AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

AGH

Using deep neural networks to improve the precision of fast-sampled particle timing detectors

M. Kocot[1], K. Misan[1], V. Avati[1], E. Bossini[2], L. Grzanka[1], N. Minafra[3]

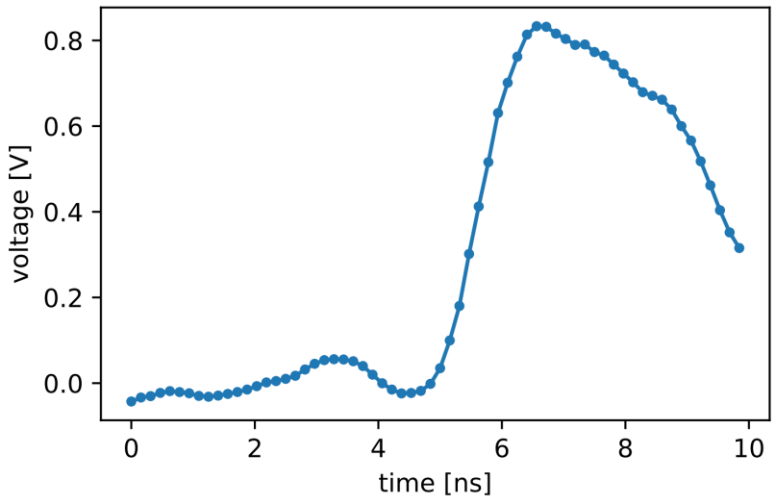
1. AGH University of Science and Technology, Kraków, Poland
2. INFN Sezione di Pisa, Pisa, Italy
3. Department of Physics and Astronomy, University of Kansas, Lawrence, KS, USA

FAST 2023

Contents

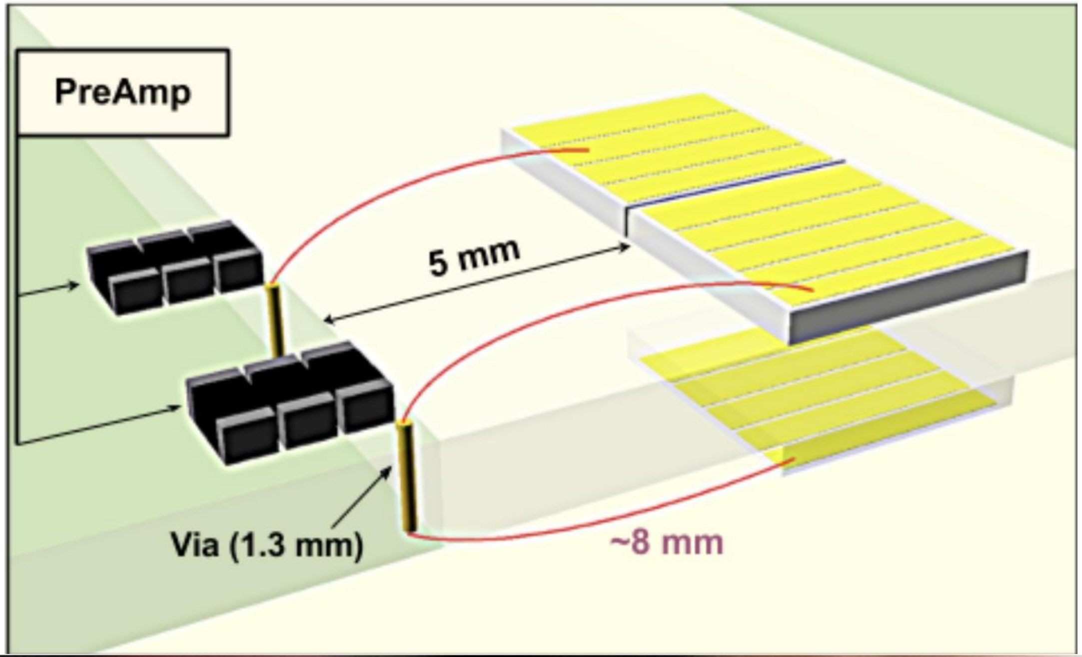
Time of arrival prediction

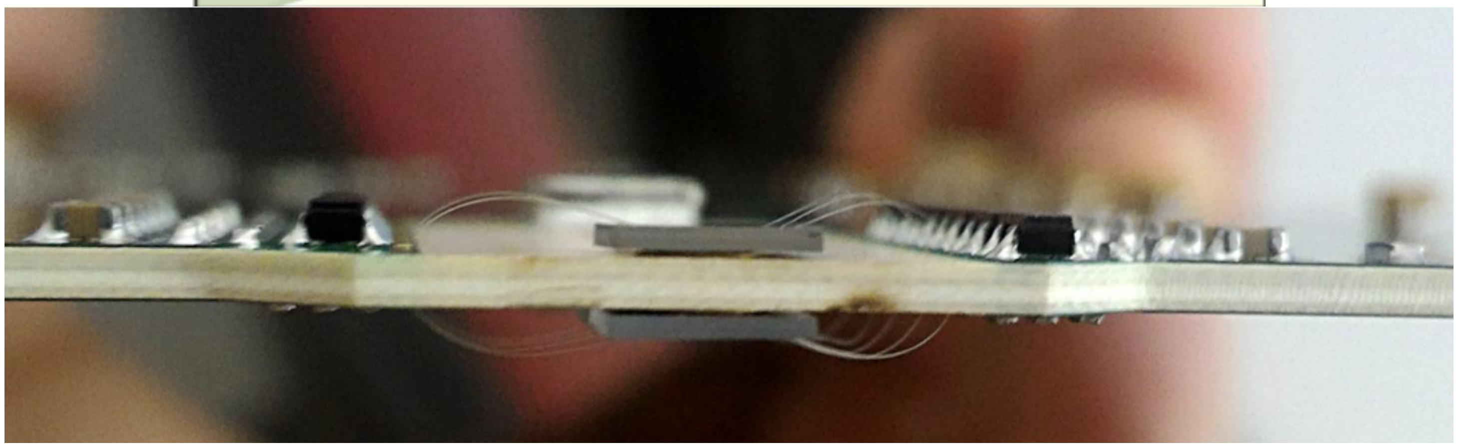
- Diamond detectors (*double diamond* architecture)
  - Devised and used in the CMS-PPS (Precision Proton Spectrometer) system, at the LHC (CERN).
- A particle flying through a detector generates a voltage signal.
- A sampling device (SAMPIC) produces a sampled **time series of voltage**.
- Measurement goal: precise timing of the passage of the particle
- **Project goal: estimate the performance of neural networks with respect to the method used currently.**



Example time series from a diamond detector

Double diamond



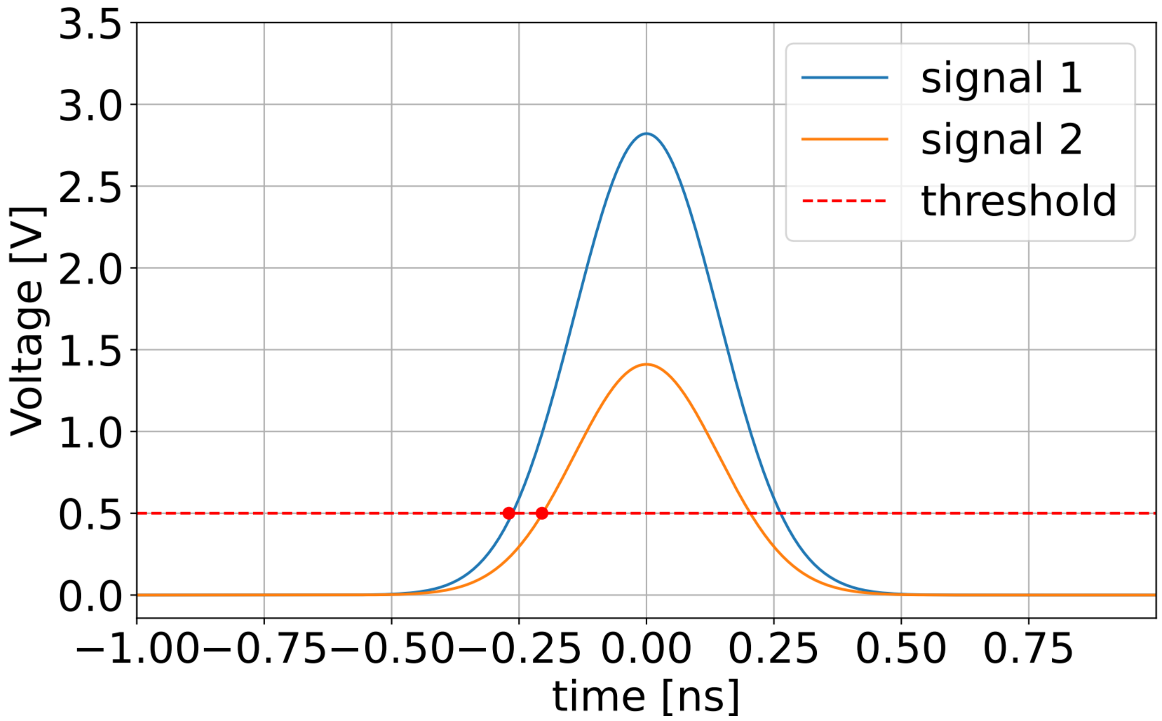


Two diamond sensors on both sides of the board are connected to the same readout channel.

Time walk effect

» Easiest algorithm to compute the time of arrival: **constant threshold**

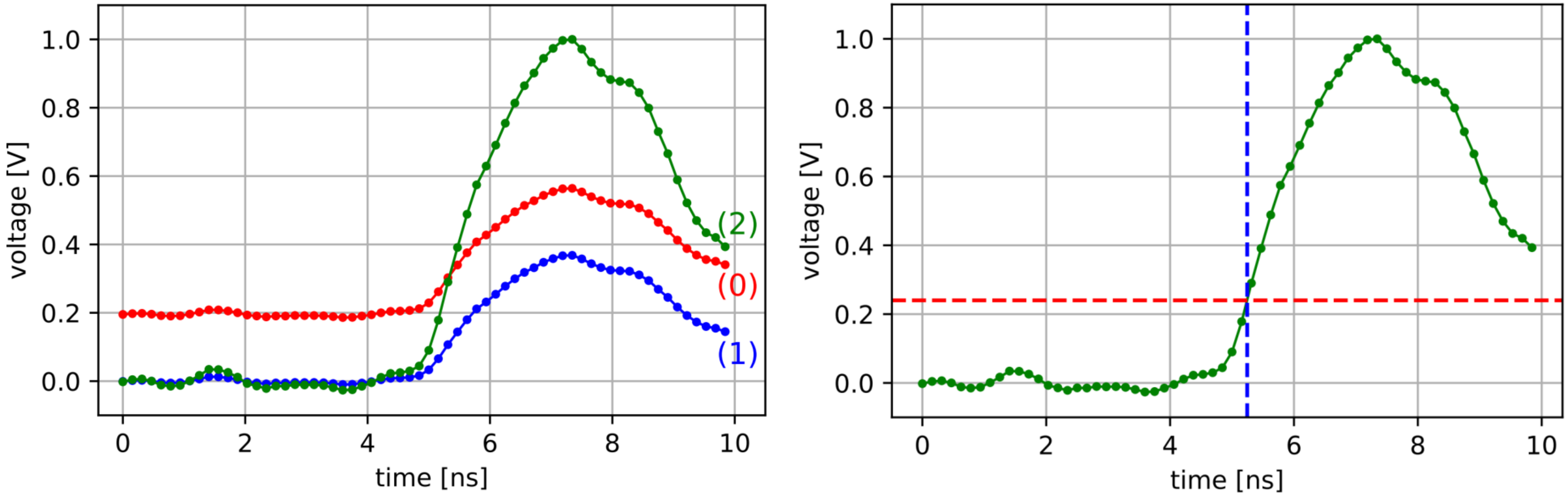
– Disadvantage: prone to the time walk effect



Example of the time walk effect. Although both signals reach their maximum at the same time, the threshold-crossing time is different.

Constant Fraction Discriminator

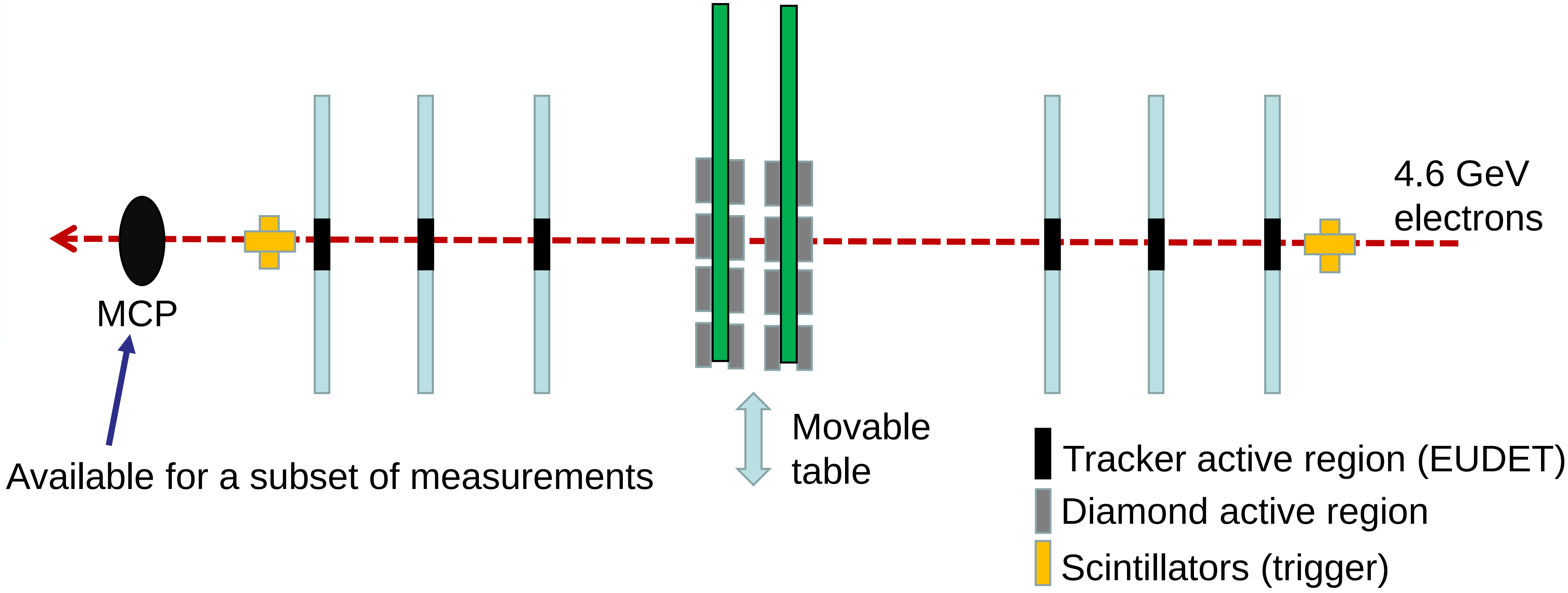
- The CFD algorithm (Constant Fraction Discriminator)
  - Method currently used in the CMS-PPS reconstruction
  - Goal: mitigation of the time walk effect
  - Implemented as the **normalised threshold algorithm** preceded by the baseline subtraction



The CFD algorithm. Left: (0) before normalisation, (1) baseline subtraction, (2) division by maximum. Right: after the normalisation the timestamp can be found using the fixed threshold algorithm

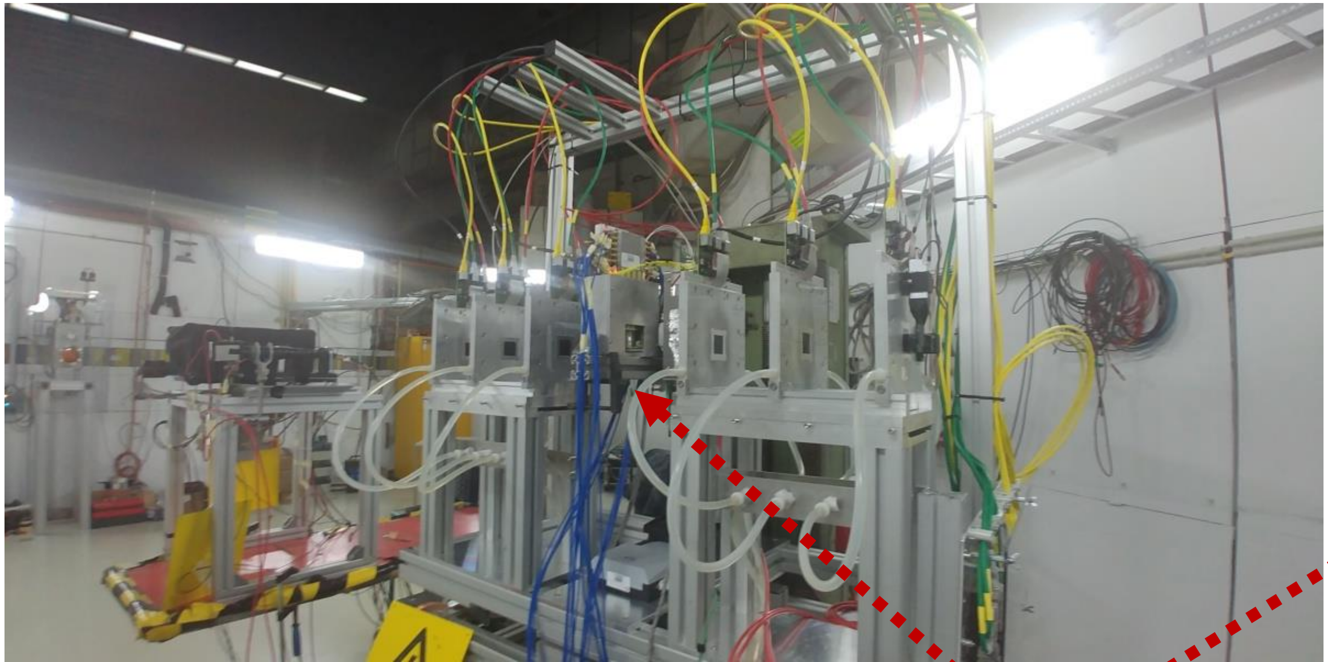
Detector setup (1)

- 2020 – test beam facility at the DESY-II synchrotron.
- Combined data taking with **diamond detectors** and a more precise **MCP-PMT** (MicroChannel Plate Photomultiplier Tube)



Test beam layout

Detector setup (2)

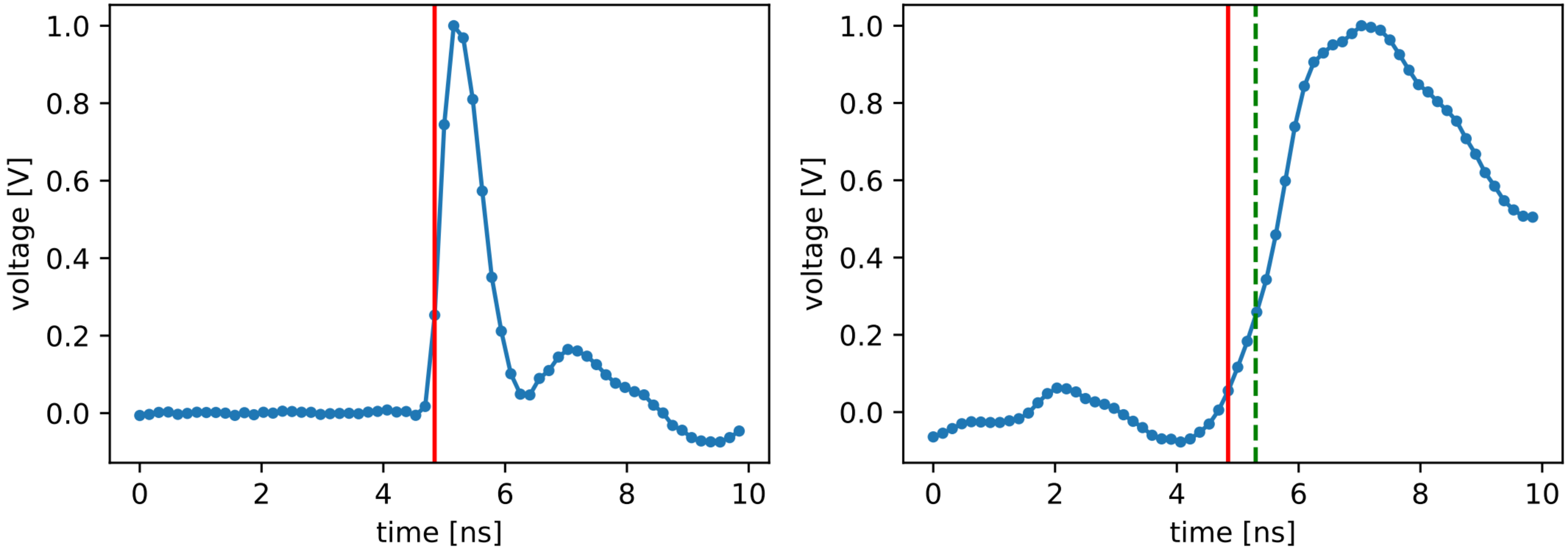


PPS timing sensors

Dataset

- The dataset needed to contain signals and the reference timestamps.
- Expected **diamond detector** precision: **50-100 ps**
- Expected **MCP-PMT** precision: **~10 ps**
- The reference timestamps (**ground-truth**) computed with the **CFD** using the **MCP-PMT** signals.
- Used only the events where a particle was detected both by a diamond detector and the MCP-PMT.
- **Goal for the neural network: minimise the difference between the predicted and ground-truth timestamps given a time series from the diamond detector.**

Dataset example



Dataset example. Left: an MCP signal with marked ground-truth timestamp. Right: a signal from a diamond detector; red: the ground-truth timestamp (includes the t_0 shift of both signals), green: the CFD timestamps computed on the diamond detector time series (used to compare the neural networks with CFD).

Neural networks

» Neural network – a machine learning algorithm modelled after the structure of the human brain.

» Made of **interconnected nodes (neurons)**, which process information.

  – Number of neurons (parameters) can reach millions or even billions.

» Used to recognise patterns in data, such as images, text or time series.

» A **neural network model** is trained on large datasets to make predictions on new data.

» **Training** – fitting the network to the data

  – Using a subset of the whole dataset – training set

» **Testing** – testing the network performance

  – Using the rest of the dataset – test set

  – Usually the training-test split is 80%-20%.

» Common testing approach: **cross-validation**

  – Divide the dataset into a few folds; test on one, train on the others.

Choosing the optimal architecture

» Tested architectures
  - Multilayer Perceptron (**MLP**)
  - Regular Convolutional Neural Network (**CNN**)
    - Devised to process images and time series.
  - **UNet**-based network
    - Devised to find keypoints or timestamps.

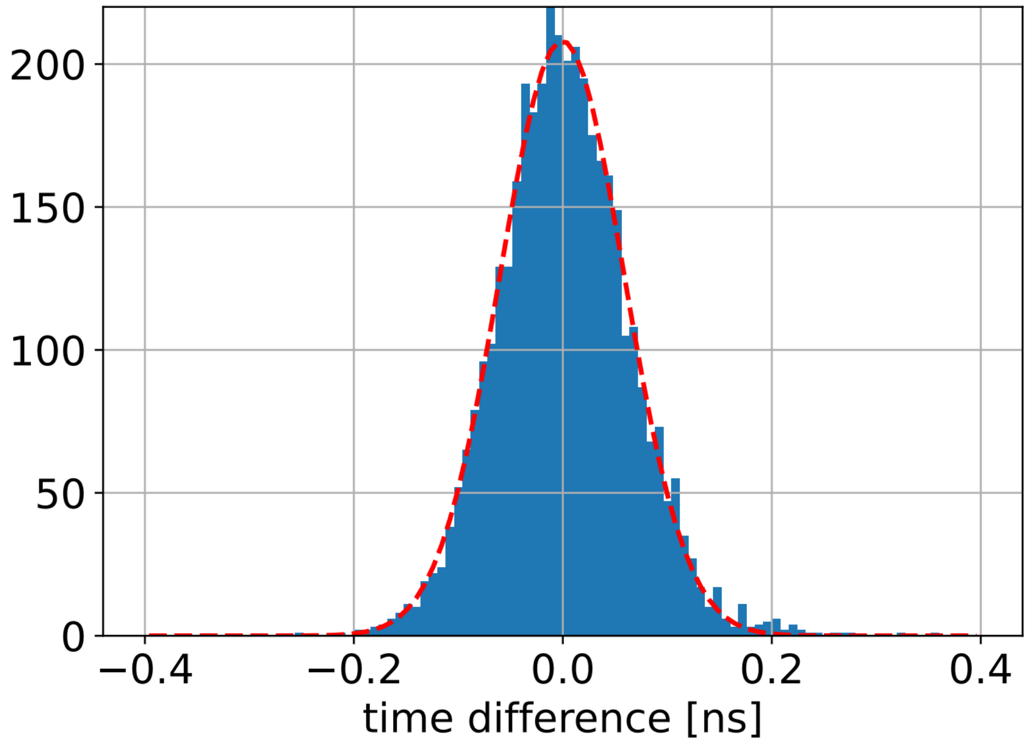
» Model selection done using **a two-step hyperparameter tuning procedure**.
  1. Find top five models using **keras-tuner** (a Python framework for TensorFlow).
  2. Use the cross-validation to find the optimal model.

» Following hyperparameters were optimised:
  - network depth,
  - number of neurons (dense layers), number of filters (convolutional layers),
  - Application of batch normalisation and/or dropout;

Precision assessment method

» Comparison with the "reference" detector – MCP
  – For each measurement: calculate the difference between the diamond det. and MCP.
  – Precision metric: std of differences

» A Gaussian can be fitted to the data to reduce the impact of outliers.
  – **Better precision metric: std of a Gaussian fitted to the difference histogram**



Example difference histogram with a fitted Gaussian

Optimal model selection

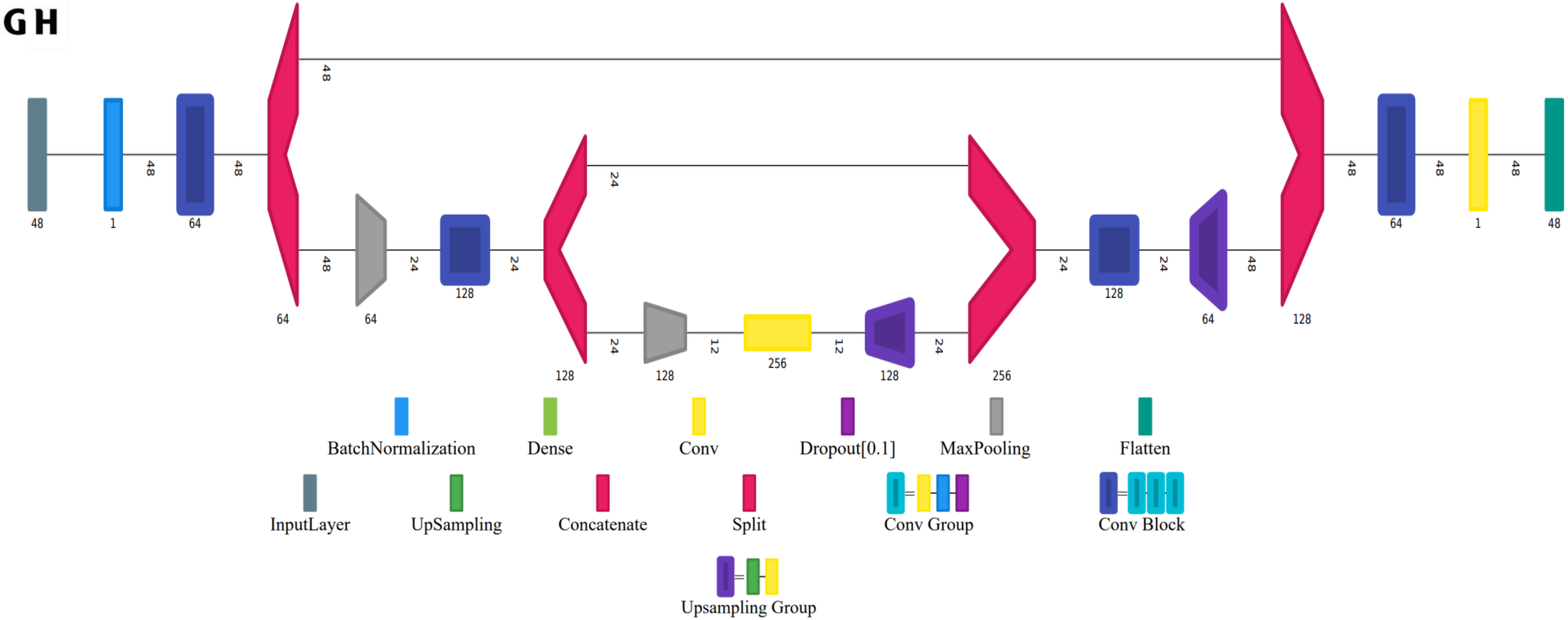
» Hyperparameter tuning used to find an optimal model for each architecture.

» Precision statistics computed through a **cross-validation** of the optimal models

architecture	mean [ps]	std [ps]	params
MLP	63.9	0.9	2,737
CNN	62.8	1.3	36,865
UNet	60.7	1.2	456,965

» **The best (smallest) precision: UNet**

Optimal UNet model

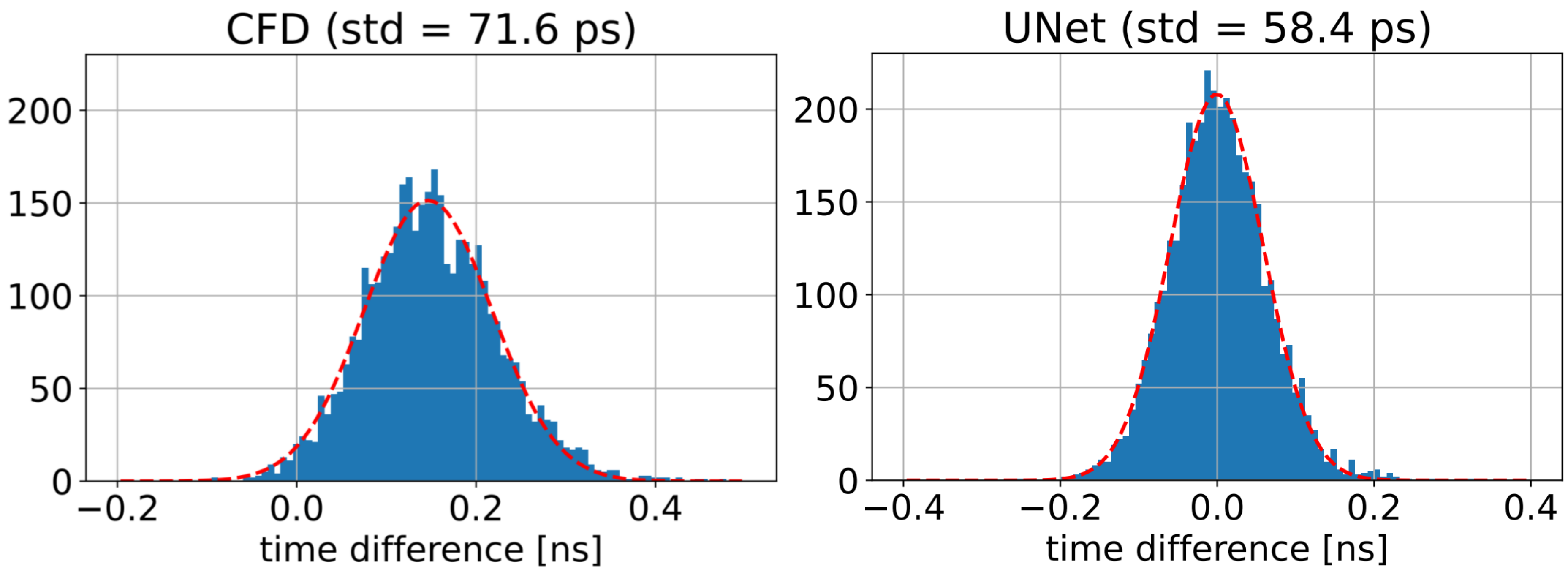


- Symmetric parts: encoder and decoder
- The encoder extracts time-independent features from a time series.
- The decoder builds a heatmap.
- The heatmap is expected to contain a Gaussian with the mean at the particle timestamp.
- The timestamp can be retrieved by applying a fit.

Results

- Final results obtained with the **test dataset** not used in the previous tests
- Precision comparison with CFD:

CFD	NN	Improvement
71.6 ps	59.4 ps	17.0%



Difference histograms with fitted Gaussians

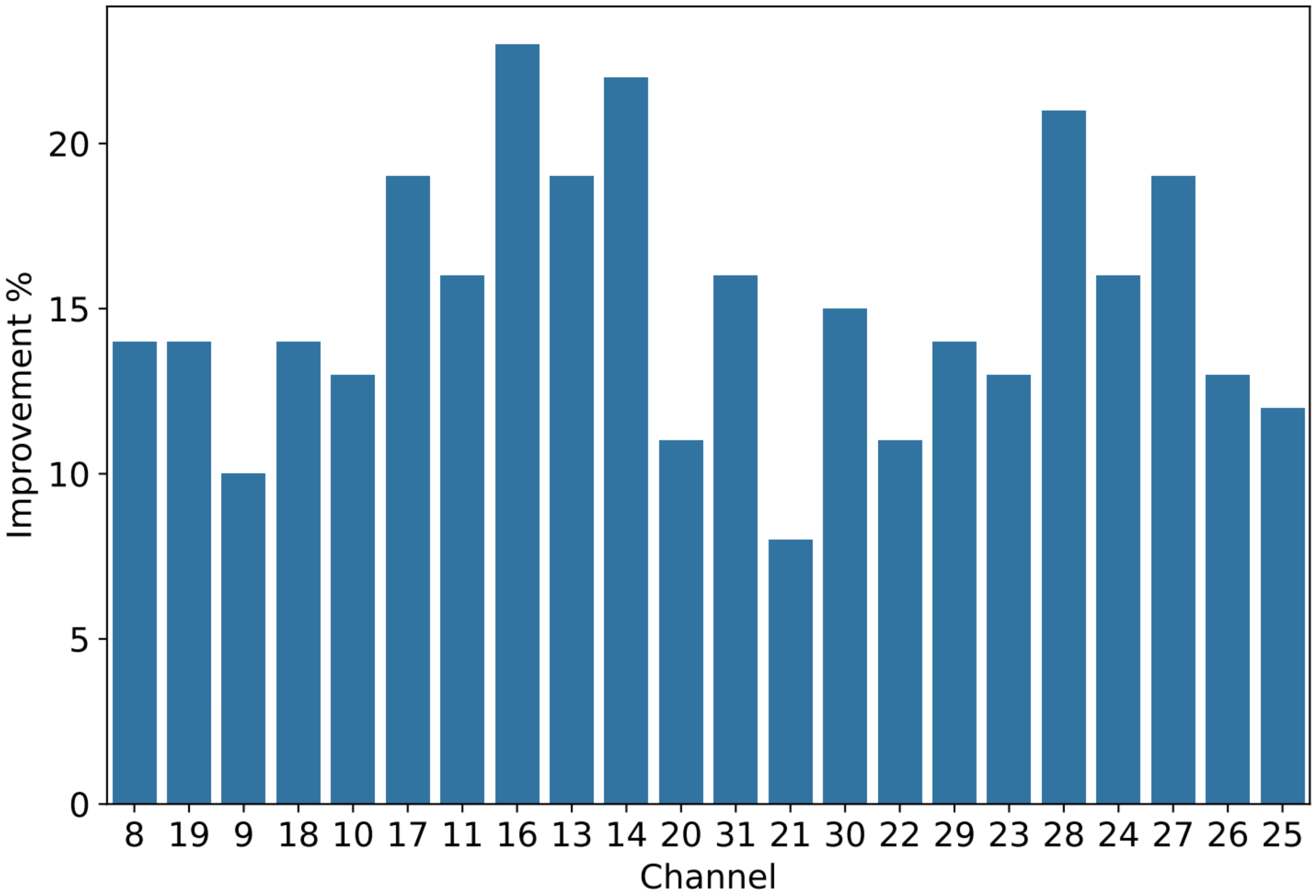
Results for many channels

» Networks trained either on single channels (representative examples) or on all the channels together (maintaining the train/test split).

» Improvements with respect to the CFD:

training channel	test channel					
	10	16	17	22	25	27
10	**13%**	10%	13%	7%	-1%	-23%
16	6%	**23%**	16%	9%	-22%	-9%
17	7%	17%	**19%**	9%	-3%	8%
22	4%	14%	-4%	11%	-84%	-51%
25	4%	4%	7%	4%	**12%**	8%
27	-13%	-10%	4%	-16%	4%	**19%**
all	8%	22%	14%	**12%**	9%	17%

Improvements for all the channels



Improvements with respect to the CFD for all the explored channels using the networks trained on particular channels. Improvements range from 8% to 23%.

Summary

» Improvements ranging 8% to 23% with respect to the CFD

» Advantages:

  – Network, once selected, has just to be trained and can work.

  – The expert knowledge is required only to find the optimal network model. Training and predicting is relatively simple.

  – In case the observed data evolves, the network can be easily retrained.

» Disadvantage: a neural network is a black-box

  – It is impossible or difficult to explain the network predictions.

» The work is continued on the LHC data.

The end

» The project was partially funded by the Polish Ministry of Education and Science, project 2022/WK/14.

» The numerical experiment was possible through computing allocation on the Ares system at ACC Cyfronet AGH under the grant plgccbmc11.