Workshop summary
CDTWIT 2017

Theme of workshop

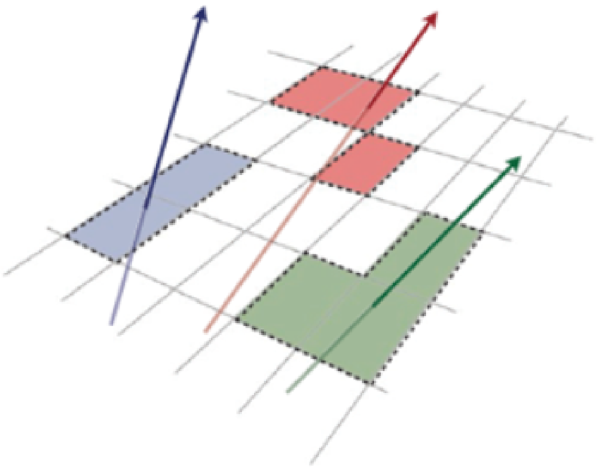
- Sensors, readout and trigger electronics, online and offline reconstruction software
- **Pattern recognition** and **machine learning algorithms** devoted to the **reconstruction of particle tracks** or jets in HEP
  - Hardware developments that enable them

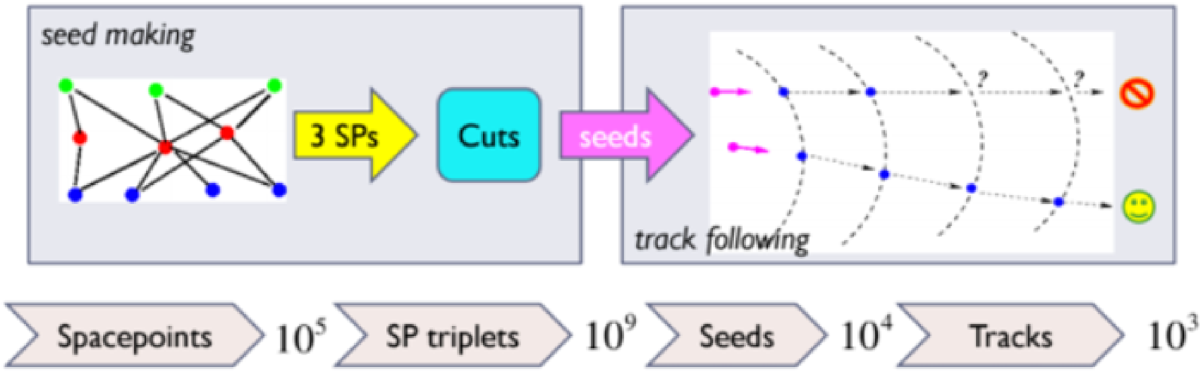
- Focus on HEP: ALICE ITS, ATLAS ITk, some mu3e/LHCb/
- Also: Space, medical (MRI vis.),

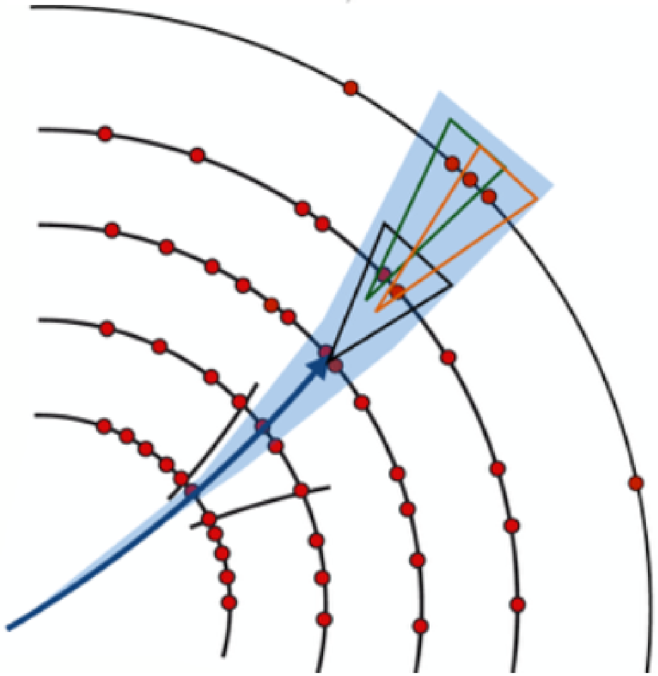
Tracking in 2017
- what can we use? -

Current algorithmic approach (ATLAS, CMS)

- Divide the problem into sequential steps
    1. Cluster hits into 3D spacepoints
    2. Build triplet “seeds”
    3. Build tracks with combinatorial Kalman Filter
    4. Resolve ambiguities and fit tracks

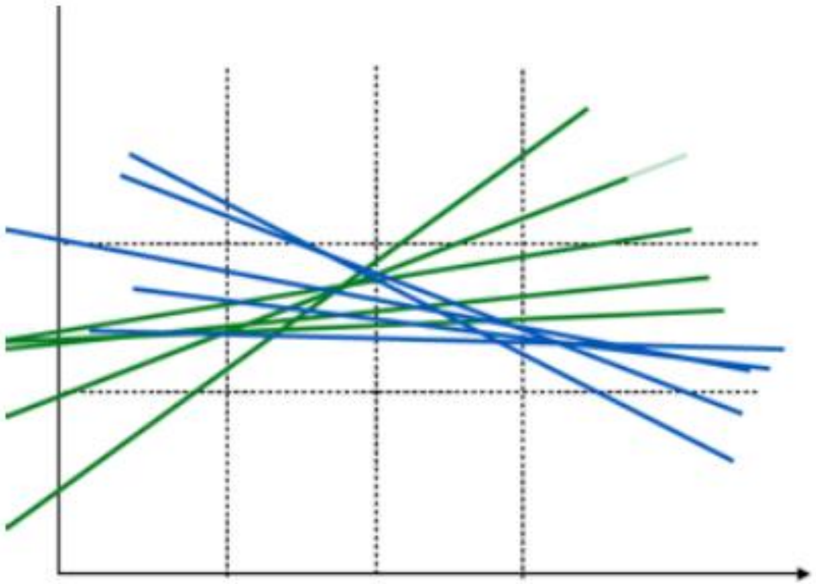




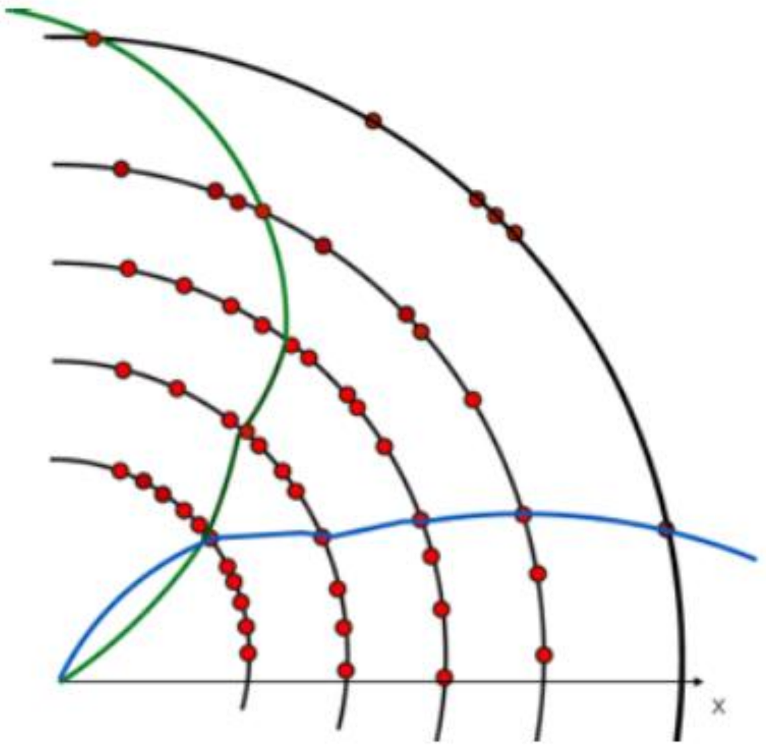


Credit: Andy Salzburger

- Hough Transform breaks down in LHC-like data due to process noise and high occupancy

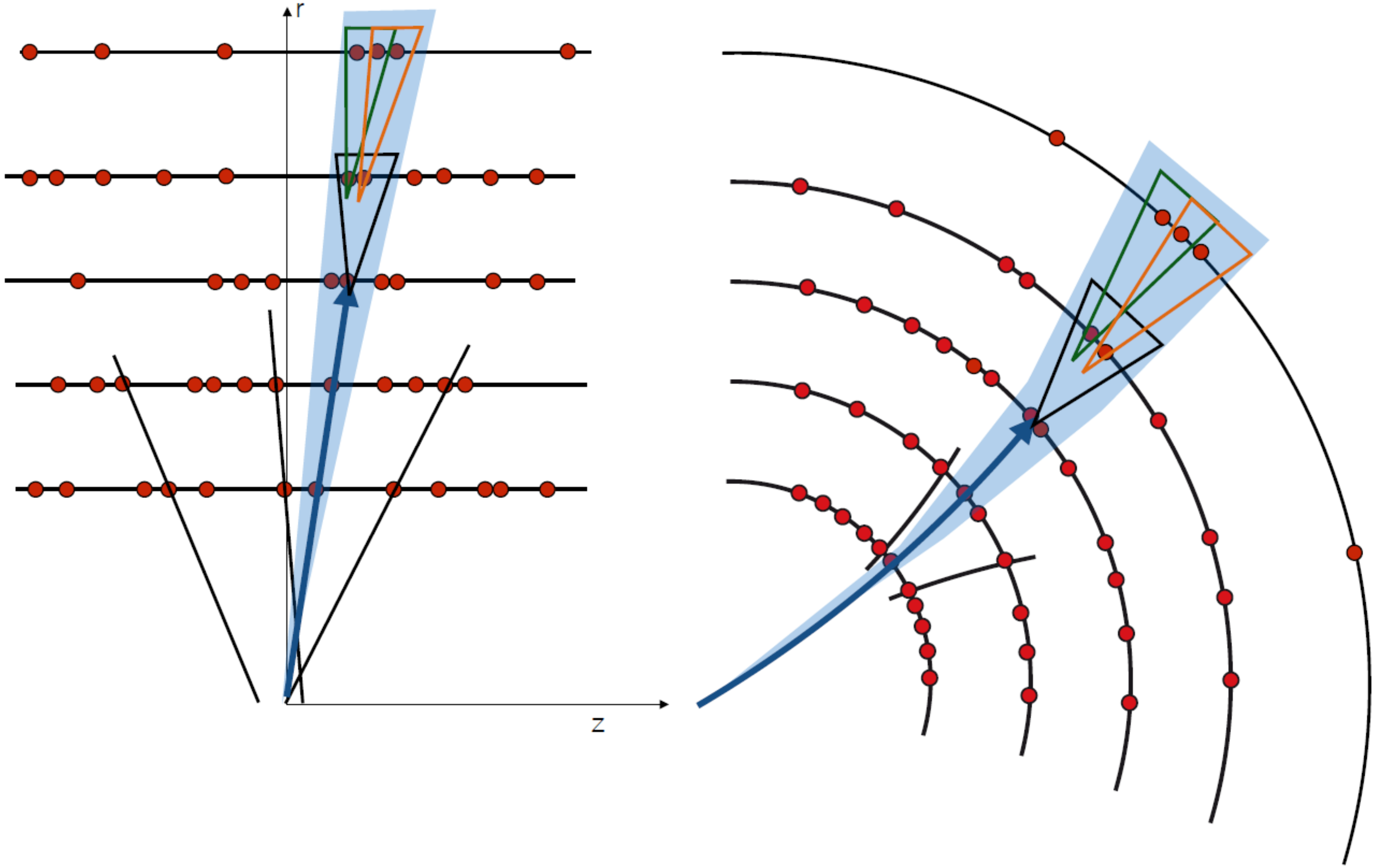


parameter space



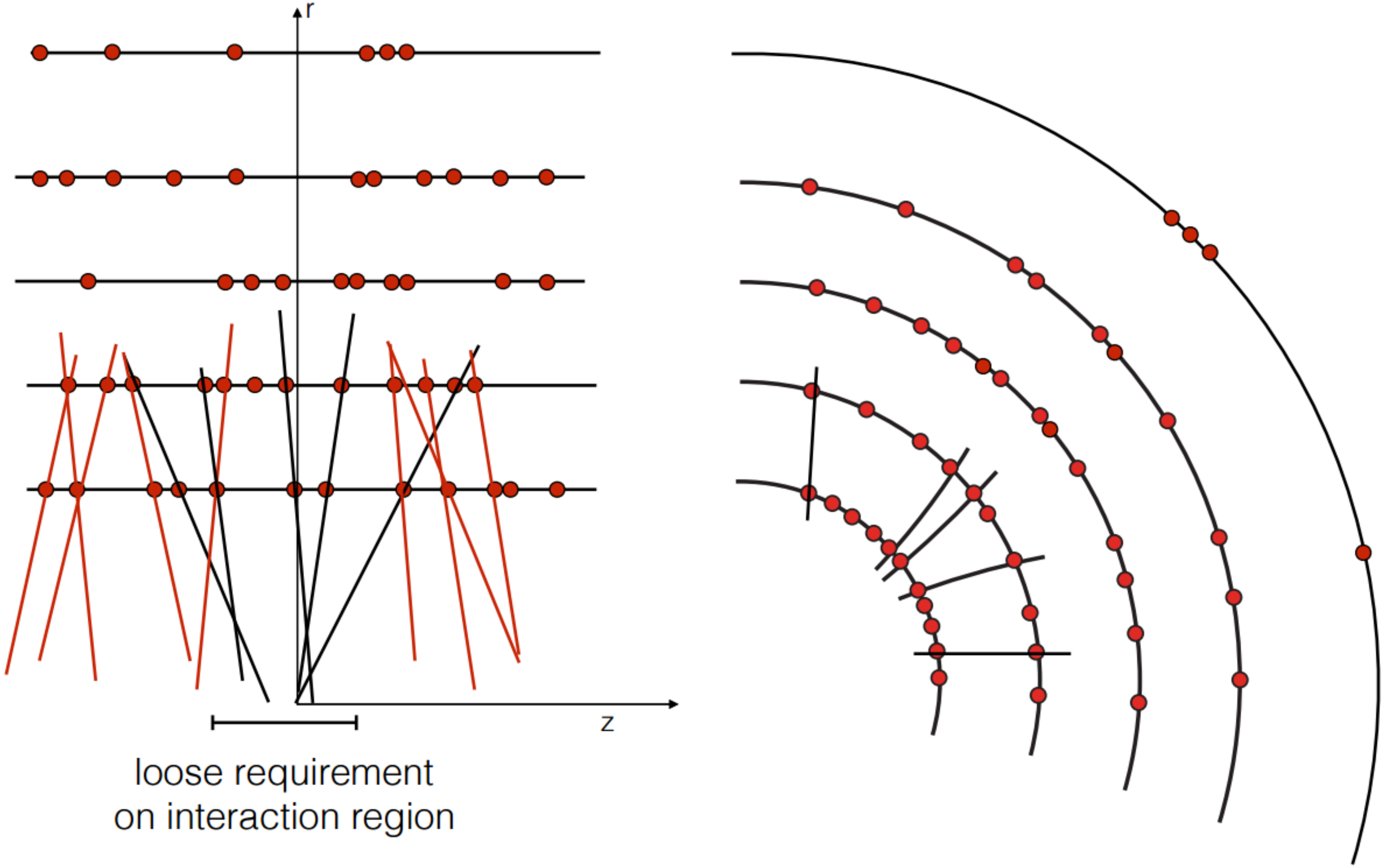
From seeds to track candidates

- Dense environments create problems for the progressive filter
  - there may not always be one obvious path to be followed: The combinatorial filter



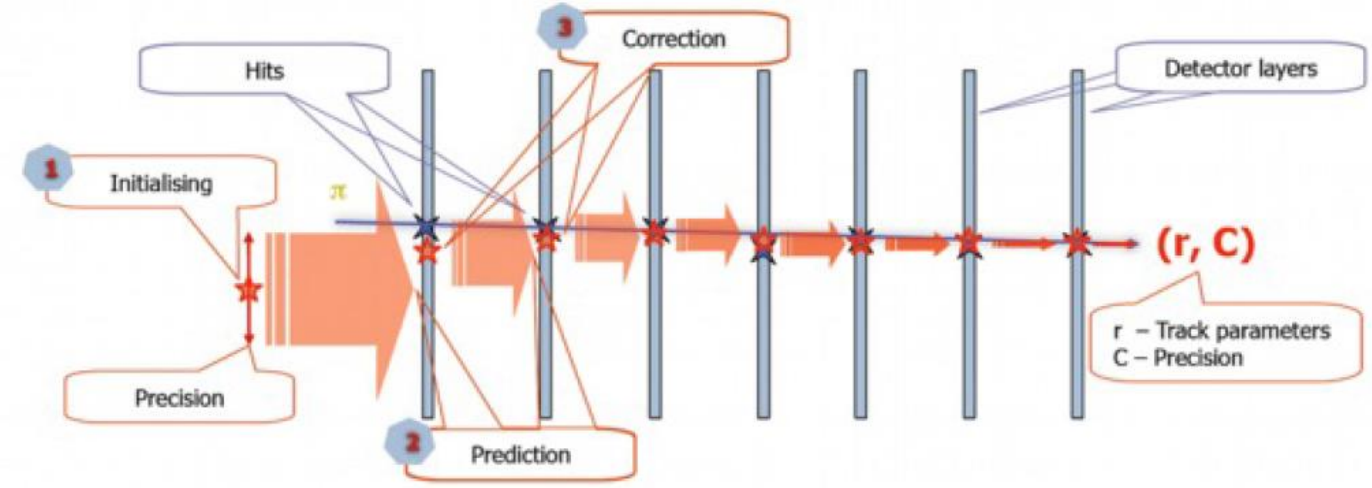
Enemy No. 2: ghosts

- avoid ghosts, i.e. <u>fake combinations from simply combinatorial grouping</u>
  - start off with high quality seeds (clearly 2 hit seeds are not very stringent)



Track reconstruction

- Pattern recognition
  - find a track seed (usually two or three compatible hits)
  - try to extend that, pick up compatible points, build a trajectory
  - ☹ mostly uses **local information**
  - ☹ number of trajectory candidates **must be limited** at each step
  - ☹ keeping some of the best hit-candidates biases the result
  - ☹ decisions are **made too early**
  - ☹ trajectories are **treated separately**



Executive summary – some thoughts

- How to select the best set of tracks?

☺ keep concurrent choices open; several hit-track assignments

⇒ **treat the hits and track candidates as a (bipartite) graph**

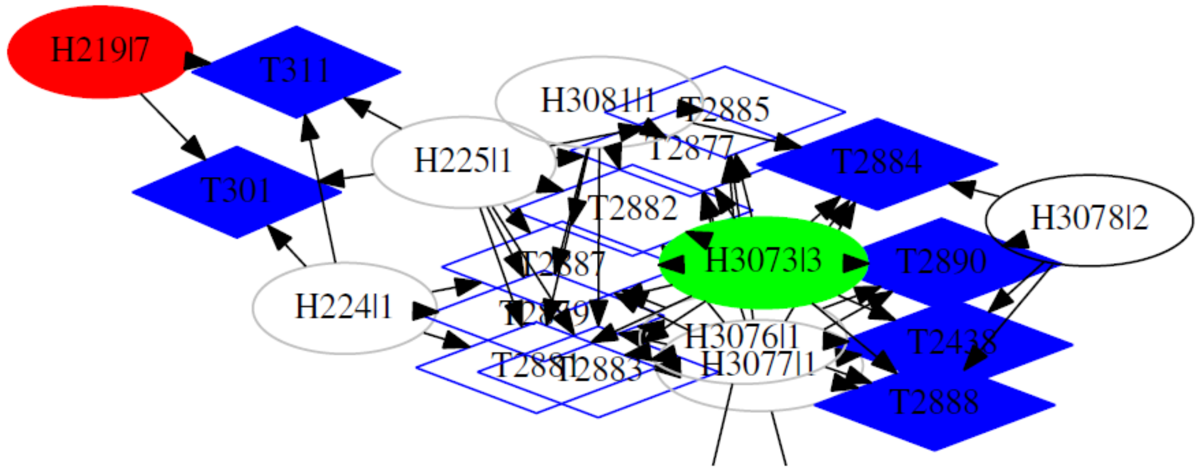
– the graph can be highly connected; but has vulnerable components

☺ disconnect it by looking for *bridges* and *articulation points*

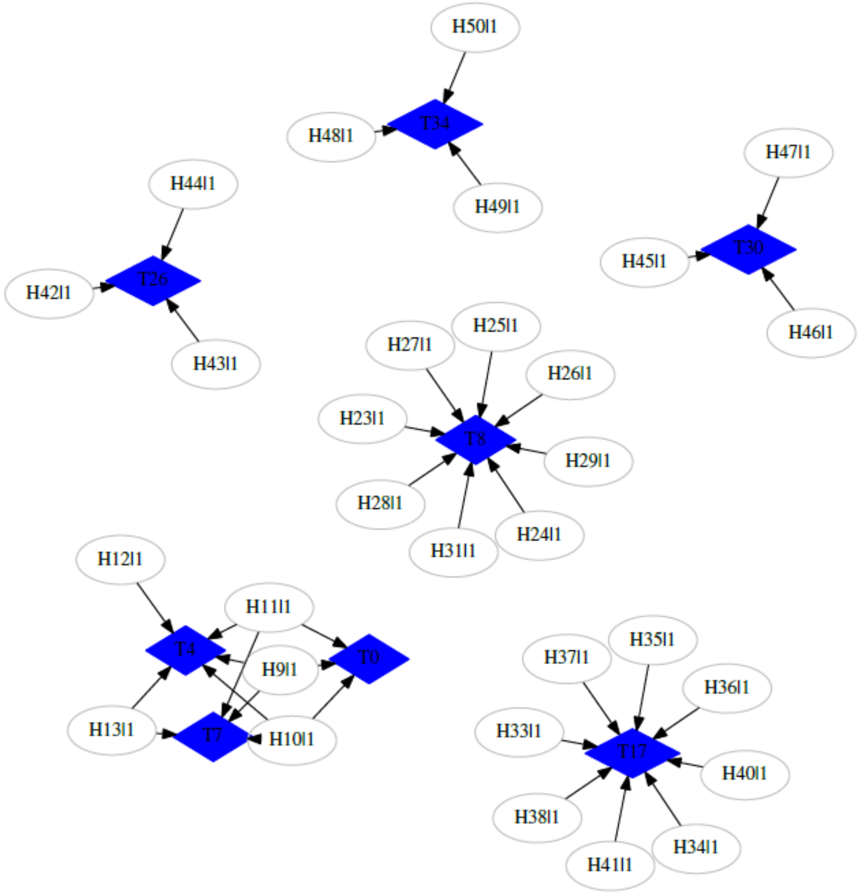
– in the end each hit must belong to at most one track

⇒ **solve subgraphs, decision tree, deterministic single-player**

– maximize the number of hits on track, then minimize $\sum \chi^2$



Convert to graph – single p-p

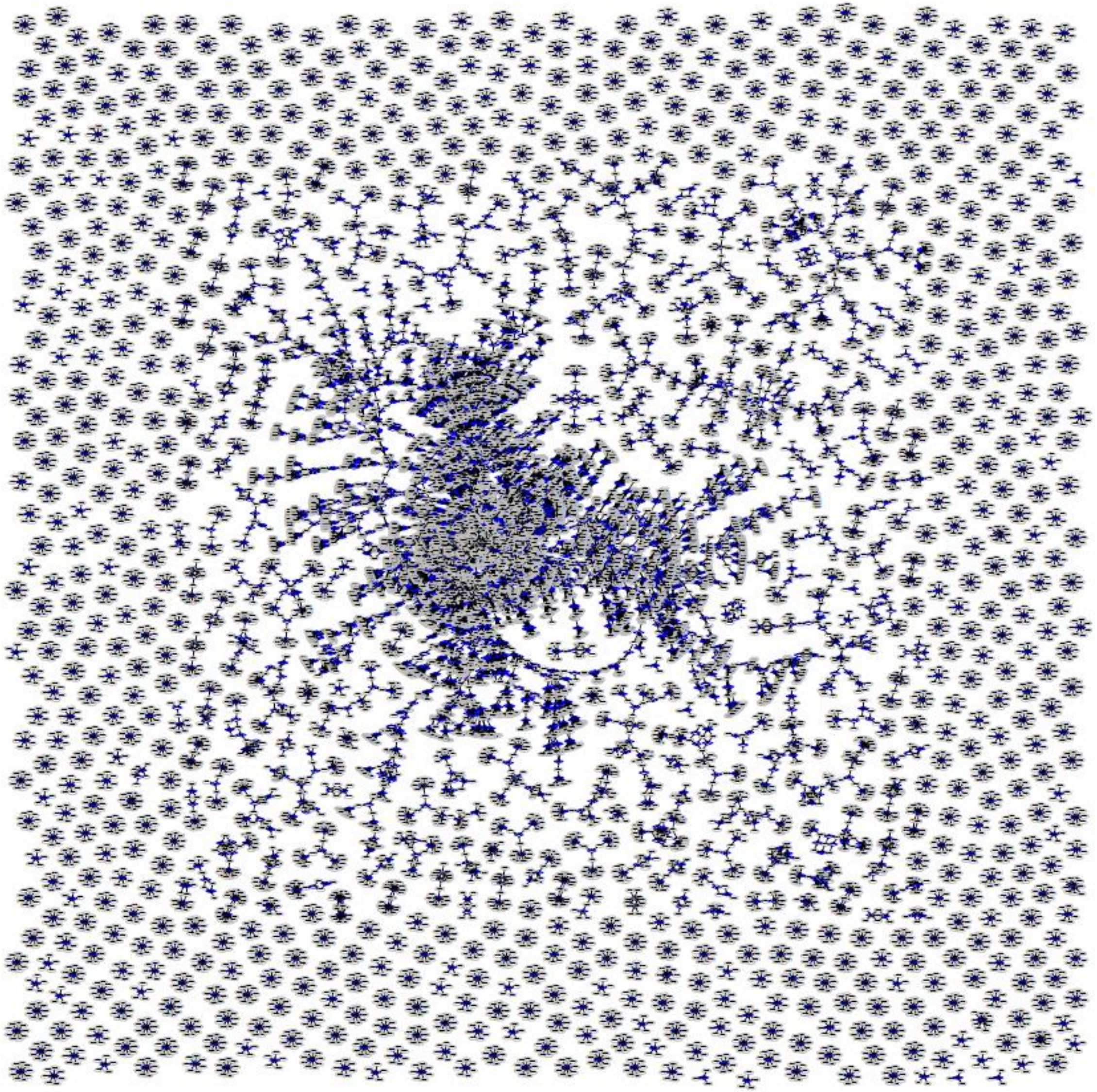


Nodes: hits (ellipses) and track candidates (blue diamonds)

Edges: hits can belong to one or more track candidates

Ambiguity, hit confusion; **optimal packing** problem

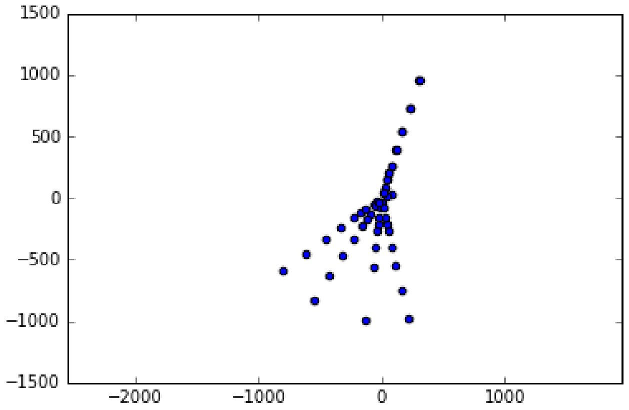
Disconnecting the graph – single Pb-Pb

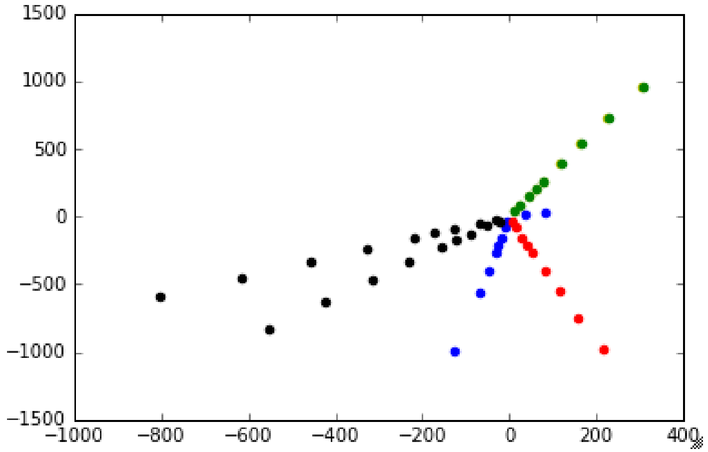


MLRAMP tracking challenge

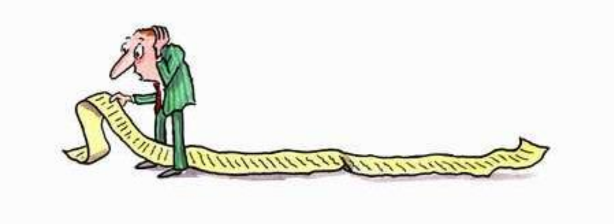
- Goal: Create a 2D tracking algorithm in Python during a 3-hour hackathon (or the following night)

	event_id	cluster_id	layer	iphi	x	y
0	3	4	4	53253	53.900430	-265.585662
1	3	1	5	37216	-47.614439	-402.191329
2	3	1	0	7181	-4.253919	-38.767308
3	3	3	2	7937	44.418132	148.499258
4	3	4	0	7657	7.588600	-38.254583
5	3	2	7	19340	226.475161	727.566493
6	3	2	3	10947	60.079357	204.351342
7	3	2	5	10360	116.504749	387.880965
8	3	2	4	13908	76.915222	259.855823
9	3	5	2	25421	-89.080606	-126.844967
10	3	2	8	25252	303.343256	952.881351
11	3	1	6	51612	-67.634707	-557.915358





team	submission	contributivity	historical contributivity	efficiency
mikhail91	fast_hough3	100	10	0.970
mcherti	starting_kit_test	0	0	0.969
Braun	hyperbelle_tree_6	0	0	0.969
mikhail91	fast_hough2	0	18	0.963
Braun	hyperbelle_tree_5	0	0	0.961
calaf	delphi_multilayer	0	0	0.961
pettersen	linear_propagation2	0	0	0.850
pettersen	linear_propagation	0	0	0.845



```
from collections import defaultdict

import numpy as np
from sklearn.base import BaseEstimator
from scipy import optimize

layer_id = 0
phi_id = 1
x_id = 2
y_id = 3
number_id = 4

# SORRY! All comments are still missing...

class Clusterer(BaseEstimator):
    def __init__(self, cut=0.0001, duplicate_cut=0.1):
        self.cut = cut
        self.duplicate_cut = duplicate_cut

        self.layers = list(range(9))

        self.hit_masks_grouped_by_layer = {}
        self.not_used_mask = None

        self.weights = np.zeros((8, 50))
        self.dphiRange = 0.1

    def fit(self, X, y):
        events = y[:, 0]
        for event in np.unique(events):
            event_indices = events == event
            X_event = X[event_indices]
            pixelx = X_event[:, 3]
            pixely = X_event[:, 4]
            phi = np.arctan2(pixely, pixelx)
            layer = X_event[:, 1]
```

Miscellanous

Multi-Purpose Particle Detector for Space Missions

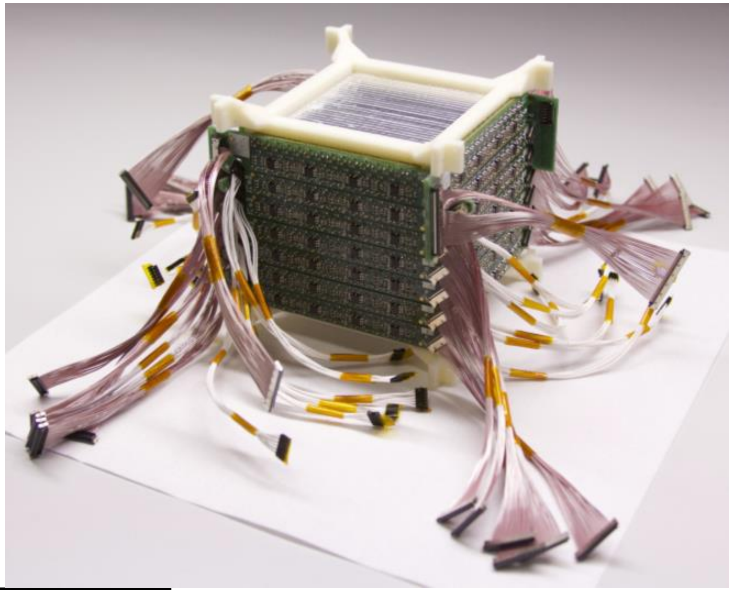
COGNITIVE NEUROSCIENCE

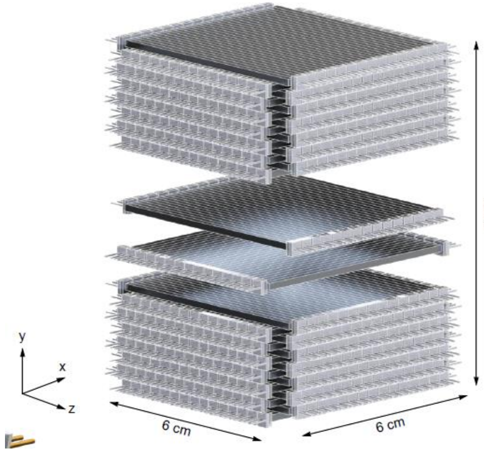
What happens to your brain on the way to Mars

Vipan K. Parihar,[1] Barrett Allen,[1] Katherine K. Tran,[1] Trisha G. Macaraeg,[1] Esther M. Chu,[1] Stephanie F. Kwok,[1] Nicole N. Chmielewski,[1] Brianna M. Craver,[1] Janet E. Baulch,[1] Munjal M. Acharya,[1] Francis A. Cucinotta,[2] Charles L. Limoli[1]*

10.1126/sciadv

As NASA prepares for the first manned spaceflight to Mars, questions have surfaced concerning the potential for increased risks associated with exposure to the spectrum of highly energetic nuclei that comprise galactic cosmic rays. Animal models have revealed an unexpected sensitivity of mature neurons in the brain to charged particles found in space. Astronaut autonomy during long-term space travel is particularly critical as is the need to properly



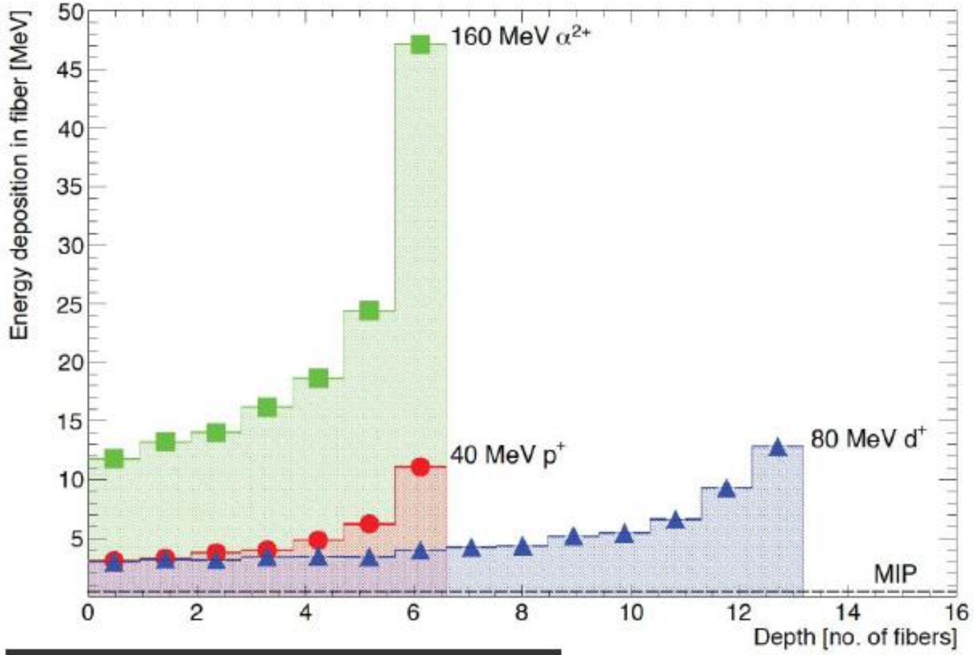


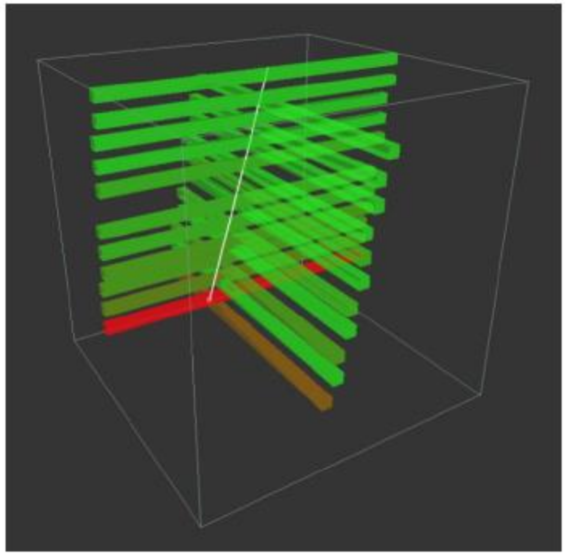
Event Reconstruction

Goal: reconstruct direction and particle characteristics (type, charge, energy) for individual events

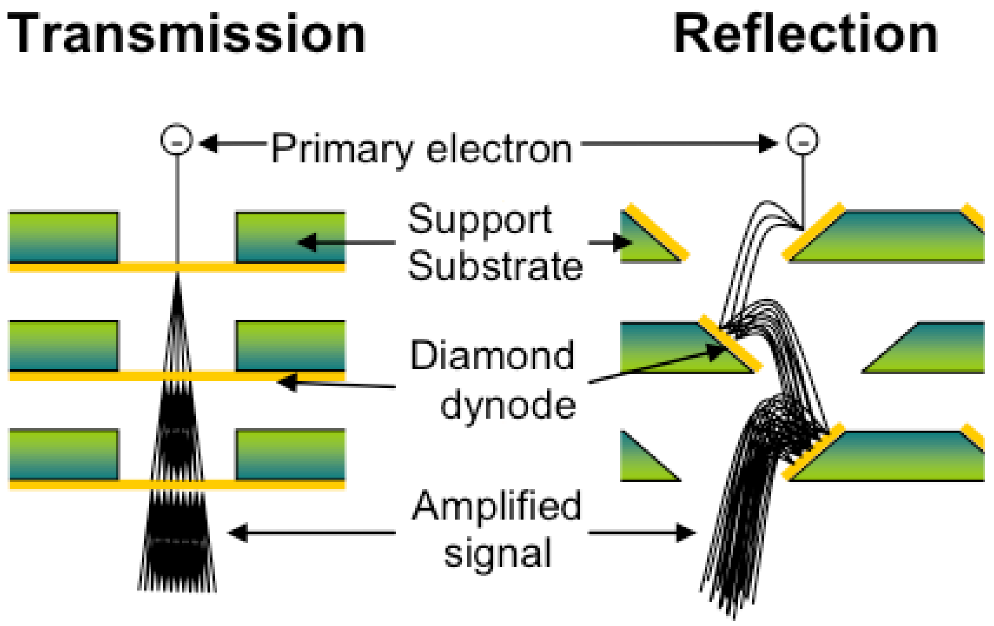
Bragg Curve Spectroscopy

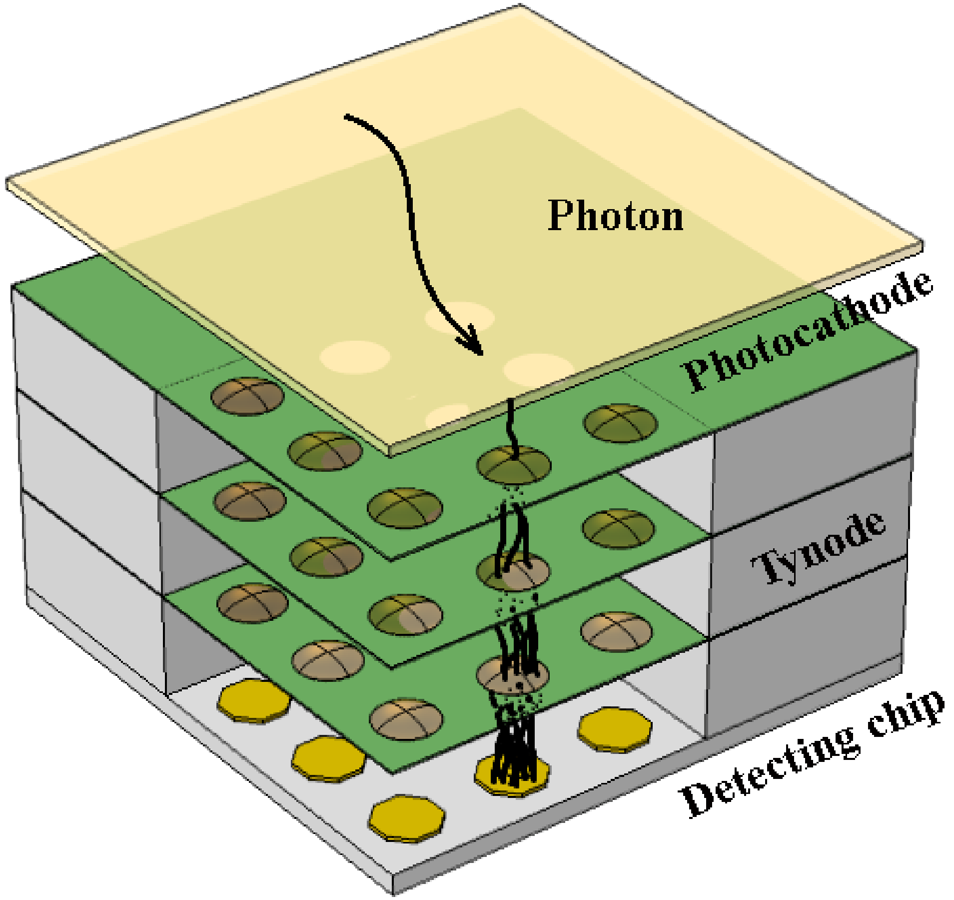
- energy-loss profile along the particle's track is unique for low-energy ions
- shape of profile depends on velocity, charge, and mass of the particle
- extrapolation of Bragg curve for through-going particles extends measurement range





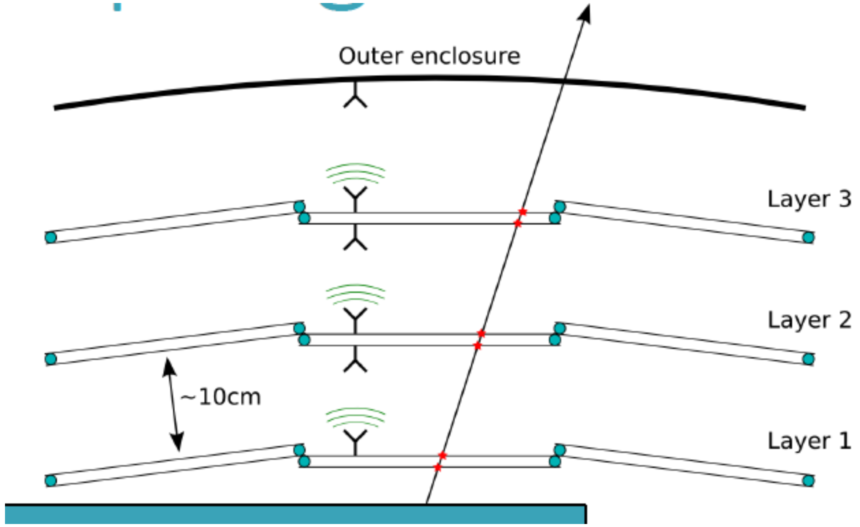
Transmission dynode: A quick e- multiplier

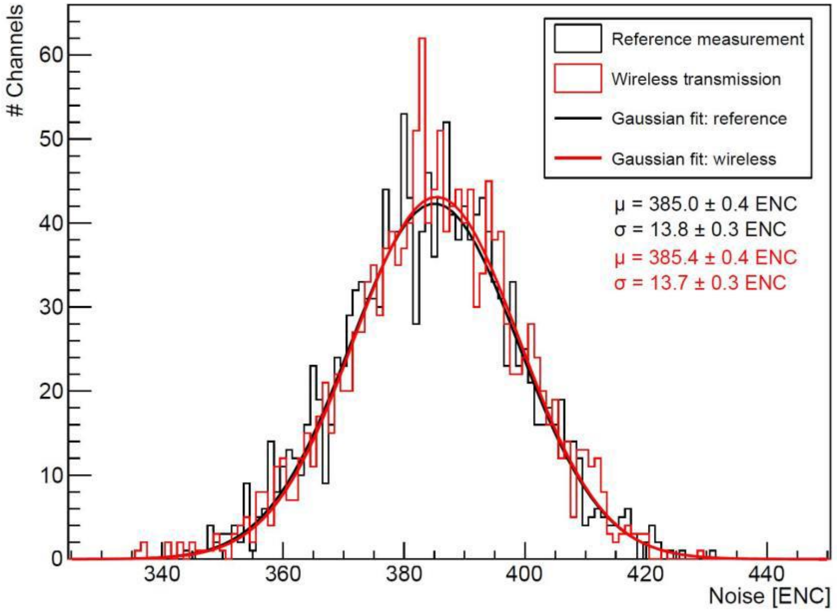


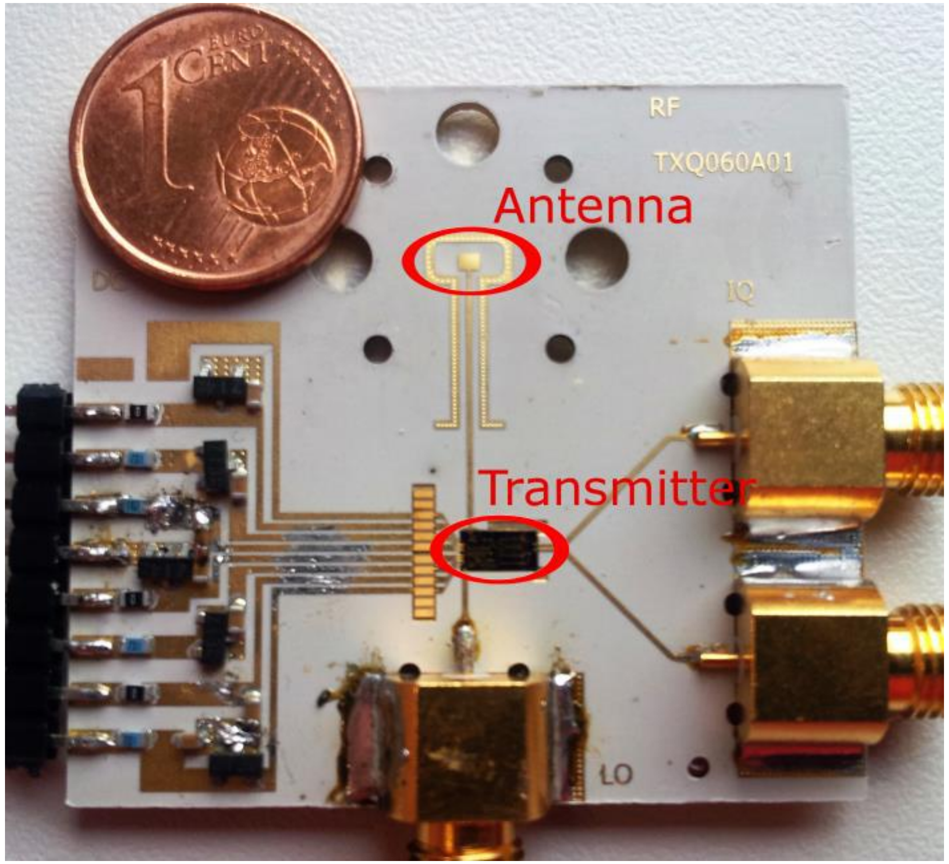


- A stack of tynodes may generate the so far fastest detector charge signal, completed within 5 ps

Wireless data transmission for HEP

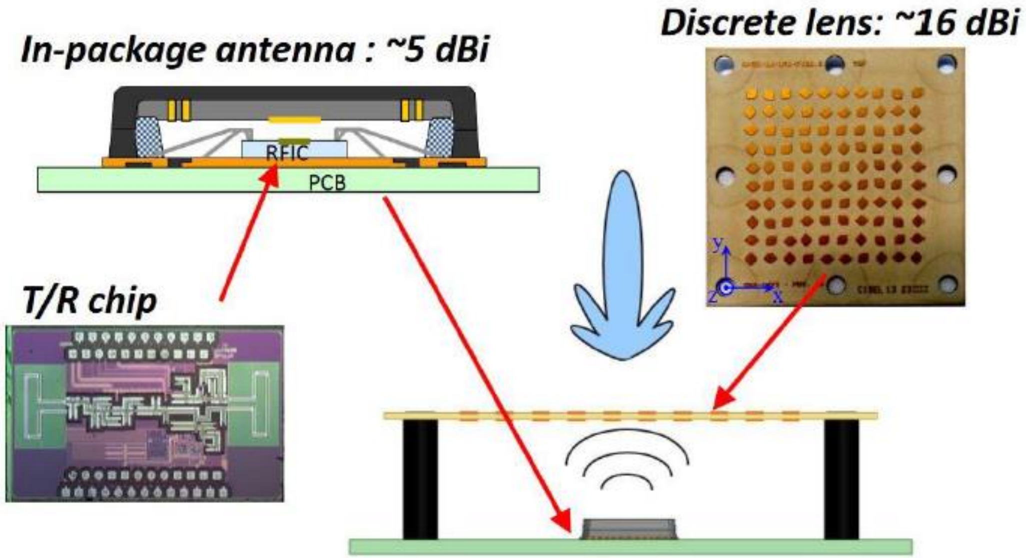






60 GHz developments (CEA-LETI)

- CMOS 65nm time domain transceiver
- OOK/ASK modulation
- Short range: 2-3 cm ⟶ can be increased with discrete lens
- Power consumption: Tx 40 mW, Rx 20mW
- Data rates 0.5 – 8 Gb/s



Gradual abstraction in medical visualization