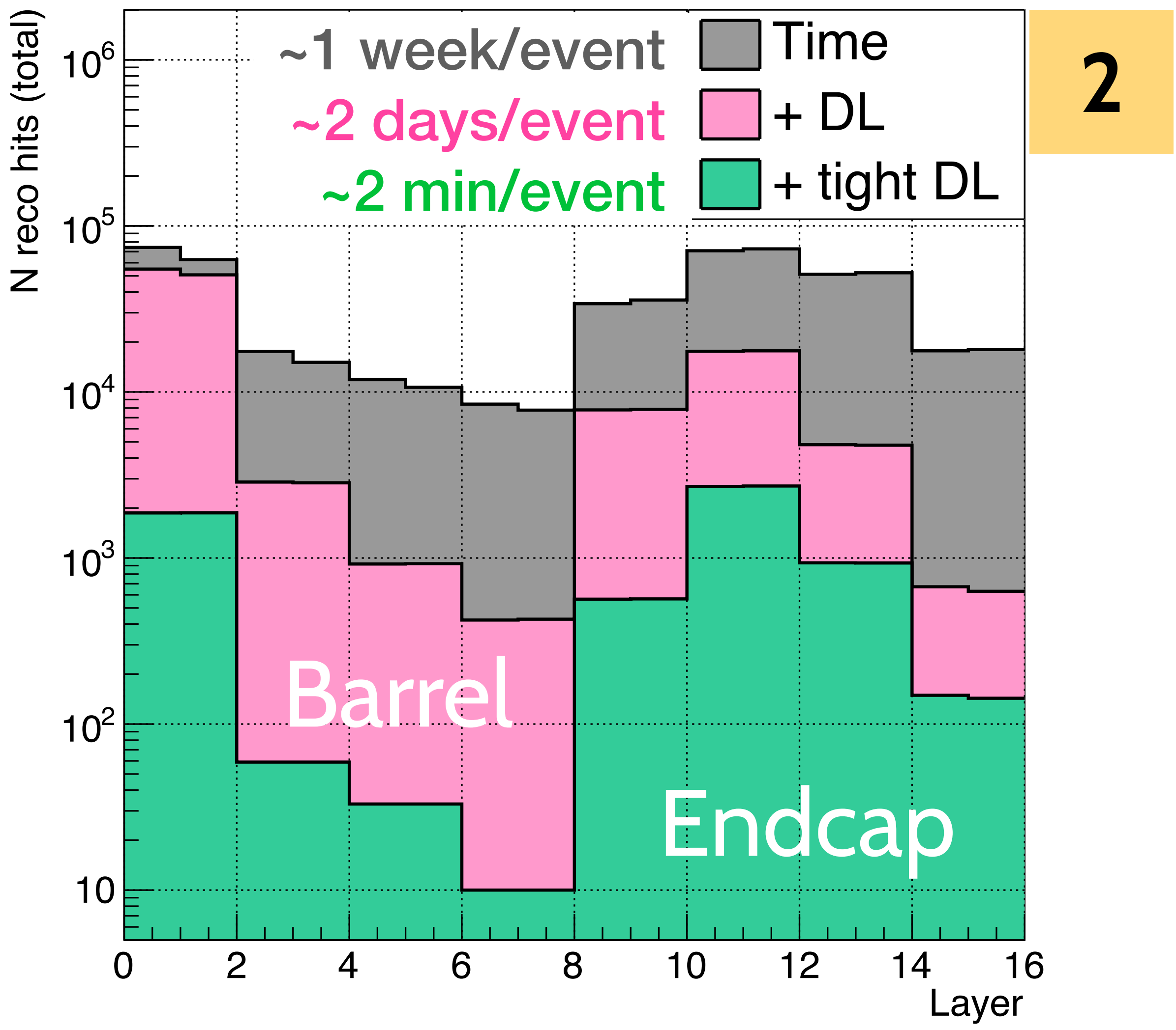
Determine IP position
with faster track reconstruction
- only central region
- using ROI

→



3. Cluster-based BIB suppression (shape + charge)
sensitivity to the particle direction in a single layer
↳ requires realistic digitisation





BIB must be included before digitisation to properly take detector effects into account

BIB overlay

Geant4 simulation is very expensive → passing 10^8 BIB particles through Geant4 in each event is not feasible

We run Geant4 simulation of a large number of BIB events in advance

↳ e.g. 1000 BXs stored in `/eos/experiment/muoncollider/data/BIB/MuCollv1_25ns_nEkin150MeV_QGSPBERT/`

only accessible by authorised users

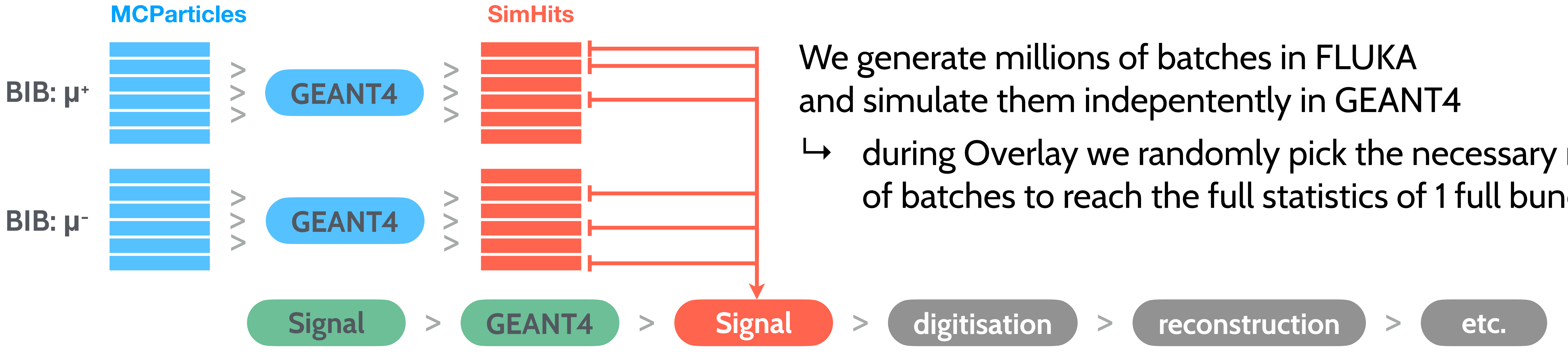
Randomly selected BIB BX added to the signal event → fine for now but not efficient for large-statistics studies

BIB from the whole BX is treated as a single entity: 1K BIB simulations → 1K independent events

work in progress

With the new simulation workflow in FLUKA we will be using a more granular approach

↳ muon decays are simulated in batches: e.g. 200 muons/batch → results written to small files (1 batch/event)



We generate millions of batches in FLUKA and simulate them indepentently in GEANT4

↳ during Overlay we randomly pick the necessary number of batches to reach the full statistics of 1 full bunch crossing

Overlay processor

Merging of BIB with the signal event is done at the SimHit level
to ensure that detector effects are applied to the combined energy deposits, like in real life

Marlin processor OverlayTimingGeneric is used for BIB overlay
↳ configured in: `mucoll-benchmarks/digitisation/marlin/subconfigs/Overlay.xml`
defined as a group of processors called **Overlay** with several different instances inside

1 event/file → 1 full BX

```
<processor name="OverlayFull" type="OverlayTimingGeneric">
  <parameter name="BackgroundFileNames" type="StringVec">
    /eos/experiment/muoncollider/data/BIB/MuCollv1_25ns_nEkin150MeV_QGSPBERT/sim_mumu-1e3x500-26m-lowth-excl_seed0000_allHits.slcio
    ...
    /eos/experiment/muoncollider/data/BIB/MuCollv1_25ns_nEkin150MeV_QGSPBERT/sim_mumu-1e3x500-26m-lowth-excl_seed0005_allHits.slcio
  </parameter>
  <parameter name="NumberBackground" type="float" value="1" />
  <parameter name="StartBackgroundFileIndex" type="int" value="-1"/>
</processor>
```

10 events/file → ~1% of BX

lightweight version for testing purposes

```
<processor name="OverlayTest" type="OverlayTimingGeneric">
  <parameter name="BackgroundFileNames" type="StringVec">
    /cvmfs/muoncollider.cern.ch/datasets/bib/MuColl_v1/example/mumi.000.slcio
    ...
    /cvmfs/muoncollider.cern.ch/datasets/bib/MuColl_v1/example/mupl.004.slcio
  </parameter>
  <parameter name="NumberBackground" type="float" value="100" />
</processor>
```

Only input files and numbers of events are different. Other parameters taken from the group.

Overlay group parameters

Parameters defined in a group are shared between all processors in the group
unless overwritten in the processor itself

The most relevant configuration parameter

```
<parameter name="Collection_IntegrationTimes" type="StringVec">
 VertexBarrelCollection -0.18 0.24
 VertexEndcapCollection -0.18 0.24

 InnerTrackerBarrelCollection -0.36 0.48
 InnerTrackerEndcapCollection -0.36 0.48

 OuterTrackerBarrelCollection -0.36 0.48
 OuterTrackerEndcapCollection -0.36 0.48

 ECalBarrelCollection -0.25 10.
 ECalEndcapCollection -0.25 10.
 ECalPlugCollection   -0.25 10.

 HCalBarrelCollection -0.25 10.
 HCalEndcapCollection -0.25 10.
 HCalRingCollection   -0.25 10.

 YokeBarrelCollection -0.25 10.
 YokeEndcapCollection -0.25 10.

</parameter>
```

◀ list of input collections to be added to the event

with corresponding time-selection cuts [ns]:
<collection name> <time min> <time max>

Time-selection cuts are slightly wider than readout times used in the corresponding detector digitisers

↳ reduces CPU and RAM wasted on overlaying useless SimHits, while keeping all the relevant ones

added extra $3\sigma_t = 3 \times \mathbf{0.03} = \underline{0.09}$ ns

$-0.09 \rightarrow -0.09 - \underline{0.09} = -0.18$

$0.15 \rightarrow 0.15 + \underline{0.09} = -0.24$

```
<processor name="VXDBarrelDigitiser" type="DDPlanarDigiProcessor">
  <!--resolution in time-->
  <parameter name="ResolutionT" type="FloatVec"> 0.03  </parameter>
  <!--Lower bound of the time window [ns]-->
  <parameter name="TimeWindowMin" type="float"> -0.09 </parameter>
  <!--Upper bound of the time window [ns]-->
  <parameter name="TimeWindowMax" type="float"> 0.15 </parameter>
```

Overlay configuration

Sequence of `<if> ... </if>` conditions define which Overlay instance to run
depending on the value of the `Config.Overlay` parameter

```
<!-- ========== Overlay.xml  ========== -->
<if condition="Config.OverlayNone">
  <processor name="OverlayNone"/>
</if>
<if condition="Config.OverlayTest">
  <processor name="OverlayTest"/>
</if>
<if condition="Config.OverlayTrimmed">
  <processor name="OverlayTrimmed"/>
</if>
<if condition="Config.OverlayFull">
  <processor name="OverlayFull"/>
</if>
```

```
<!-- Defining other processors directly in the steering file -->
<processor name="Config" type="CLICRecoConfig" >
  <parameter name="Verbosity" options="DEBUG0-9,MESSAGE0-9,WARNING0-9,ERROR0-9,SILENT"> MESSAGE  </parameter>
  <!--Which option to use for Overlay: False, BIB. Then use, e.g., Config.OverlayFalse in the condition-->
  <parameter name="Overlay" type="string">False</parameter>
  <!--Possible values and conditions for option Overlay-->
  <parameter name="OverlayChoices" type="StringVec">False None Test Trimmed Full</parameter>
```

By default `Config.Overlay` is `False` → no Overlay processor included in the <execute> section

↳ now we'll rerun the digitisation step with `Config.Overlay` set to `Test`

Hands-on

Instructions for the hands-on part documented in the Wiki:

https://mcdwiki.docs.cern.ch/tutorials/cern2023/reconstruction/bib/

ATTENTION!

Before proceeding - update your mucoll-benchmarks repository with the latest changes

```
# Login to LXPLUS with AlmaLinux9
ssh -XY ${USER}@lxplus9.cern.ch
# Go to the folder of the reference repository
cd ~/work/mucoll-tutorial-2023/mucoll-benchmarks
# Pull the latest changes from GitHub repository
git pull
# Go back to the main working directory
cd ~/work/mucoll-tutorial-2023/
```