TimeSPOT WP4:
Stato delle attività a Milano

Marco Petruzzo

23 Giugno 2020

Current status

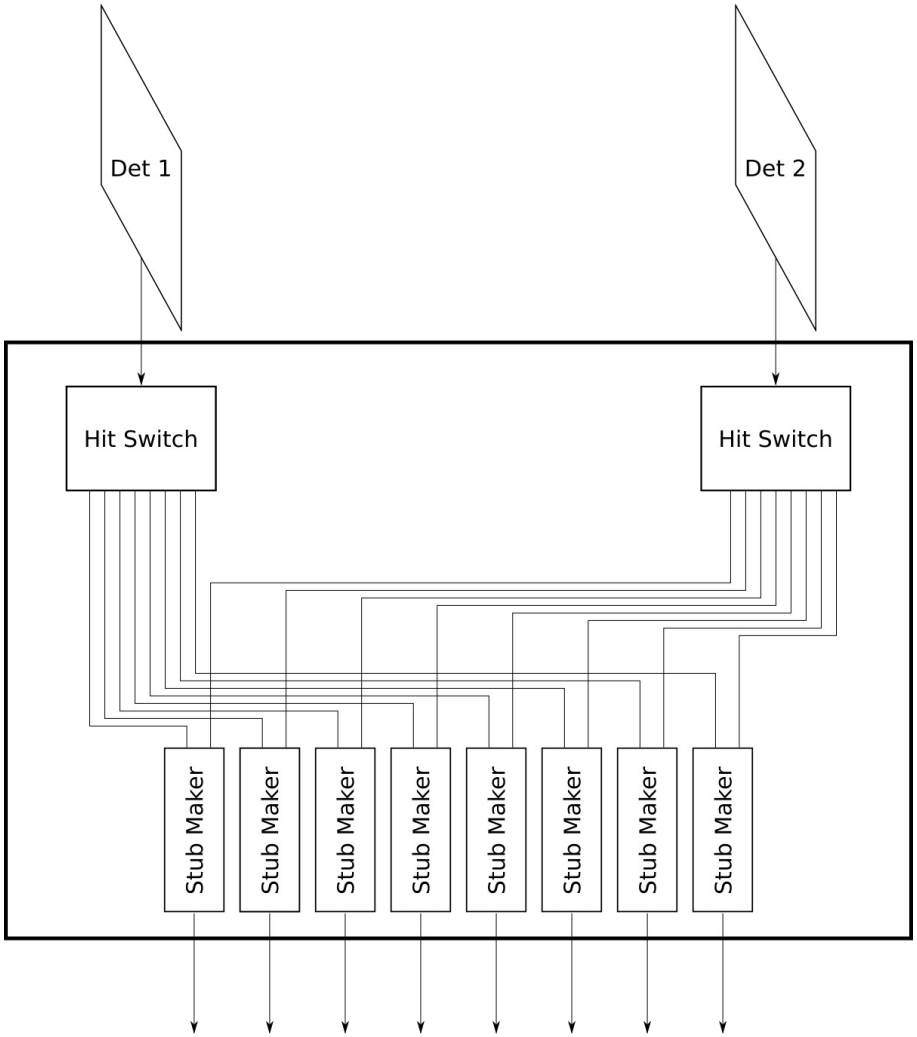
- **High level simulations** of the algorithm performed
  - a VELO-like detector in LHCb Upgrade II conditions
  - **30 ps timing** resolution

- **Device architecture** has been defined and **implemented**
  - Switch + Engine tested in hardware on the **gFEX board**
  - Stub constructor implementation ongoing on **Xilinx VC709 Evaluation board**

- **Data generation on FPGA** to test the device at high rate:
  - Implementation of the communication between the two boards (VC709/gFEX) to be fixed. Optical links works in loop-back configurations on both the boards.

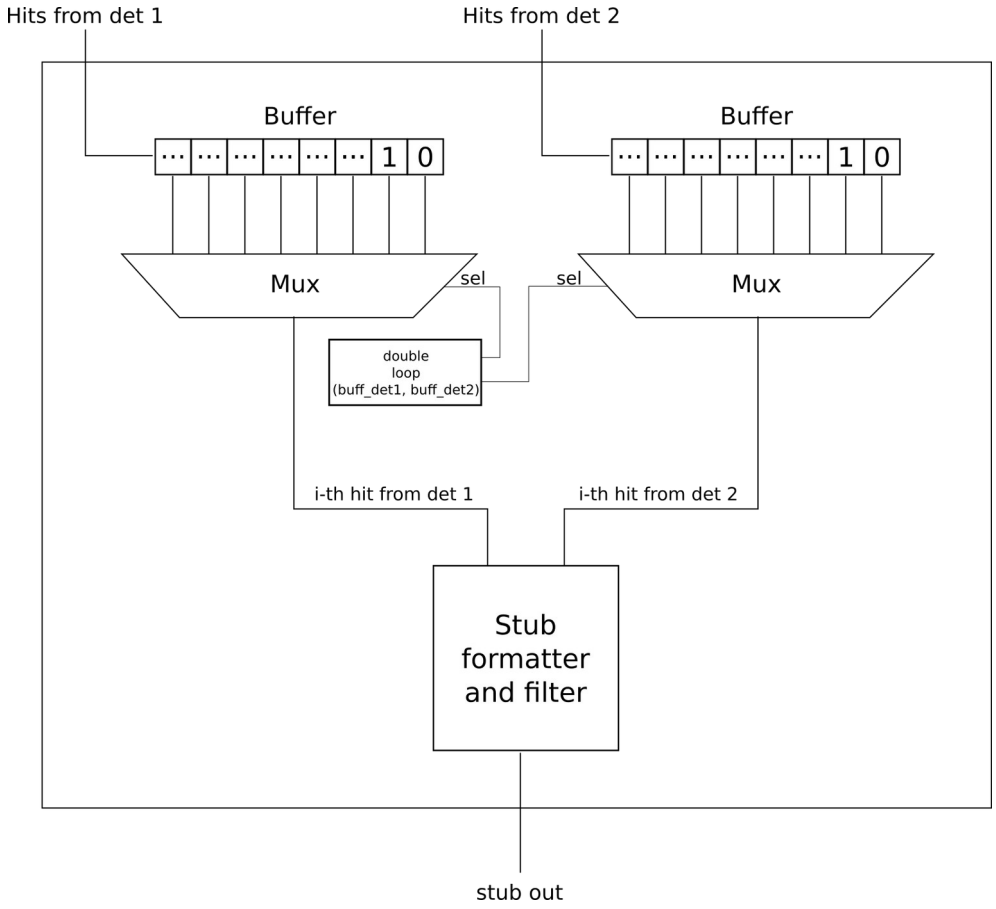
Stub constructor

- The **Stub Constructor** receives the detector hits from a couple of sensor planes
  - Two Switches deliver the hits to a pool of Stub Makers
- Each Stub Maker receives **data from exclusive sensor regions**, uniformly distributed w.r.t. to phi, NOT uniformly distributed w.r.t to radial coordinate
- **The (r,phi) coordinates** of each hit need to be evaluated before the Hit Switches.
- The Hit Switch delivers the hits to the proper/expected Stub Makers
- **The Stub Maker processes every combination** of hits that are received from the region it is associated with



Stub constructor status

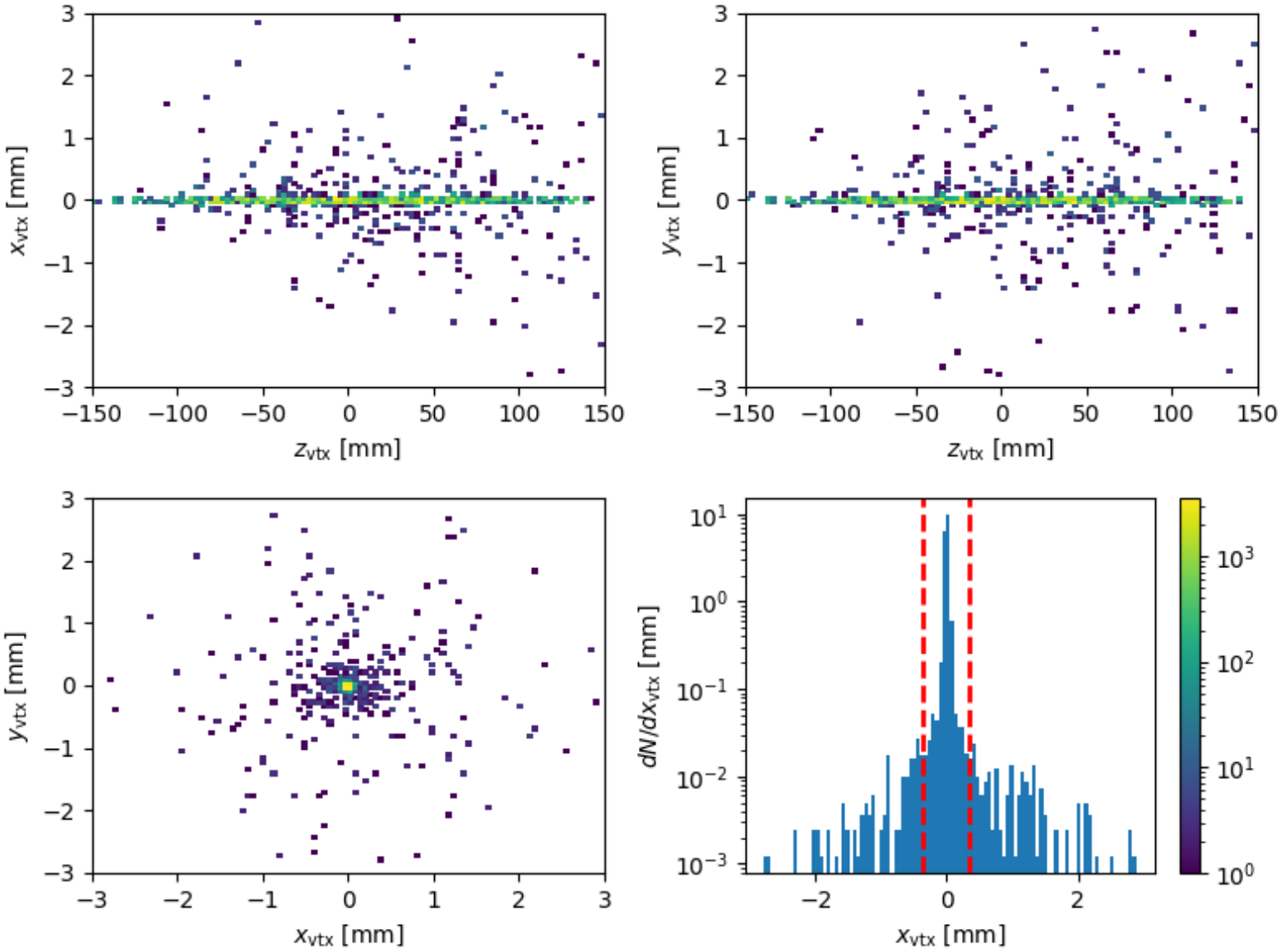
- **Hit Switch:**
  - Identical to the Stub Switch, **already tested** on the previous implementation of the Switch+Engines architecture (on gFEX board)
- **Xy-to-rphi** converter:
  - Tested on the VC709 board with 200 MHz ref. clock
- **Stub Maker:**
  - **Combinatorial logic tested** on the VC709 with 200 MHz ref. Clock
  - Stub Formatter and Filter to be finalized
- **Full Stub Constructor:**
  - An array/grid of Stub Makers has to be instantiated and tested with data from physics events



Software simulation

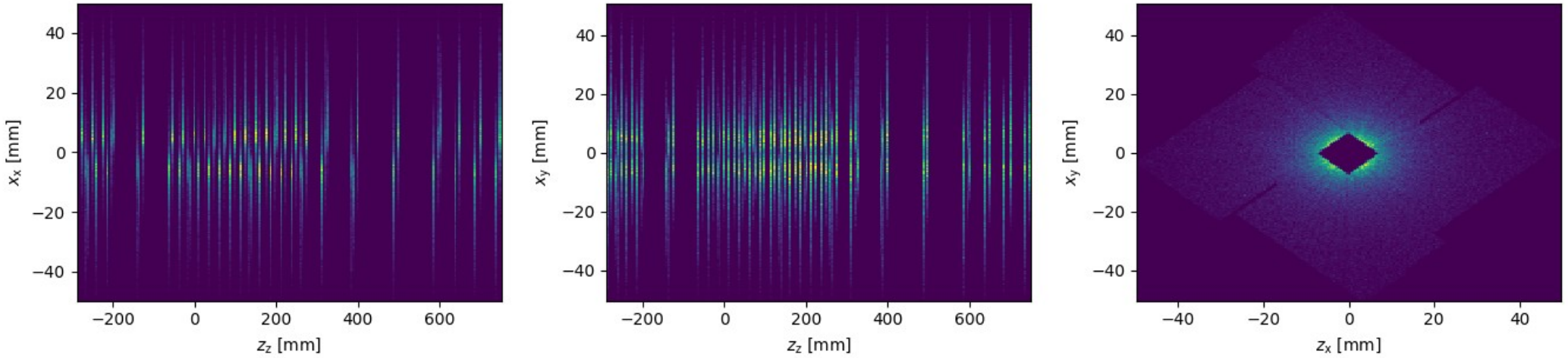
- The **software simulation** is needed to “train”/design properly the hardware system:
  - **Distributions of hits, number of Stub Makers, number of Engines**
  - Validation of the algorithm and **evaluation of the performances**
- In the past I used a fully standalone simulation with:
  - **Simplified generation** of tracks and hits
  - **Simplified** detector **geometry**
  - Implementation of the reconstruction algorithm based on Stubs identification and then Tracks identification as “clusters” of Stubs with similar parameters (performed by the Engines)
- At the moment I am working on a **porting of the simulation to Python**:
  - Mainly based on **Pandas**
  - Particles and detector hits are **imported from a ROOT file**
  - Data from **LHCb Upgrade-I** and **timing not included**
  - Integration of data from Bologna is **easy to implement** (provided that the format is similar)

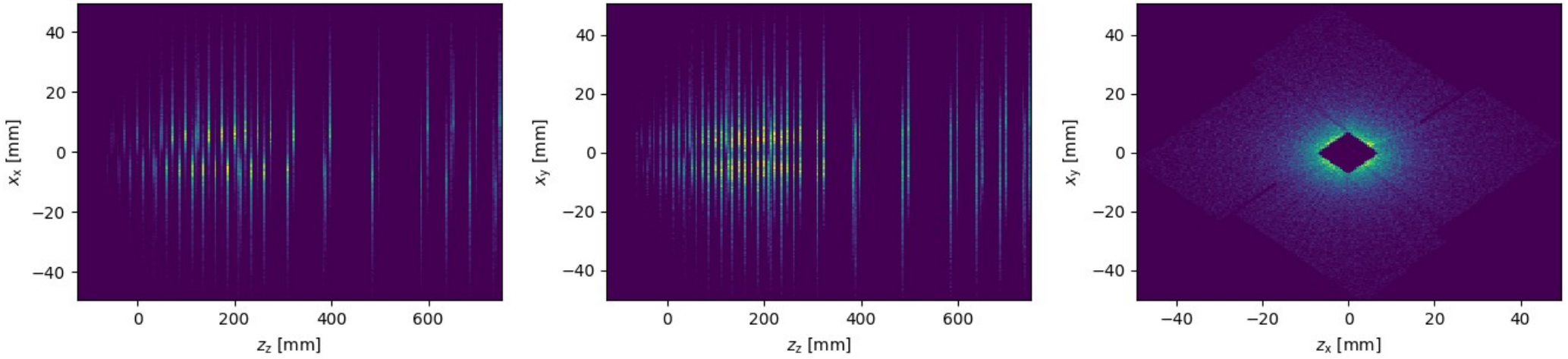
Vertex distribution



Hits distribution

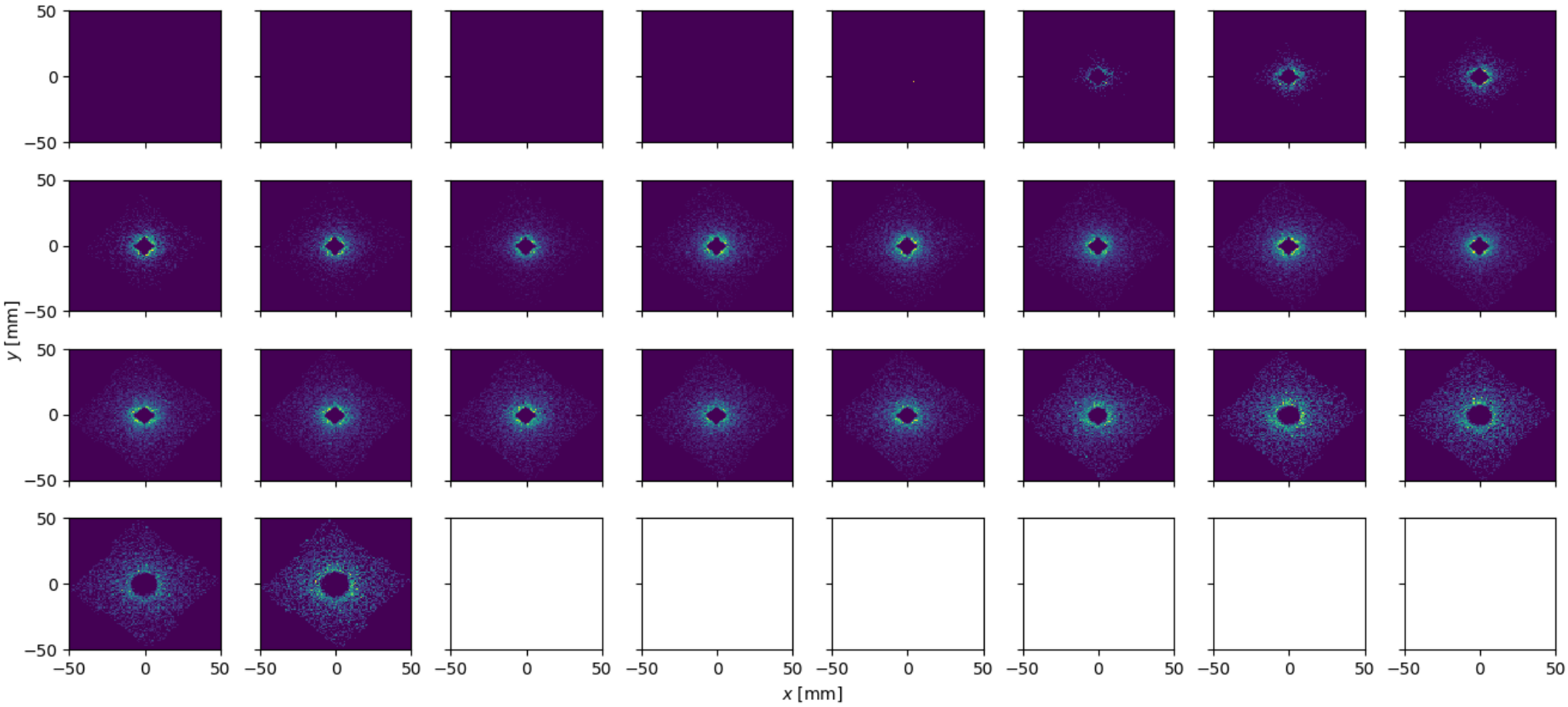
Hits from all the particles





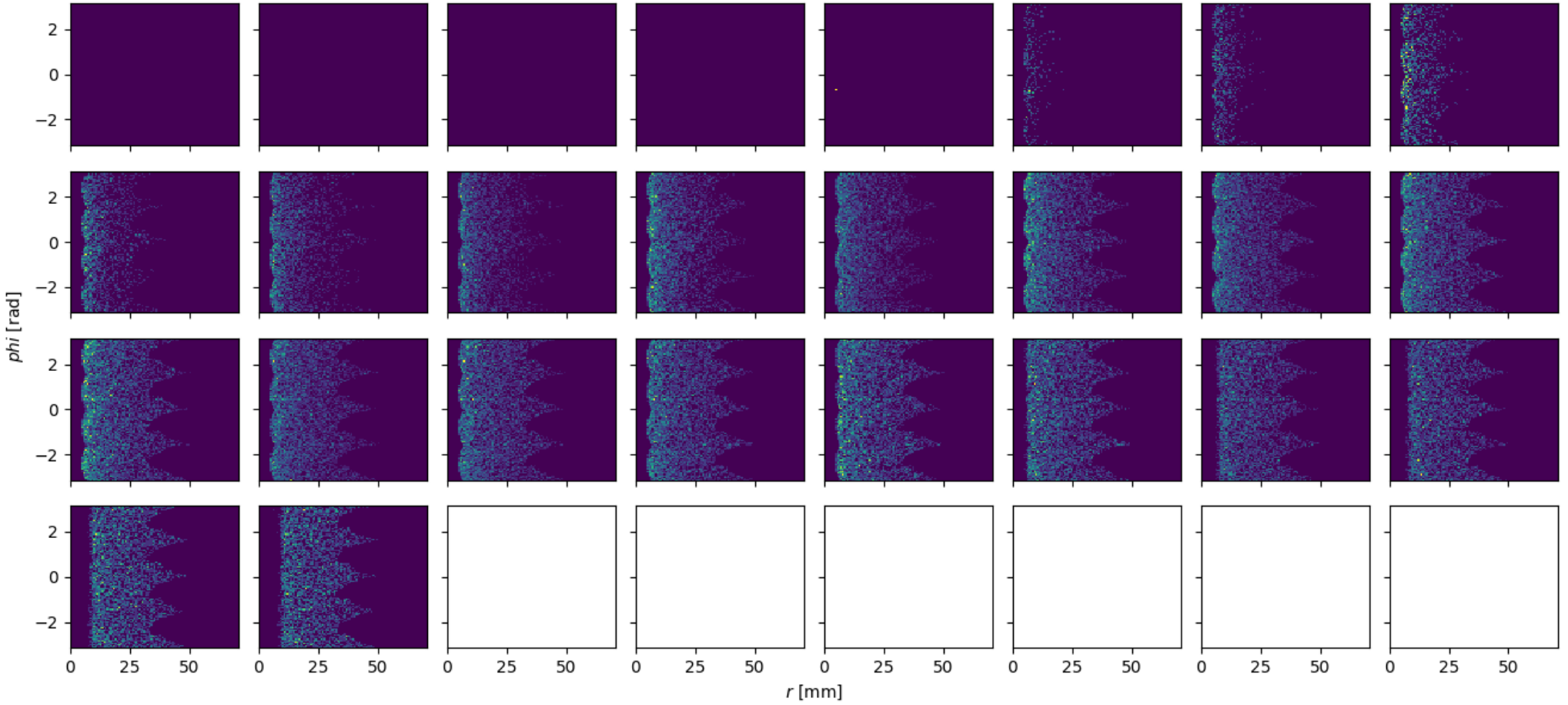
Hits from “trackable” particles

Sensor hits distribution (x-y)



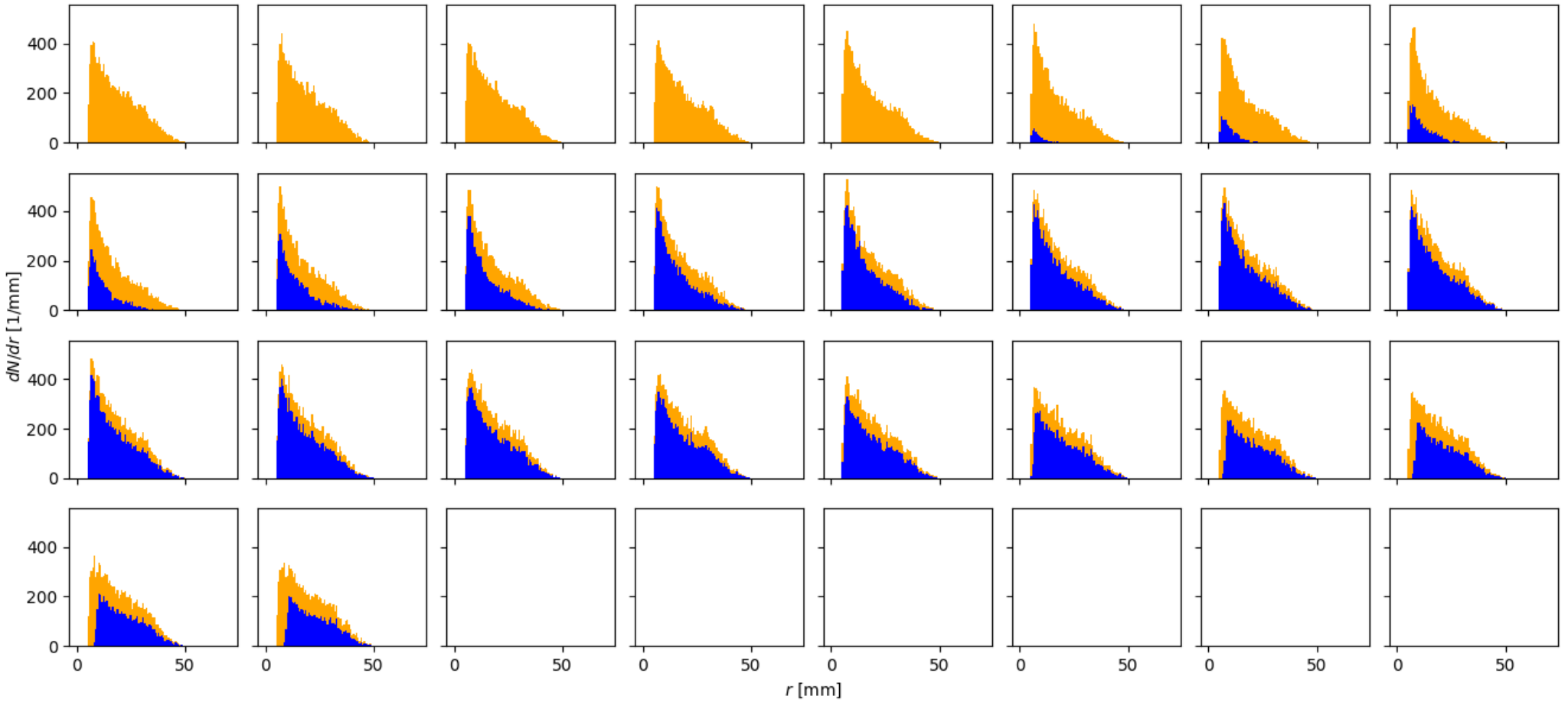
Hits from “trackable” particles

Sensor hits distribution (r-phi)



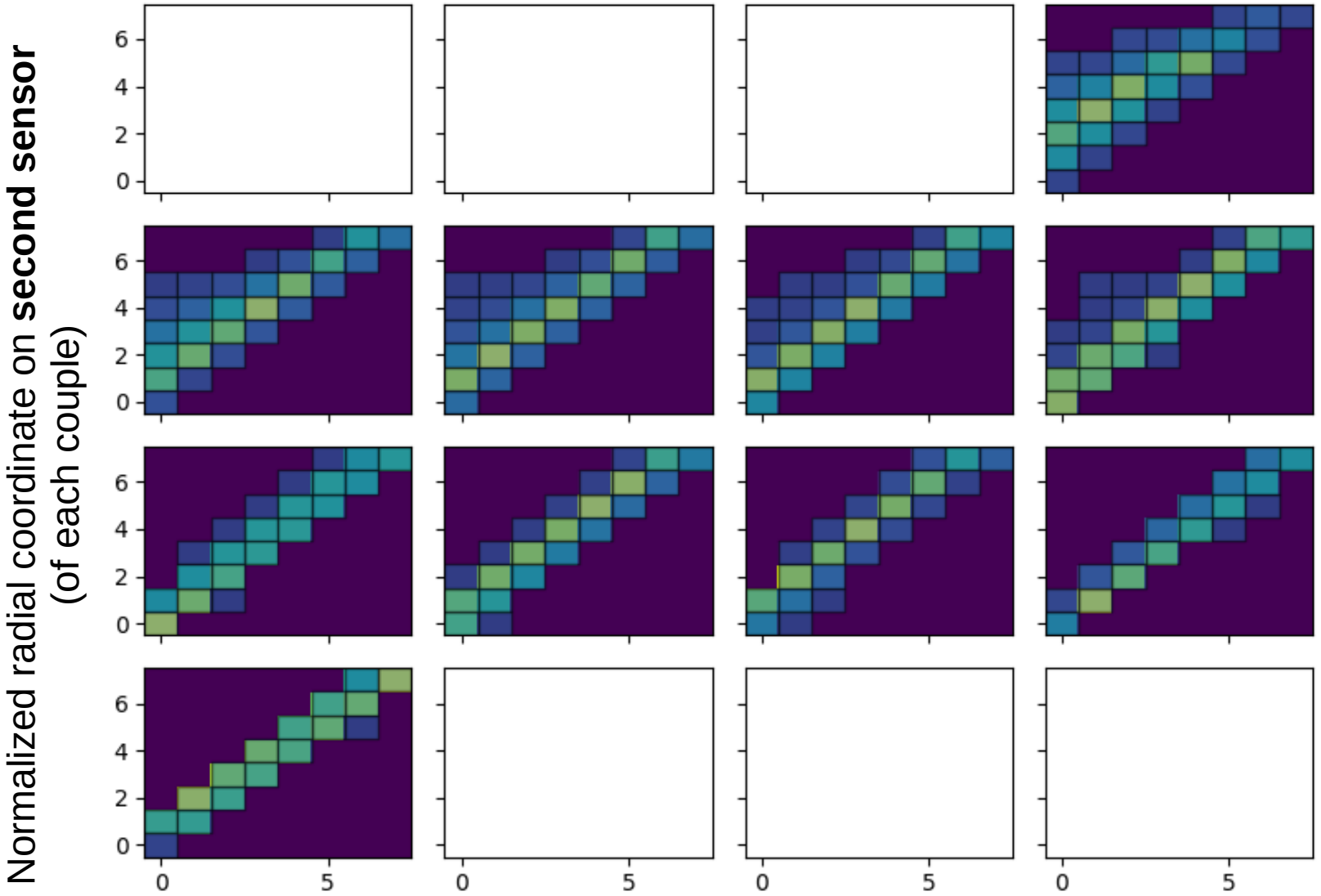
Hits from “trackable” particles

Sensor hits distribution (radial)

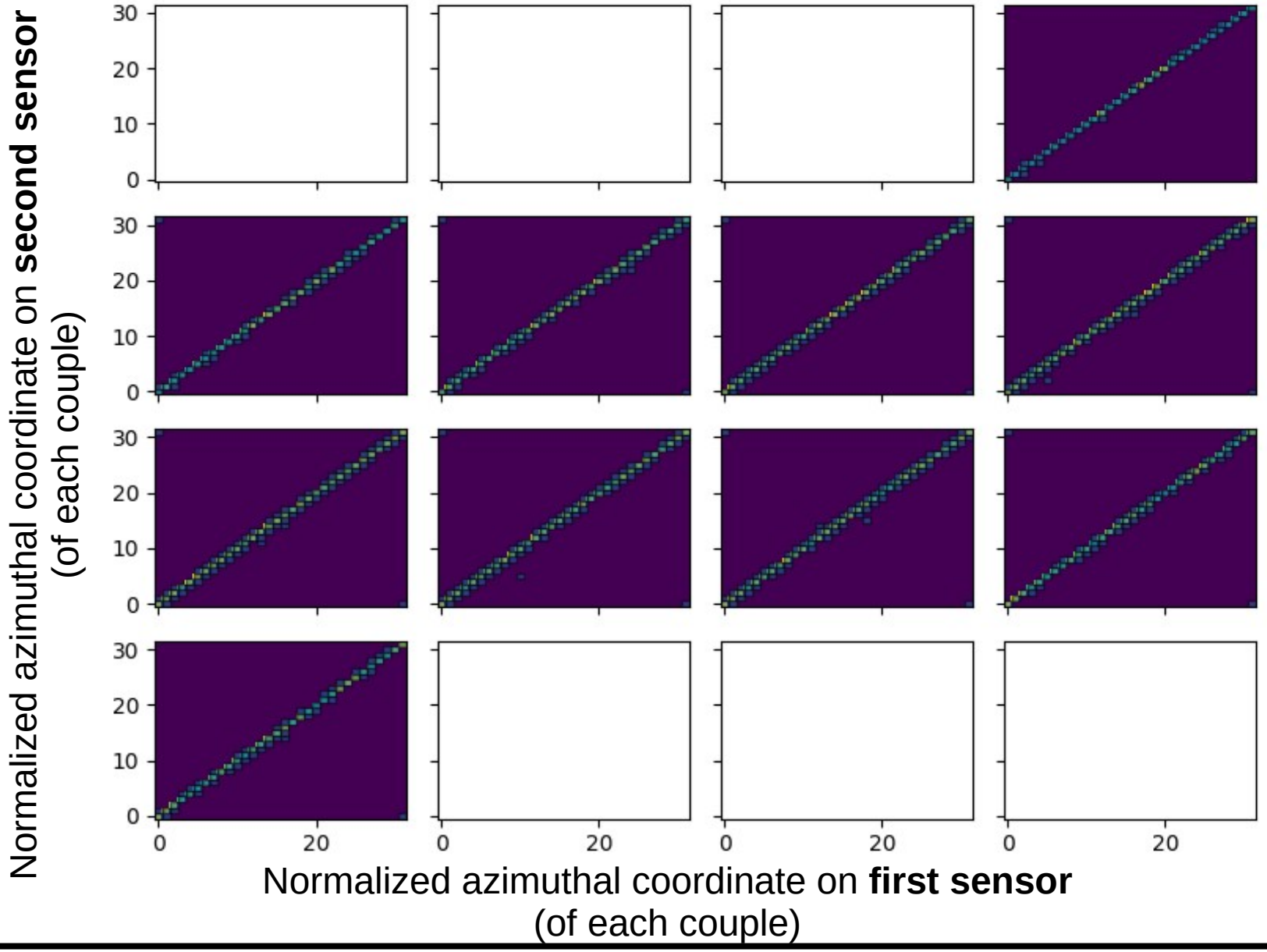


Orange: Hits from all particles
Blue: Hits from “trackable” particles

Hits distribution on sensor couples



Hits distribution on sensor couples



What is new / missing

- What is **new**:
  - Data are read from an externally generated file **from LHCb simulation**
  - Able to **read ROOT files**, ported to Pandas tables using “uproot”

- What is **missing**:
  - **Finalize the porting** of the engine (software) implementation
  - Re-evaluate the **performances** (using the old routines)
  - **Timing**, but it is easy to implement as soon as the information is included in the particles/hits file

- **Why it is needed:**
  - **Better evaluation of the performances**, using proper LHCb data
  - **Evaluation of the hardware resources** distribution (active Stub Makers, active Engines)
  - Data preparation for **high-rate hardware tests**