MEG II陽電子スペクトロメータにおける機械学習を活用したヒット再構成の改善

Improving hit reconstruction for MEG II positron spectrometer using machine learning technique

内山 雄祐
on behalf of MEG II collaboration

日本物理学会2022年秋季大会
令和4年9月8日

8aA421-2

MEG II in search of $\mu^+ \to e^+\gamma$

- Upgraded from MEG experiment
- An intensity frontier experiment
- To get definitive evidence for BSM

UTokyo
KEK
Kobe Uni.

PSI
ETHZ

INFN Genoa
INFN Lecce
INFN Pavia
INFN Pisa
INFN Roma

BINP
JINR

UC Irvine

~60 physicists

MEG result (2016)

$$B(\mu^+ \to e^+\gamma) < 4.2\times10^{-13}$$

@90% C.L.
(while 5.3×10^{-13} expected)

higher intensity muon beam
higher resolution everywhere
higher efficiency

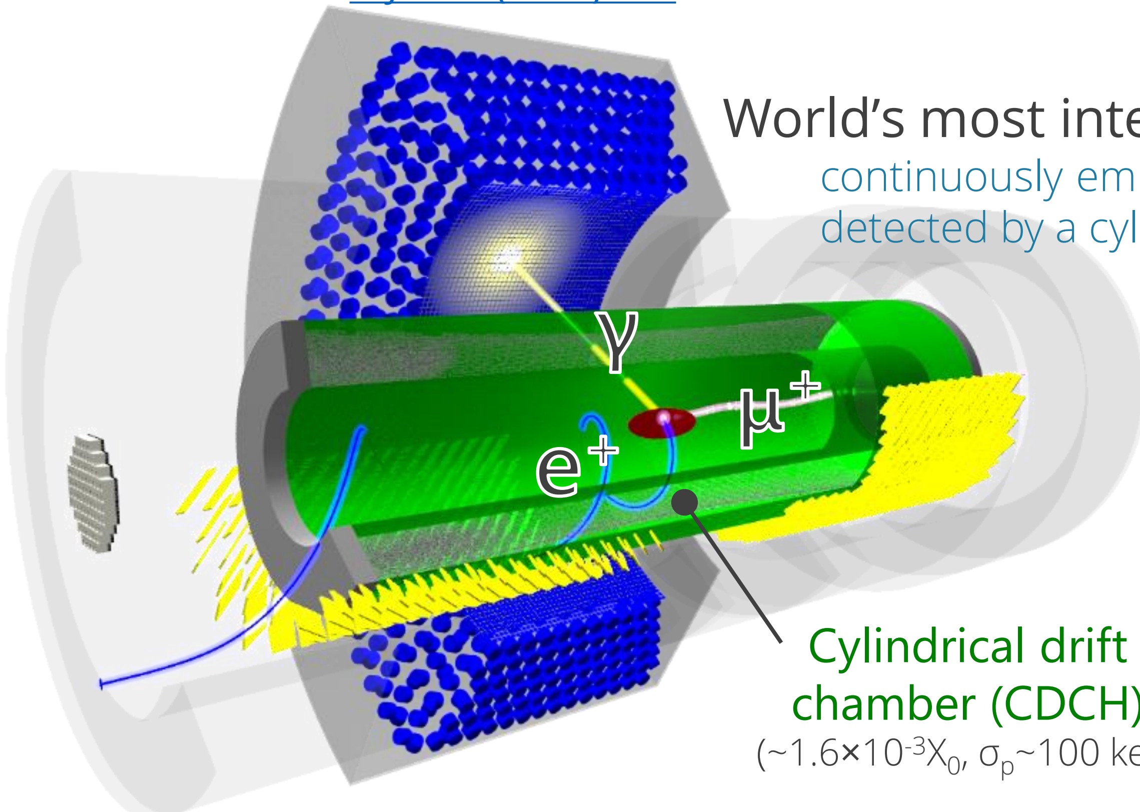
From 2021, searching for $\mu^+ \to e^+\gamma$ down to

$$6\times10^{-14}$$

(90% C.L. sensitivity)

MEG II

EPJ-C **78** (2018) 380



World's most intense DC muon beam

continuously emits (3 – 6) ×10^{7} e^{+}/s

detected by a cylindrical drift chamber

Cylindrical drift chamber (CDCH)

(~1.6×10^{-3}X$_0$, σ_p~100 keV)

Detector signals are read out as waveform

by DRS4 waveform digitizer

1024 points @ 1.2 GSPS

Started physics data taking in 2021!

In this study, use 2021 data.

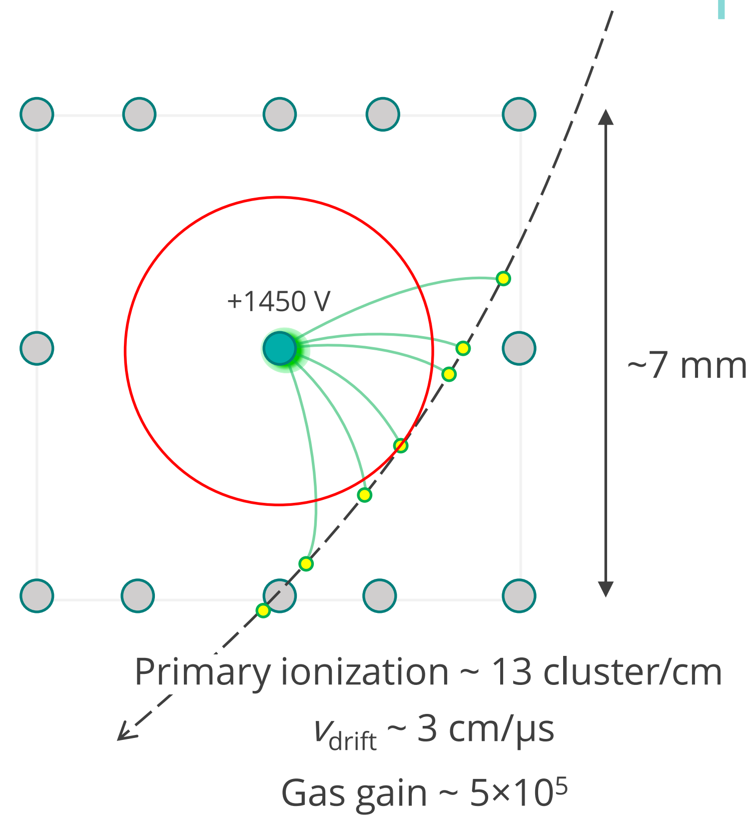
Drift chamber: a nutshell

Signal formation

1. Charged particle generates primary ionization clusters discretely in gas
2. The ionized e⁻s drift to an anode wire and form avalanche near the wire

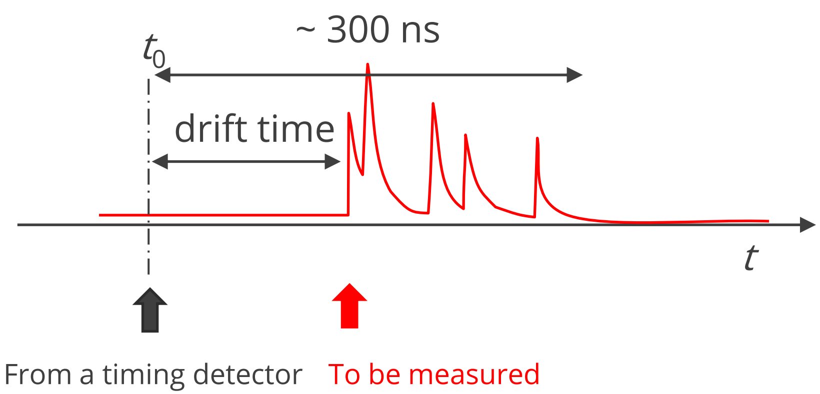
Reconstruction

1. Measure the timing of the 1st cluster
2. Draw a drift circle
3. Fit a track to the drift circles



MEG II CDCH: an ultra low-mass chamber

Gas: He:iC_4H_{10} = 90:10
Wires: 20 μm W anode + 40/50 μm Al cathode
2 m long, 9 layers, 1152 readout cells in total



Challenges

Detecting the 1st cluster signal is essential for the experiment

The efficiency is directly connected to the e^+ reconstruction efficiency, and thus, search sensitivity.

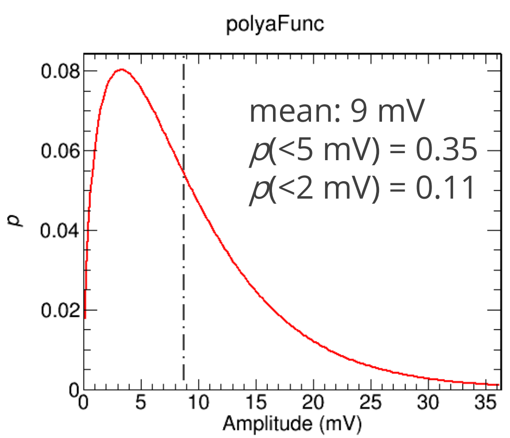
Two difficulties:

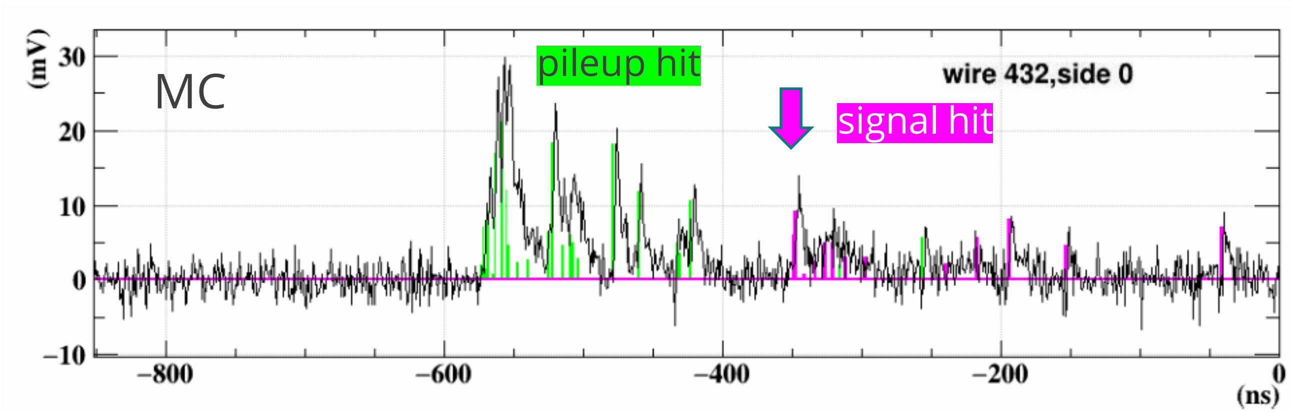
1. S/N

   The amplification process (gas gain) has **large fluctuation** obeying a Polya distribution.
   The 1st cluster signal can be **very small**.

2. Pileup

   Very high hit rate in MEG II: up to **1.7 MHz** per cell, 35% occupancy in 250ns.



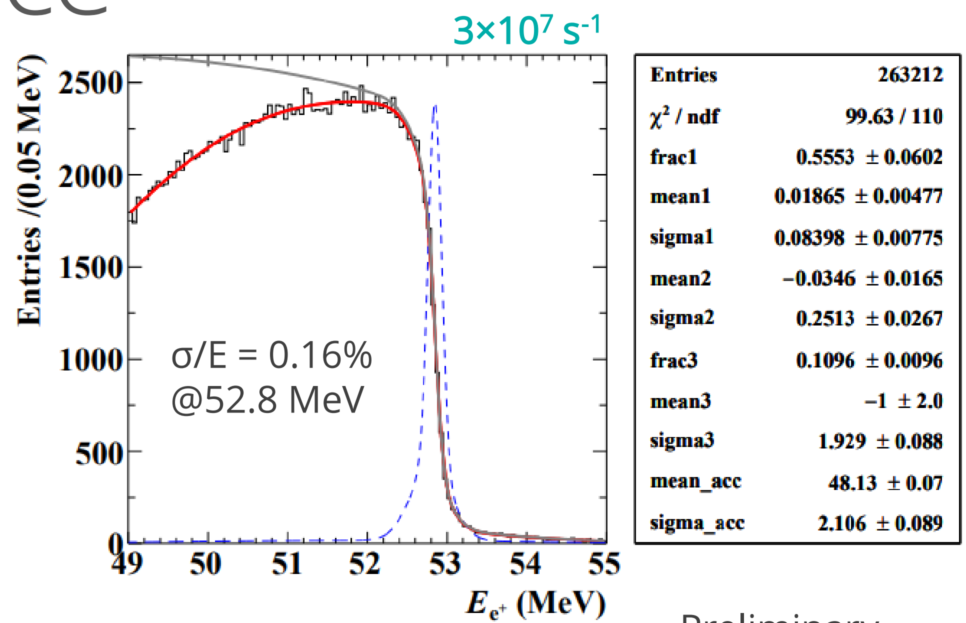


Apply ML to the complicated waveform analysis.

Current performance

A good momentum resolution achieved

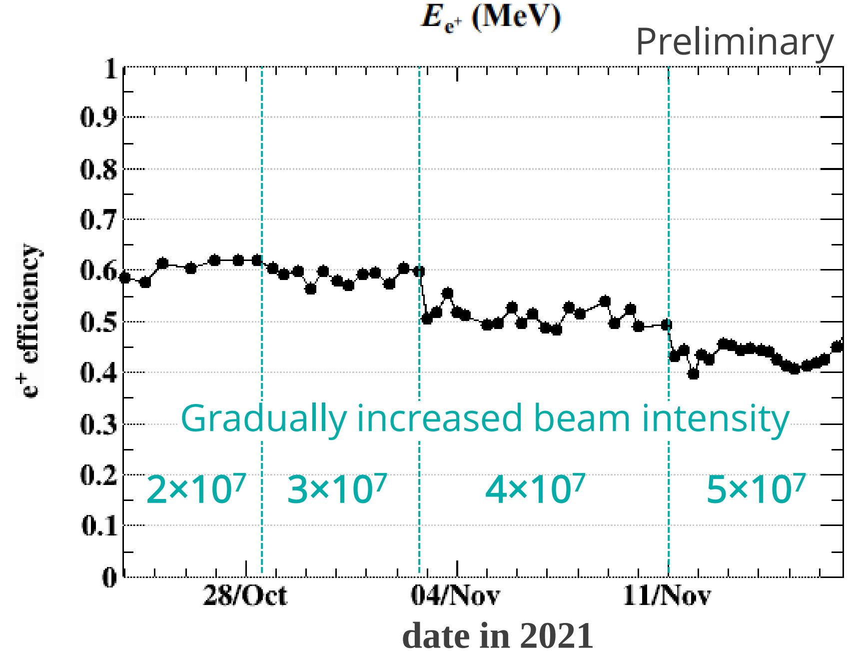
- >3 times better than that in MEG



Efficiency is not as good as designed

- In particular, at high beam intensity,
- nevertheless, ~2 times better than MEG

- The goal: 70%

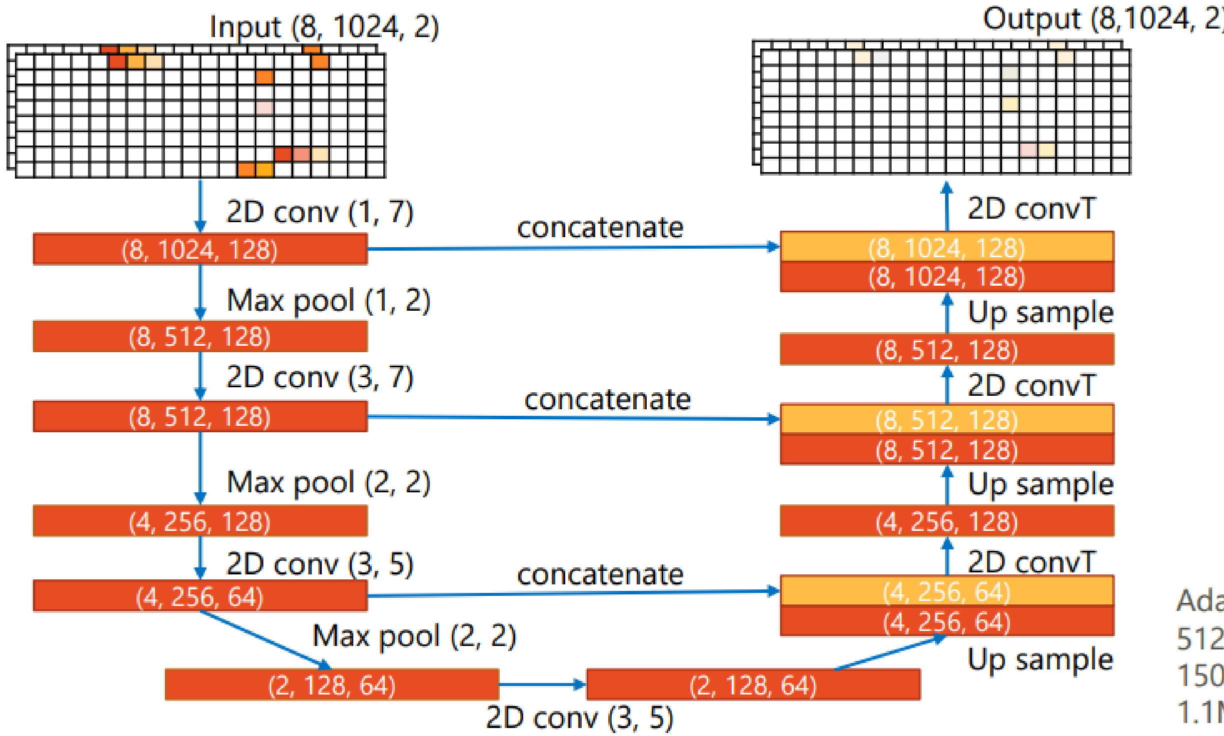


9

The model: noise estimation

Waveform from 8 wires × 2 ends

Training with real noise data + MC signal

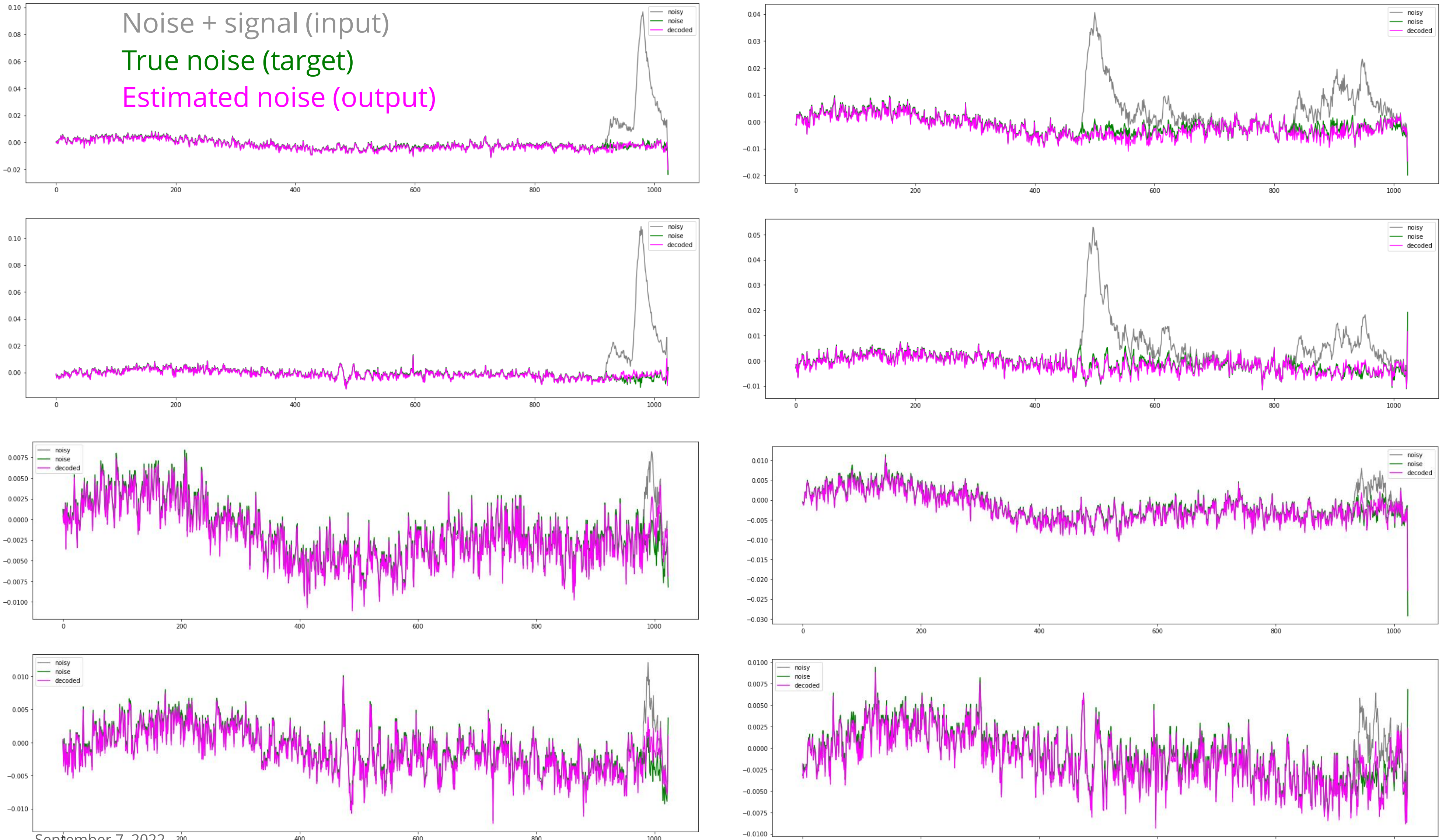


Adam, ReLU,
512 batch size,
150 epochs,
1.1M parameters

- 2D convolutional network
- 'UNet'-like structure with skip connections https://arxiv.org/abs/1505.04597 (image segmentation)
- 2-channel input with 2-end waveforms from 8 wires
- Use 'mean squared error (mse)' loss function.

This is equivalent to the residual learning

Noise estimation with 2D CNN autoencoder



Implementation

TRAINING

Tensorflow 2.8 + Keras

in Python3.7

on Google Colab Pro

with Tensor Processing Unit (TPU)

convert to ONNX format

INFERENCE

ROOT based MEG II reconstruction framework

in C++17

ONNX Runtime C++ API

with CPU single thread (Xeon Gold 6138 2.0 GHz)

High flexibility × Easy maintenance

Use one's preferred package for model building & training.

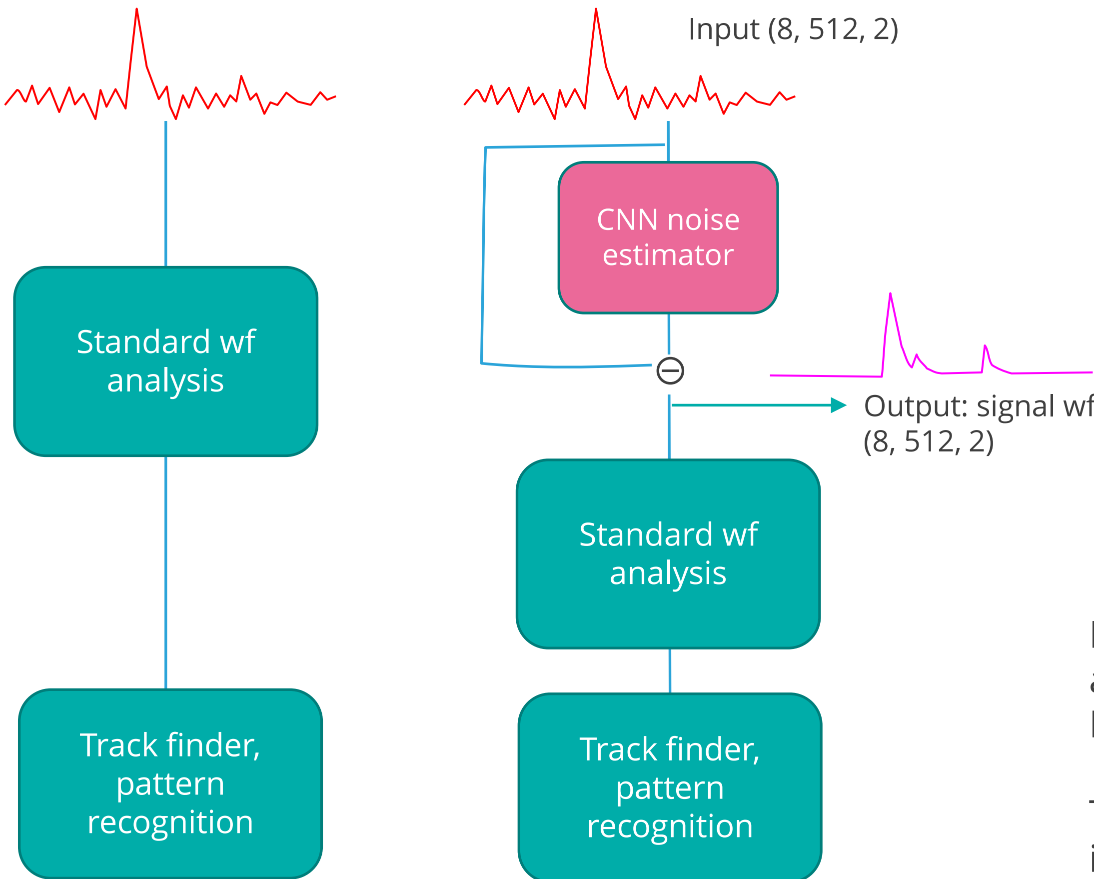
Use a common interface in C++ to use the trained model in inference/prediction.

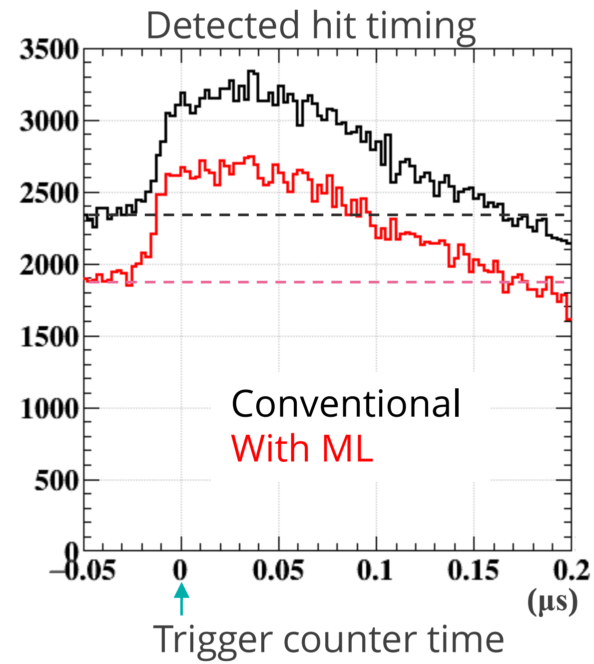
GPU/TPU in cloud are available for training, while only CPU (single thread) is available in the MEG II resource & framework.

Apply to 2021 data

Not better than the conventional analysis.

- Almost comparable performance with the best model
- but with > 10 times computation time (3.4 s/event)

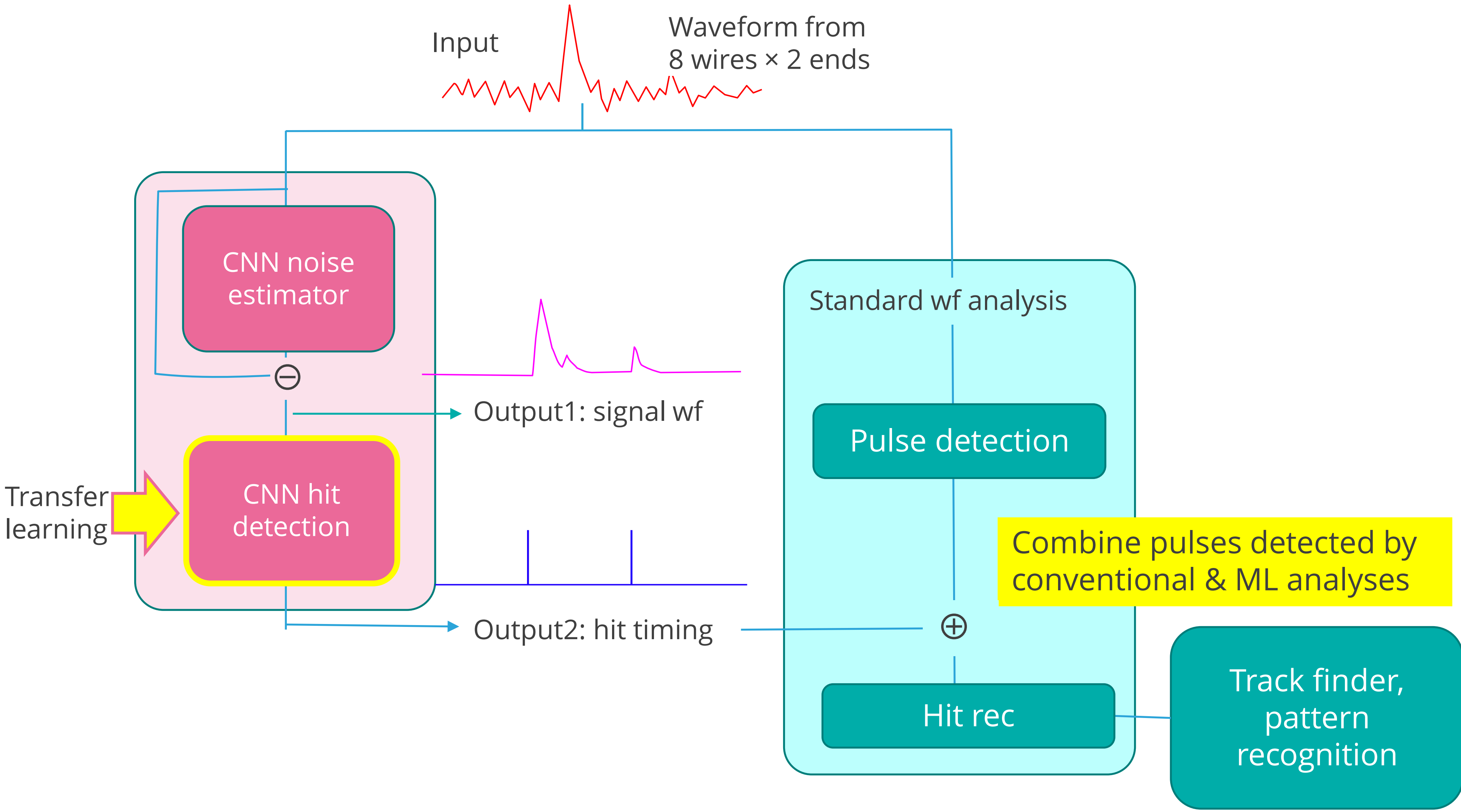




ML detects almost same number of hits associated with trigger e^+ with less fake hits. However, final tracking efficiency is lower.

Track finder can reject fake hits. Better to input more hits.

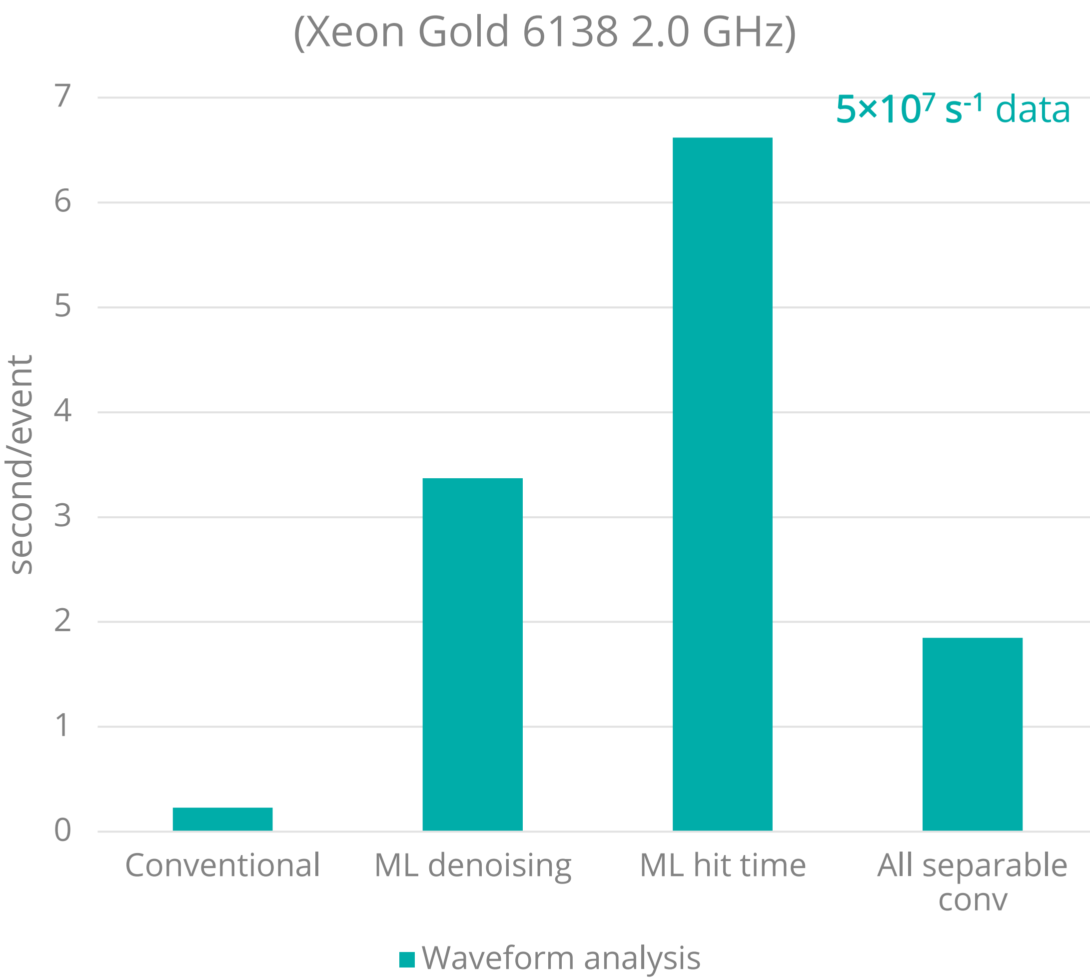
Extended model



Efficiency

0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
3×10^7 s^-1 data
5×10^7 s^-1 data
14%
23%
Conventional (before this study)
Conventional (tuned)
ML denoising
ML denoising + ML hit timing
Conventinal + ML hit timing
Conventinal + ML hit timing (separable conv)
3×10^7
5×10^7
Two different architectures (Discussed later)

Computation time (ML)



~2300 ch waveform / event

Inference speed is an issue!

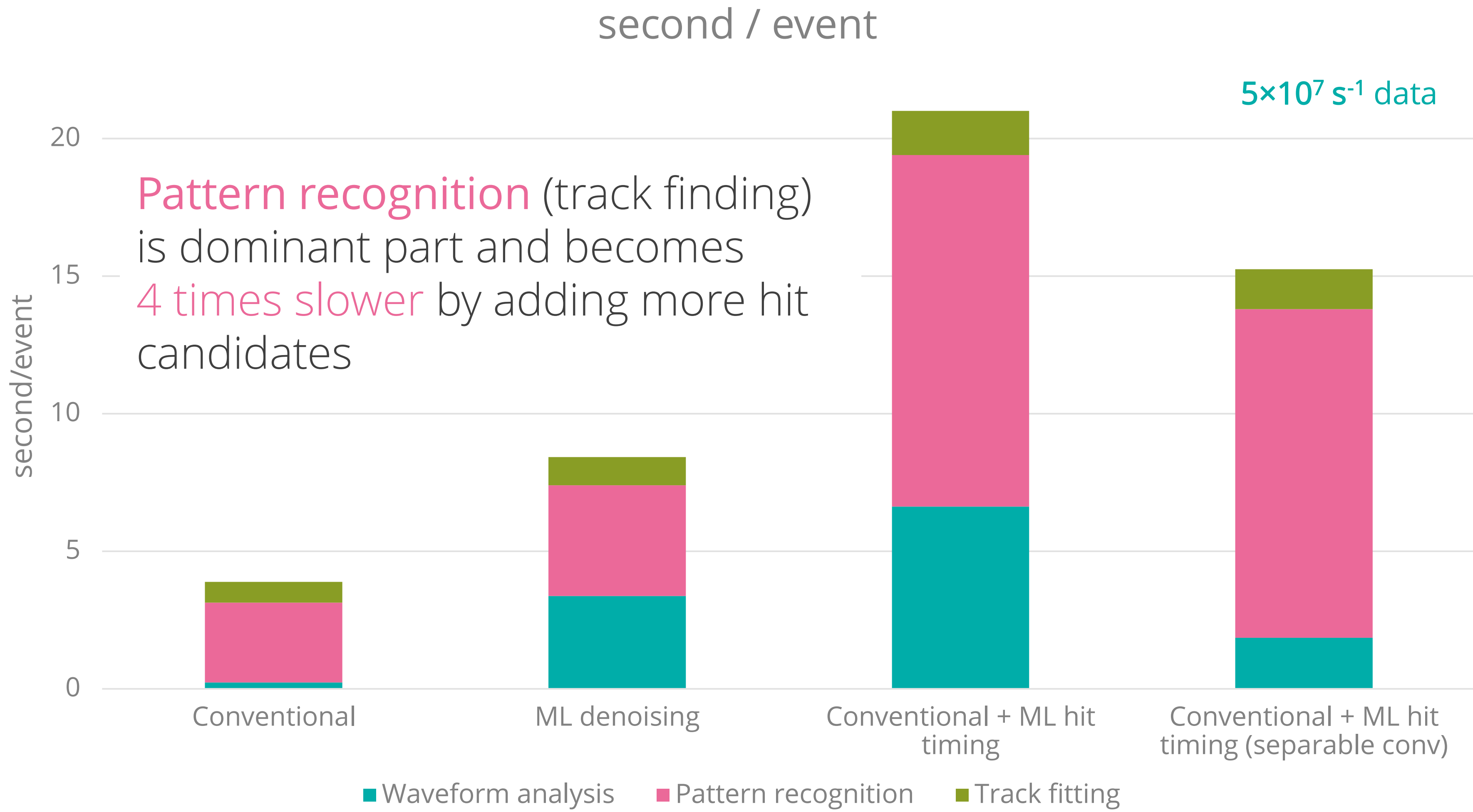
- GPU/TPU is not available in MEG II resource & framework; only CPU (single thread).
- ~30 times slower (Unfeasible!)
- Can speedup with 'separable convolution'

  Replacing all conv-layers with separable conv speeds up by factor 3.5 with comparable performance.

  Feasible!

However, more practical problem is…

Computation time (Total)



With current resource, it is not feasible to apply this to all events…

Conclusions

Efficiency can be improved by feeding more hits to track finder

Even contaminated by fake or wrong-timing hits
Track finder is clever enough to find true tracks.

ML model was extended (from denoising) to detecting hit timing and the results are added to the results from conventional waveform analysis. → 23% higher efficiency.

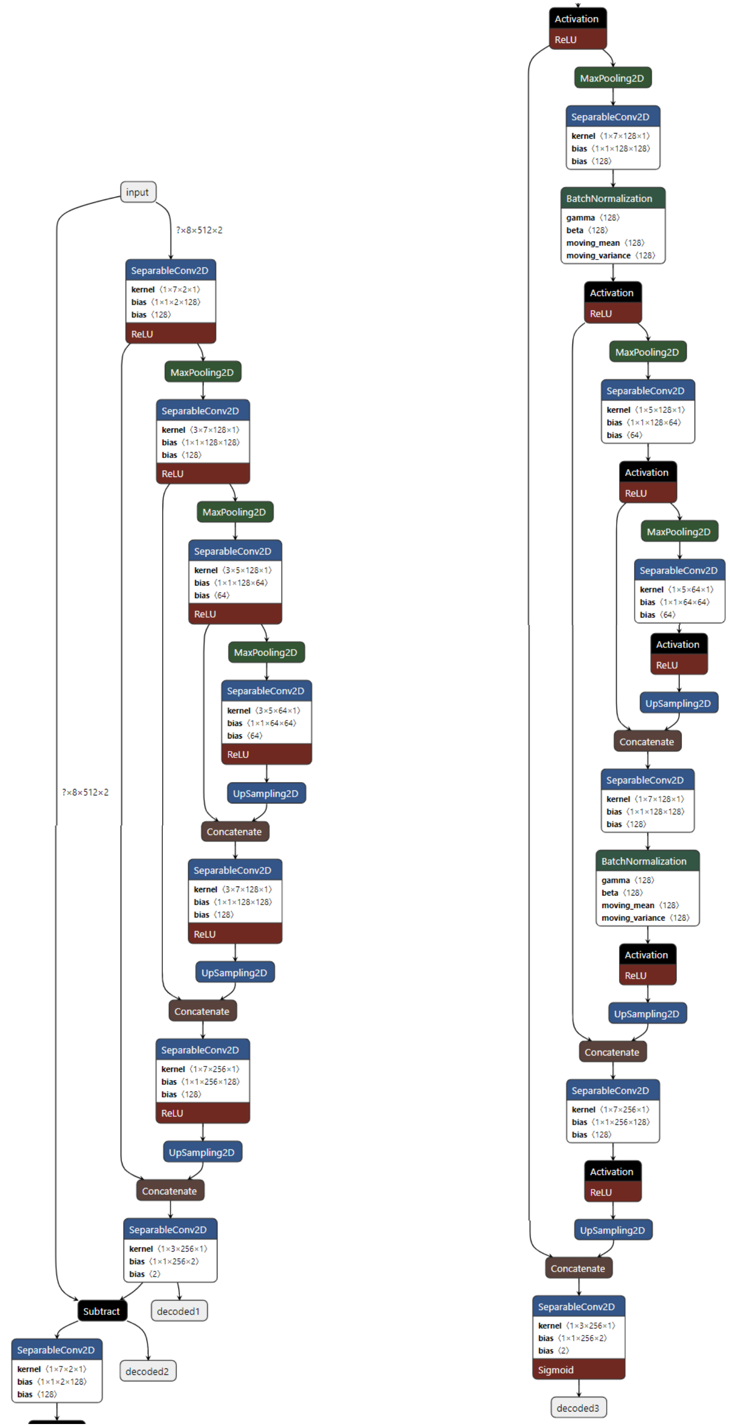
Try transformer, maybe better for hit detection.

Significant increase in computation time is a problem

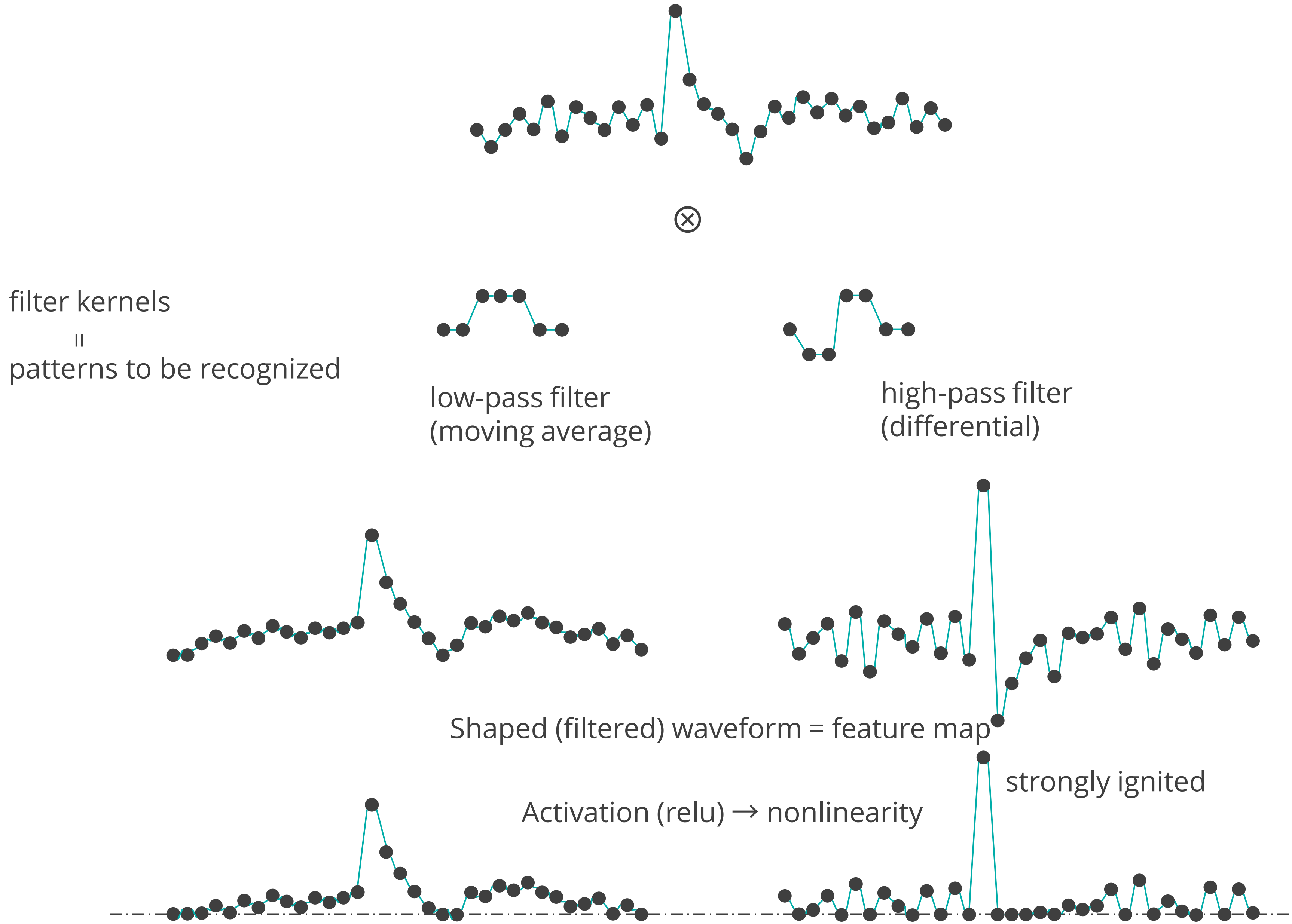
More by track finder than ML itself.

Next: developing global pattern recognition with ML to handle large number of hits within realistic computation time.

Architecture

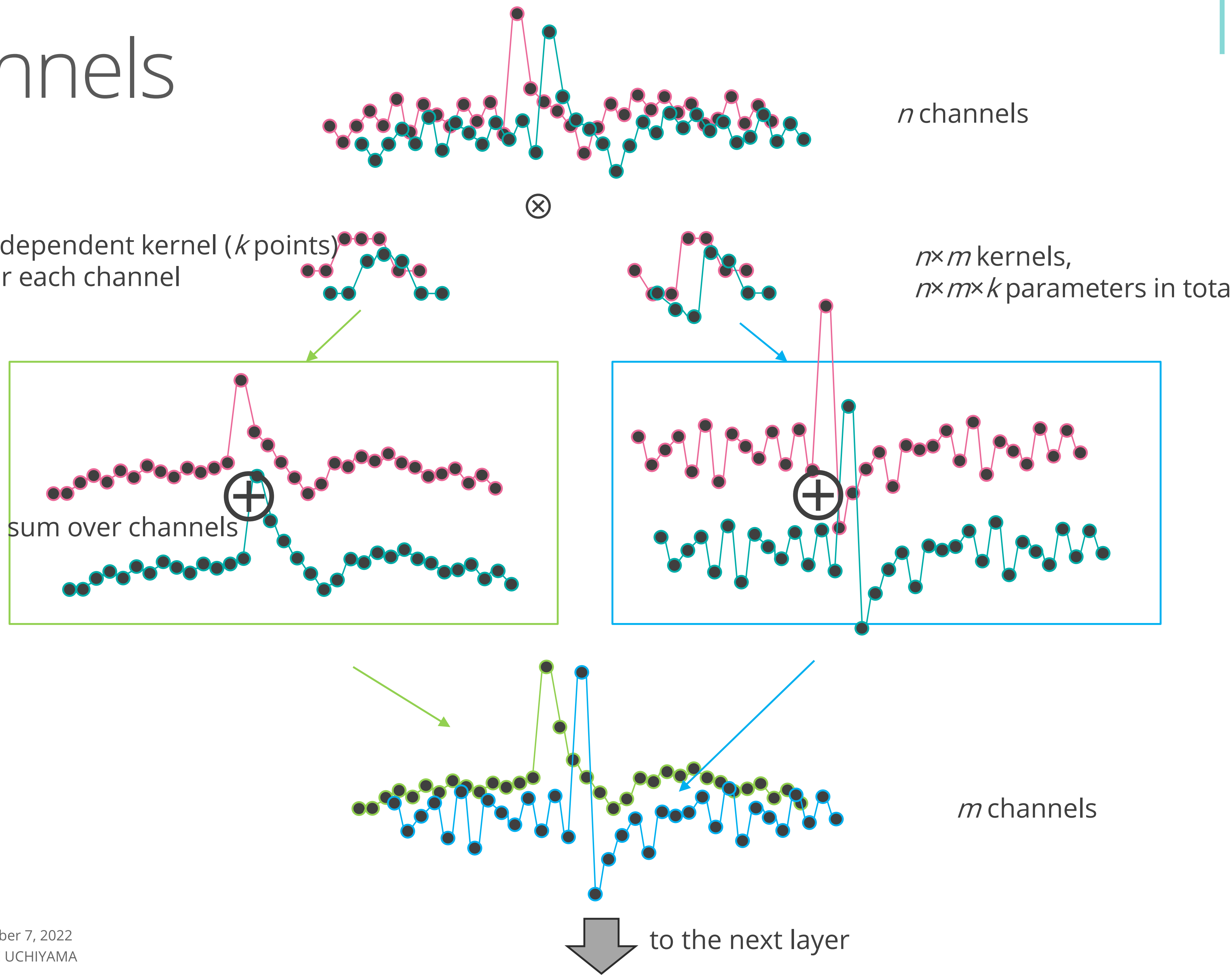


September 7, 2022
YUSUKE UCHIYAMA



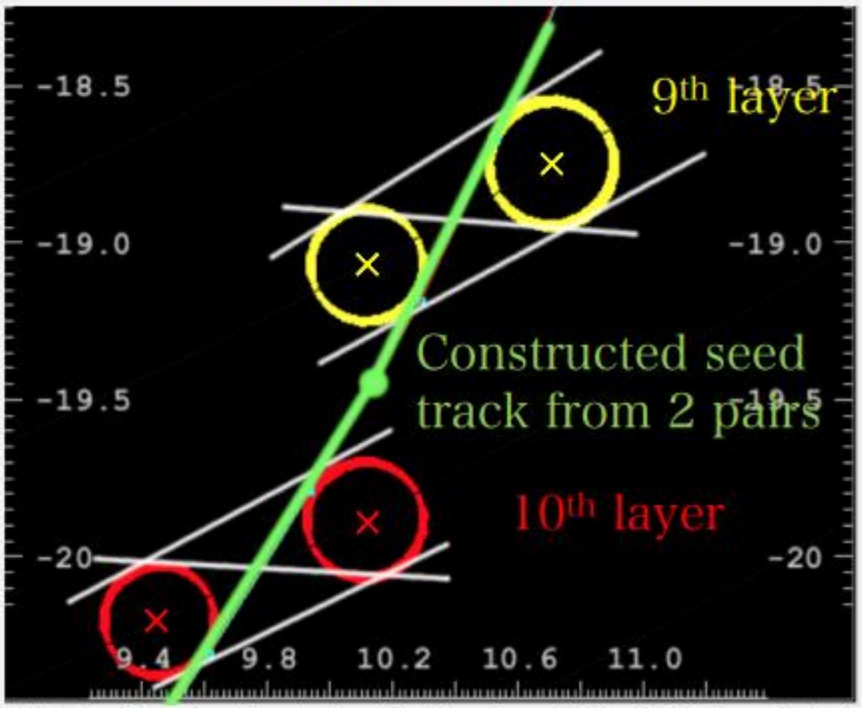
⊗
filter kernels
=
patterns to be recognized
low-pass filter
(moving average)
high-pass filter
(differential)
Shaped (filtered) waveform = feature map
Activation (relu) → nonlinearity
strongly ignited

channels



n channels
⊗
independent kernel (k points)
for each channel
n×m kernels,
n×m×k parameters in total
sum over channels
m channels
to the next layer

Track finder



Original: http://cmd.inp.nsk.su/~ignatov/meg2Tracking_Lecce_2016.pdf, 2016 June, Fedor Ignatov

<u>**Local (track following) method:**</u>

1. T0 from scintillation counter.
2. Pick up pair-hits in the same layer
3. Select 2 pairs in adjacent layers and test the 4 possible patterns. → A track seed.
4. Prolong the track seed to add hits with Kalman filter.