4th MODE workshop Valencia
23 september 2024

Reconstruction of p-p collisions with CMS using neuromorphic computing

Emanuele Coradin

Muhammad Awais, Fabio Cufino, Tommaso Dorigo,

Enrico Lupi, Jinu Raji, Fredrik Sandin, Mia Tosi

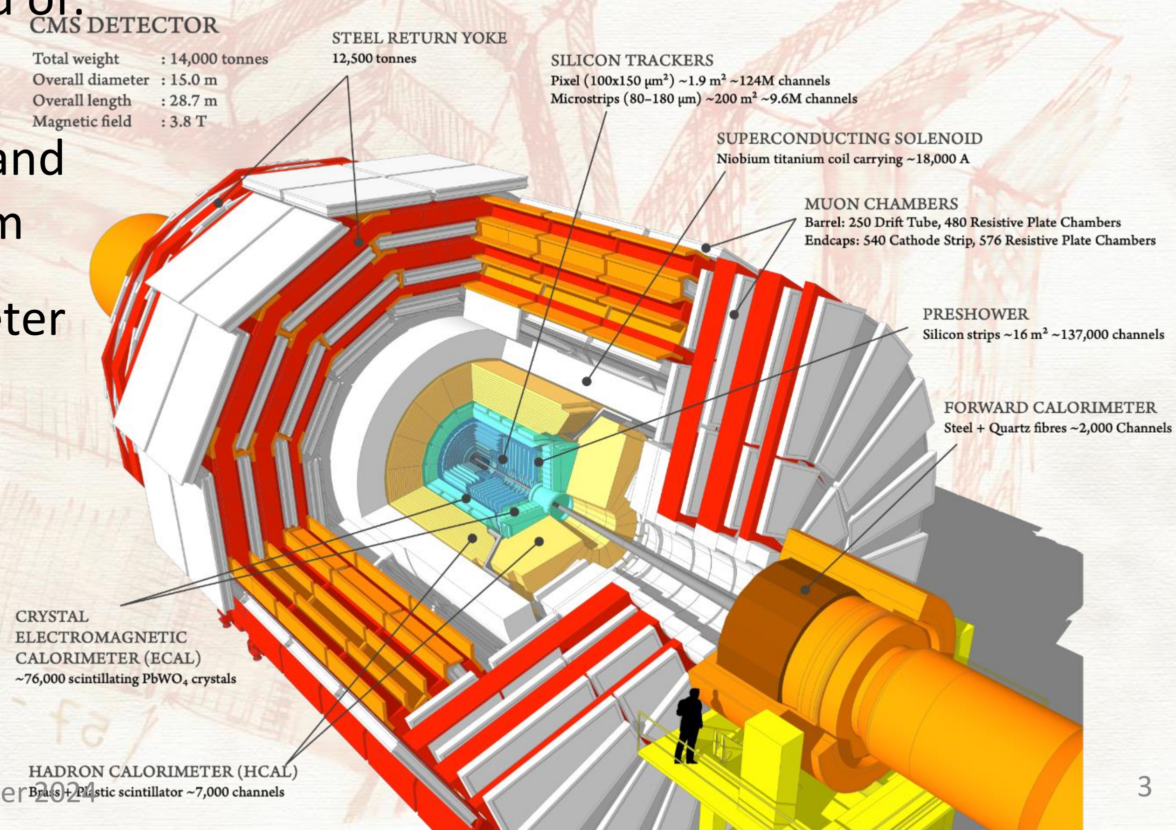
INFN, Universitá di Padova, IMAPP, Lulea University of Technology,
Universal Scientific Education and Research Network, Central University of Tamilnadu

Contents

Introduction: CMS

CMS detector

- “ Compact Muon Solenoid “
- Cylindrical structure composed of:
  - **Silicon tracker**:
    to track charge particles and measure their momentum
  - Electromagnetic calorimeter
  - Hadronic calorimeter
  - Superconducting magnet
    - $\vec{B} = (3.8\,T)\,\hat{z}$
  - Muon chambers



Introduction: CMS

Phase2 tracker

- Silicon detectors
  → excellent spatial resolution $O(\mu m)$
- Passage of a charged particle:

  → local charge accumulation (***cluster***)

  → 3D position (***hit***)

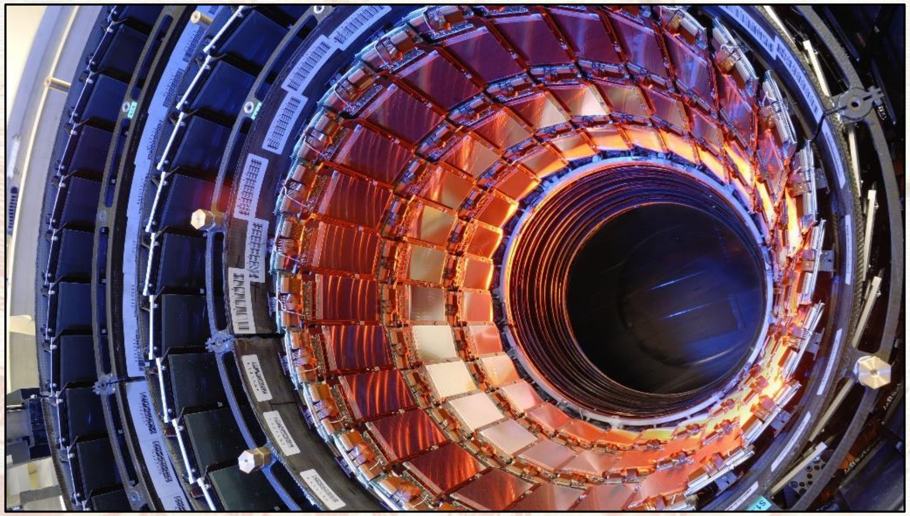
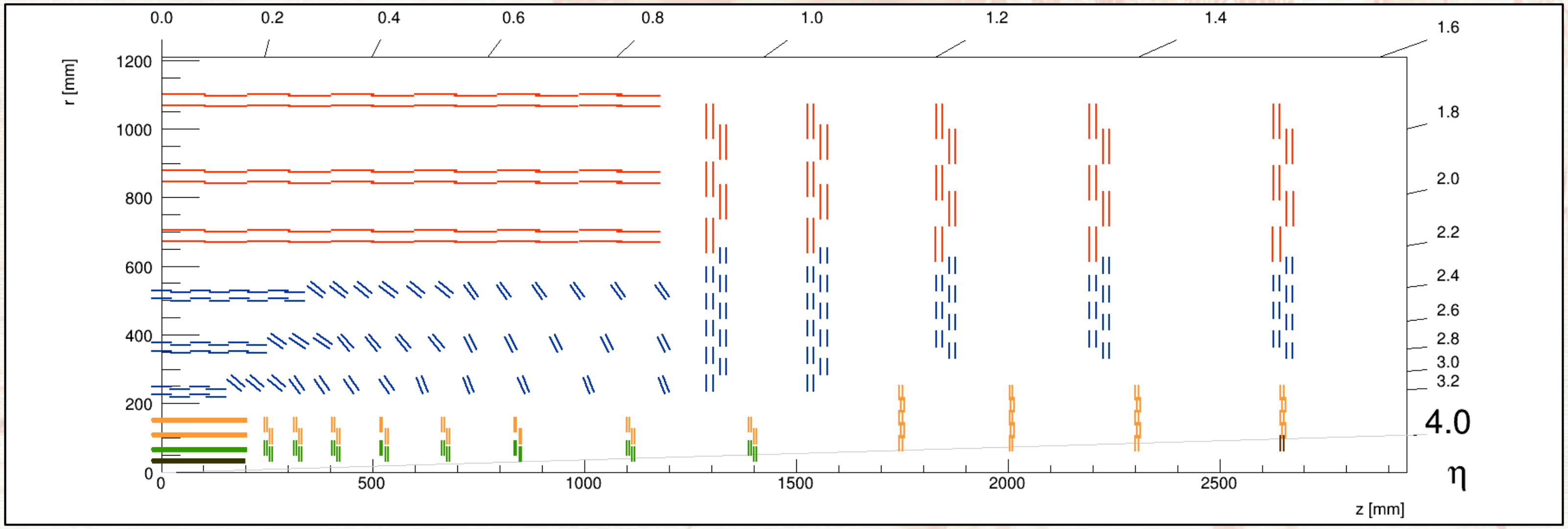


Photo: strip detector (Run 1, 2 e 3)



Sketch: Phase2 tracker layout

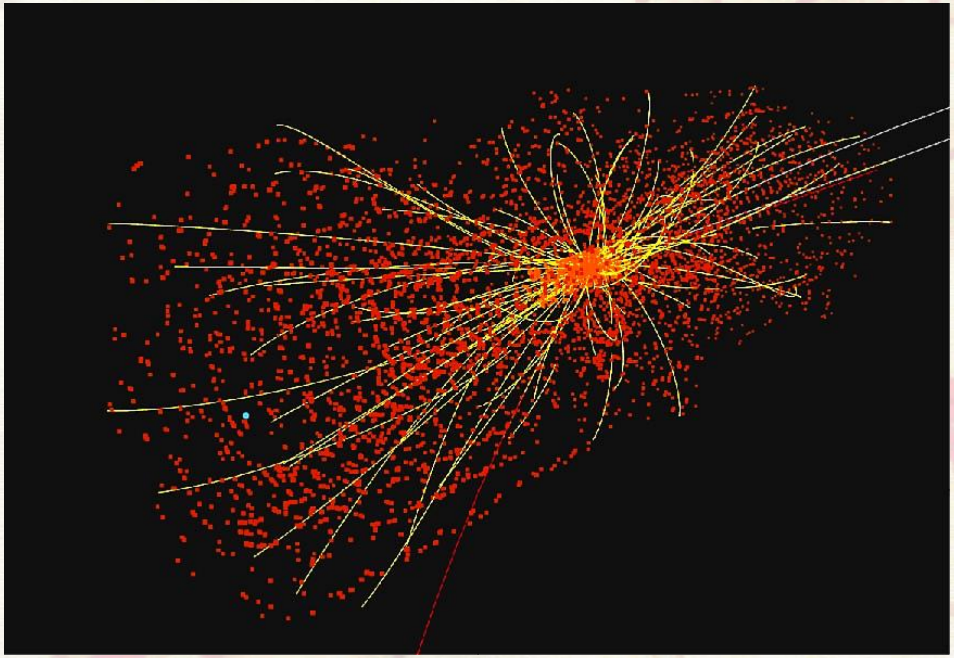
Tracking in dense environments

Proton-proton collision event revealed by CMS

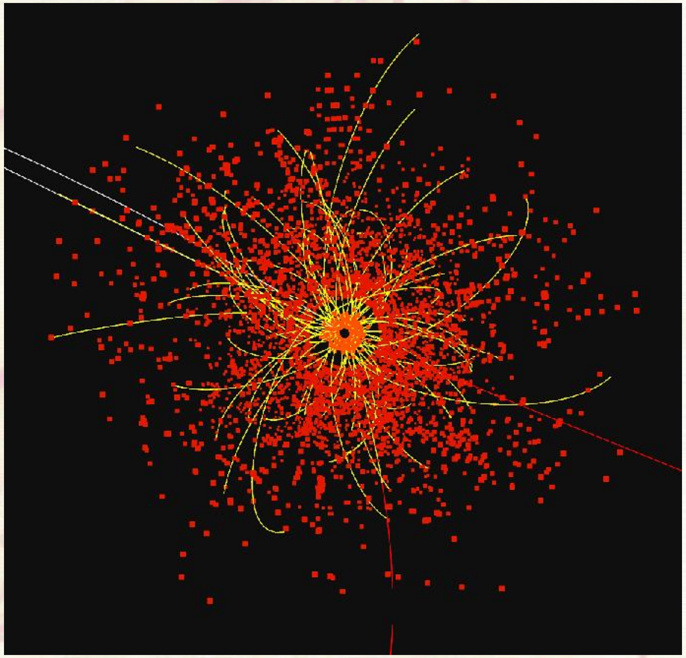
Hit on the detectors

Reconstructed track

The density will increase even further during the High Luminosity LHC!



3D vision of an event in the CMS tracker

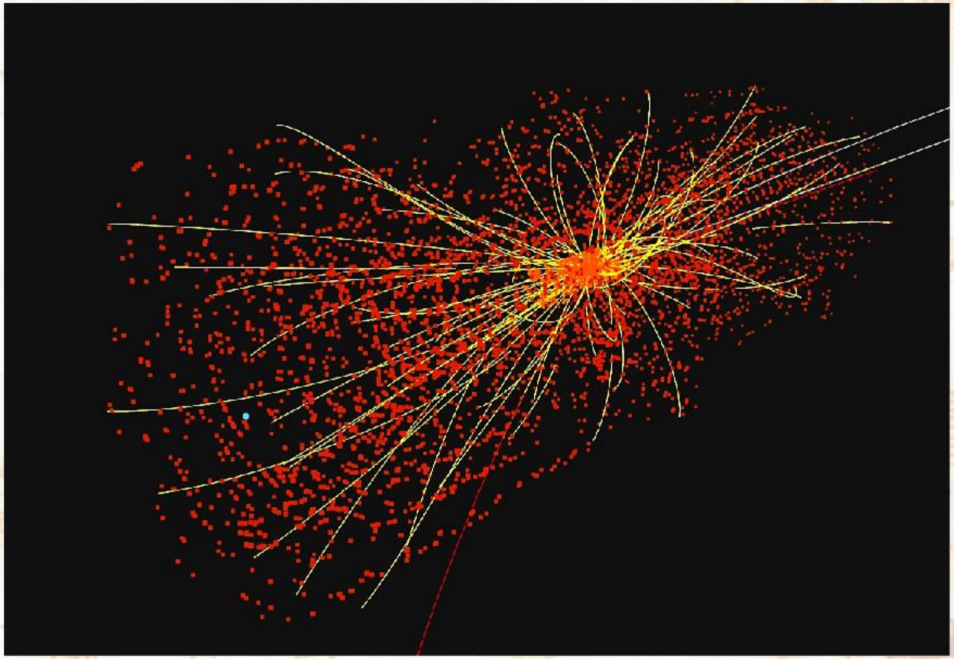


Transverse plane vision

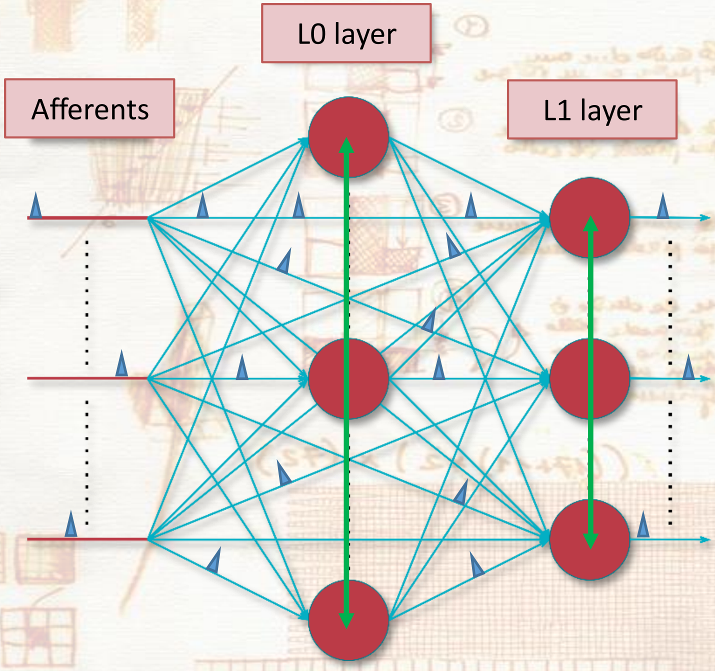
Goal

Identifying and reconstructing tracks is computationally complex problem

Innovative use of ***Spiking Neural Networks*** to complement *pattern recognition*



3D vision of an event in the CMS tracker



Spiking Neural Network architecture

Spiking neural network

What's neuromorphic computing

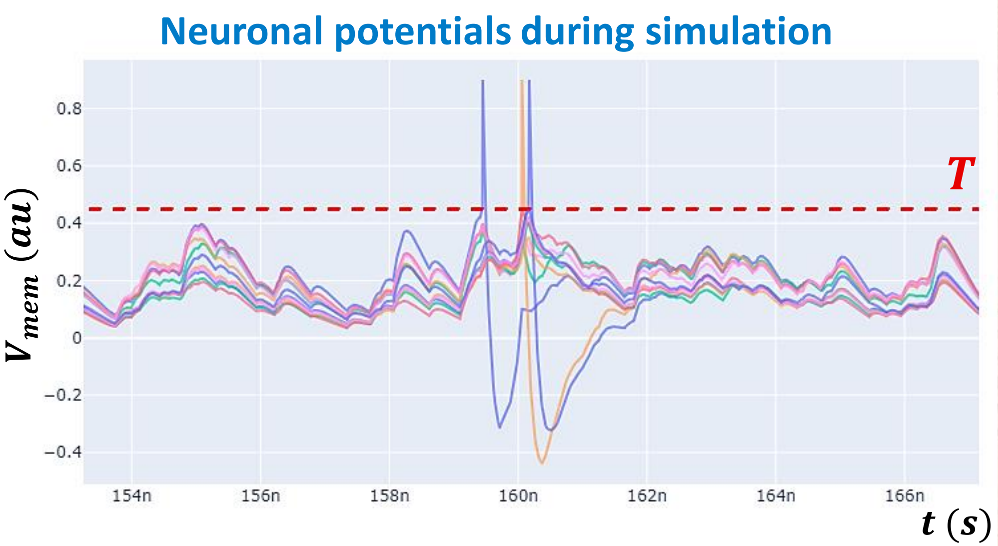
- **Spiking neural networks:** biologically plausible modelling of the behaviour of natural neural networks
- Implementing neurons and synapses in specialized hardware

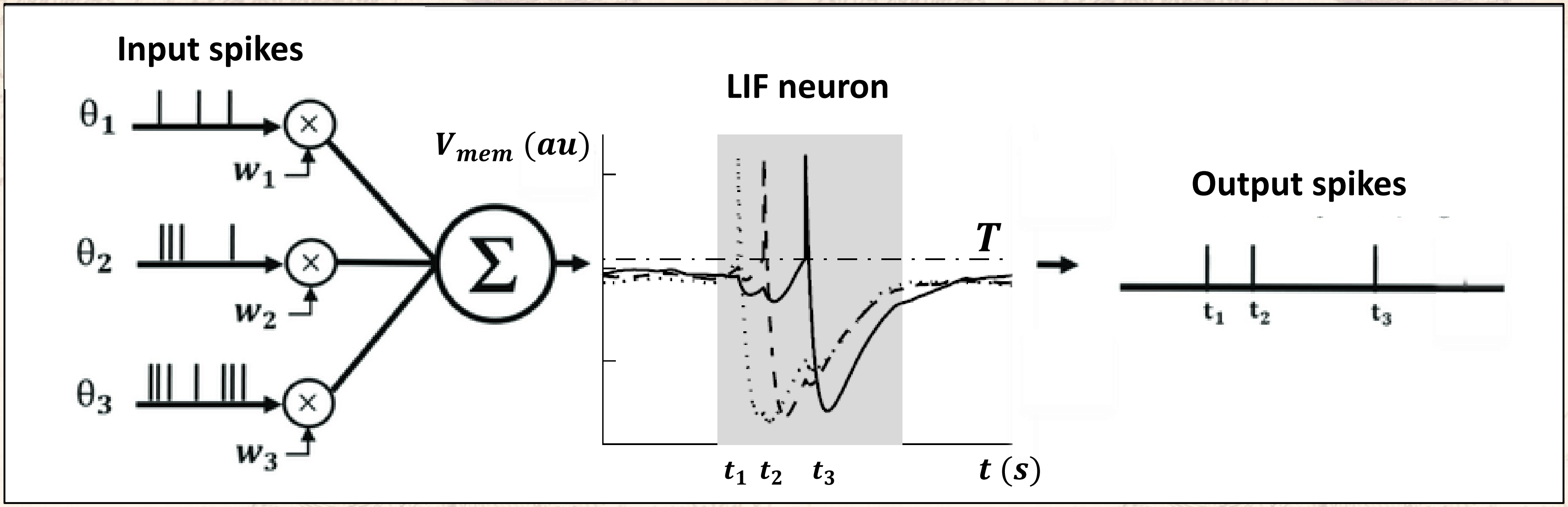
Distinctive features

- **Energy efficiency:** Operates with minimal power
- **Different computing paradigm:**
  - Both processing and memory are governed by the neurons and the synapses
  - **Event-driven computation**
  - Natural encoding of temporal information
- But usually harder to train

Leaky integrate-and-fire neuron (LIF)

- The **neuron** is modelled with an **RC circuit**
  - R → ion dispersion
  - C → membrane capacitance
- Information encoded in the arrival time of electrical impulses (*Spikes*)

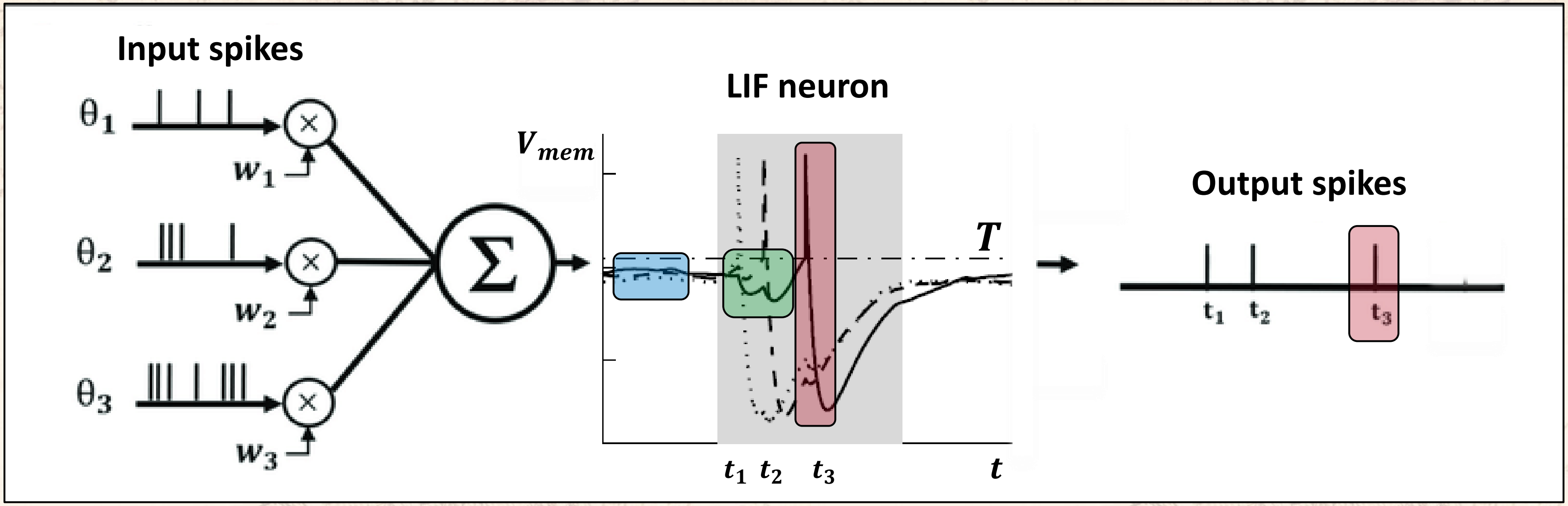




T. Masquelier et al., "Competitive STDP-based spike pattern learning", Neural Computation, may 2009

Leaky integrate-and-fire neuron (LIF)

- Input spike → **Increasing membrane potential**
- Membrane potential V_{mem} exceeds threshold **T** → **Neuron activation**
- **Inhibition** of neurons belonging to the **same layer**
  → Competition between neurons → **Specialization**

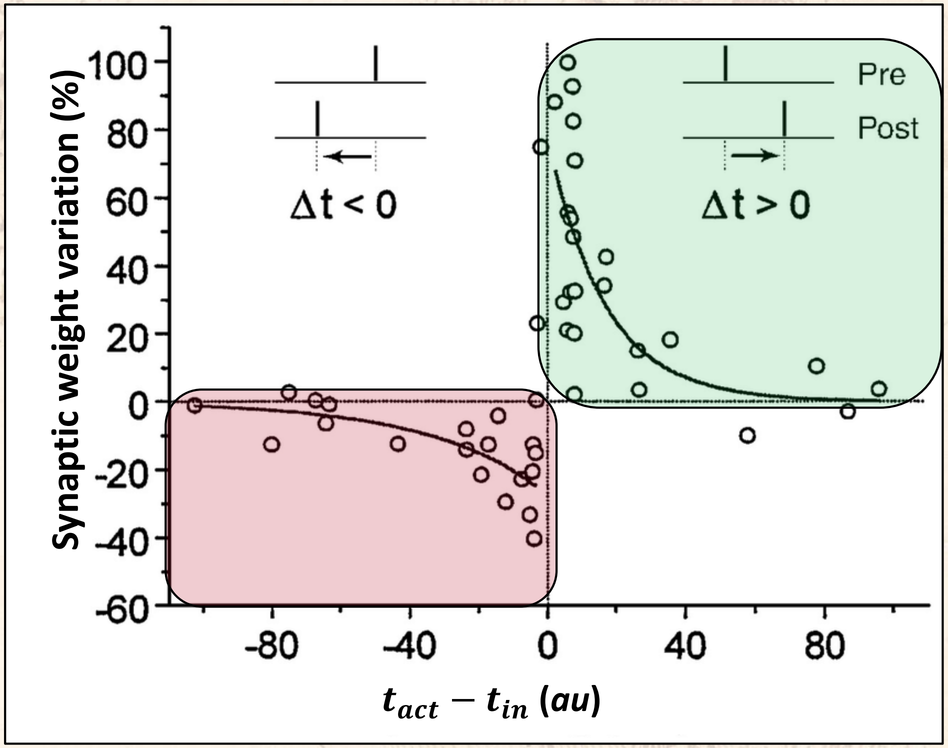


T. Masquelier et al., "Competitive STDP-based spike pattern learning", Neural Computation, may 2009

Learning algorithm: STDP

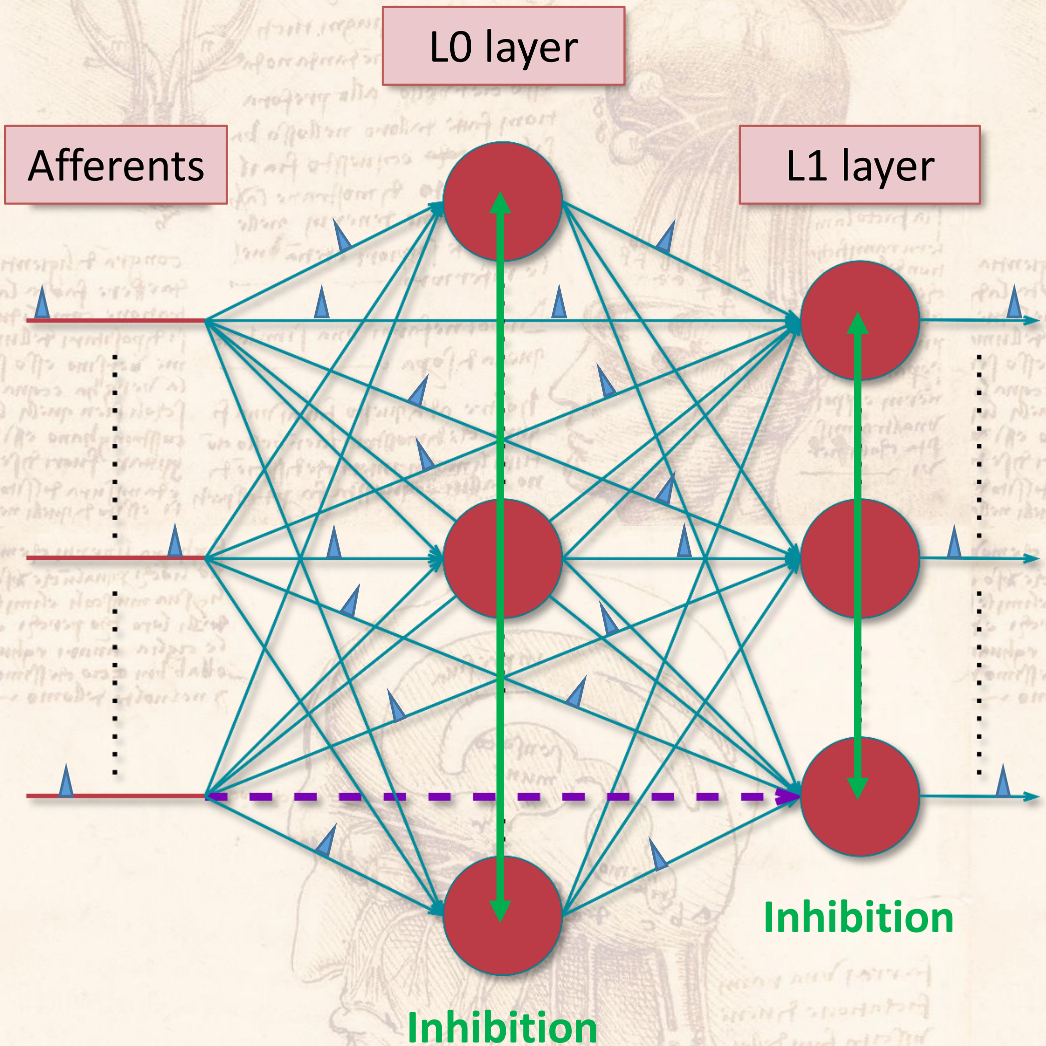
Spike timing-dependent plasticity

- Arrival **time** of **input** spikes ($\boldsymbol{t_{in}}$) and **activation time** of a neuron $\boldsymbol{t_{act}}$
- If $t_{in} < t_{act}$ → causal effect → potentiation of the synaptic weight
- If $t_{in} > t_{act}$ → anti-causal effect → depression of the synaptic weight



- Completely **unsupervised learning**
- Synaptic weights are adjusted just according to the relative time between the neuron activation and the arrival of input spikes

Network architecture

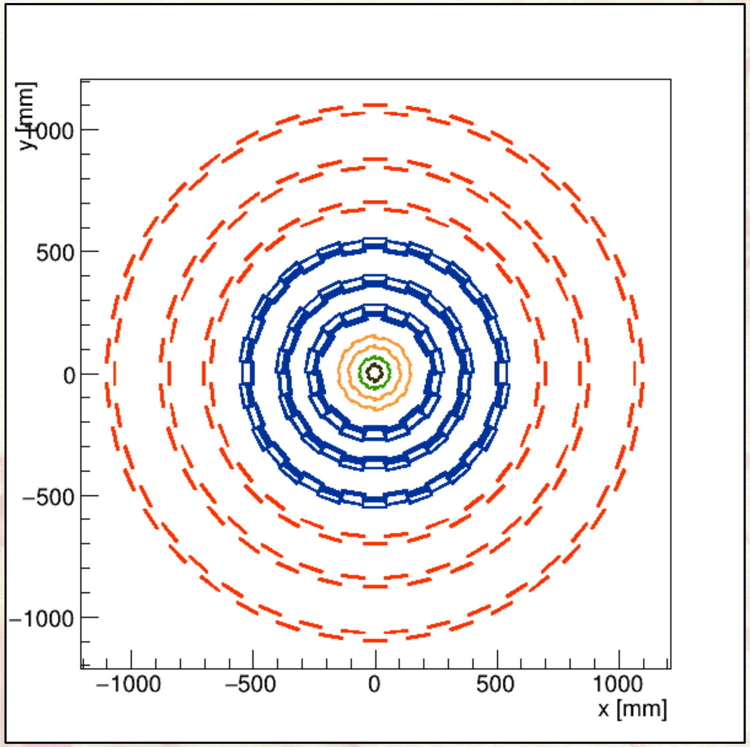


- **Afferents**: Fibres that carry electrical signals from the detector layers to the neural network
- **Two layers** of neurons **L0, L1** :
  - ➢ Possibility to eliminate connectons
- **20 hyperparameters to be optimize!**

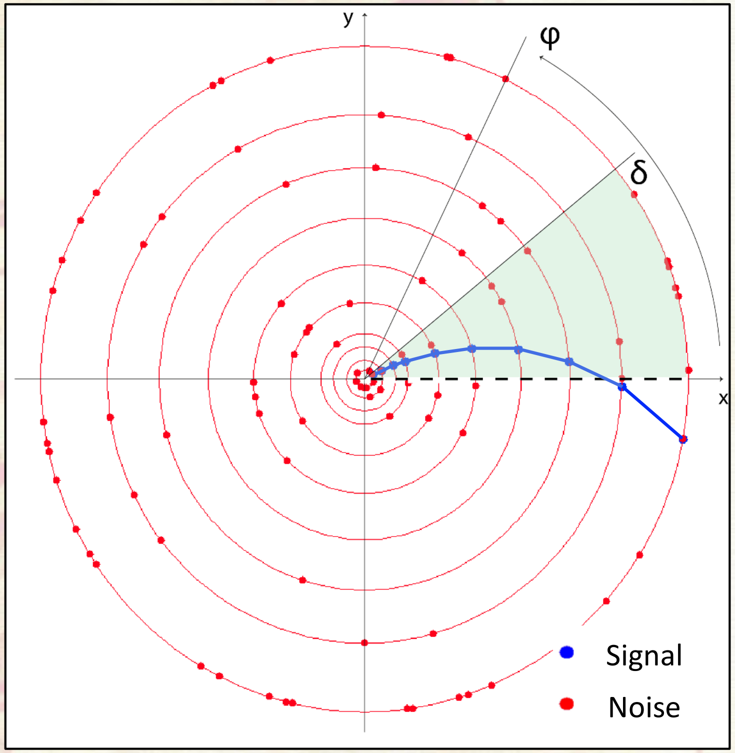
Information encoding

- Detection layer → Afferent
- Reading frequency: $f = 40\ MHz$
- Angular reading speed: $\omega = (2\pi + \delta) \cdot f$

- **Encoding time:** $\boldsymbol{t = \frac{\varphi}{\omega}}$
- $\delta = 0.7\ rad$ re-reading window to handle border effects

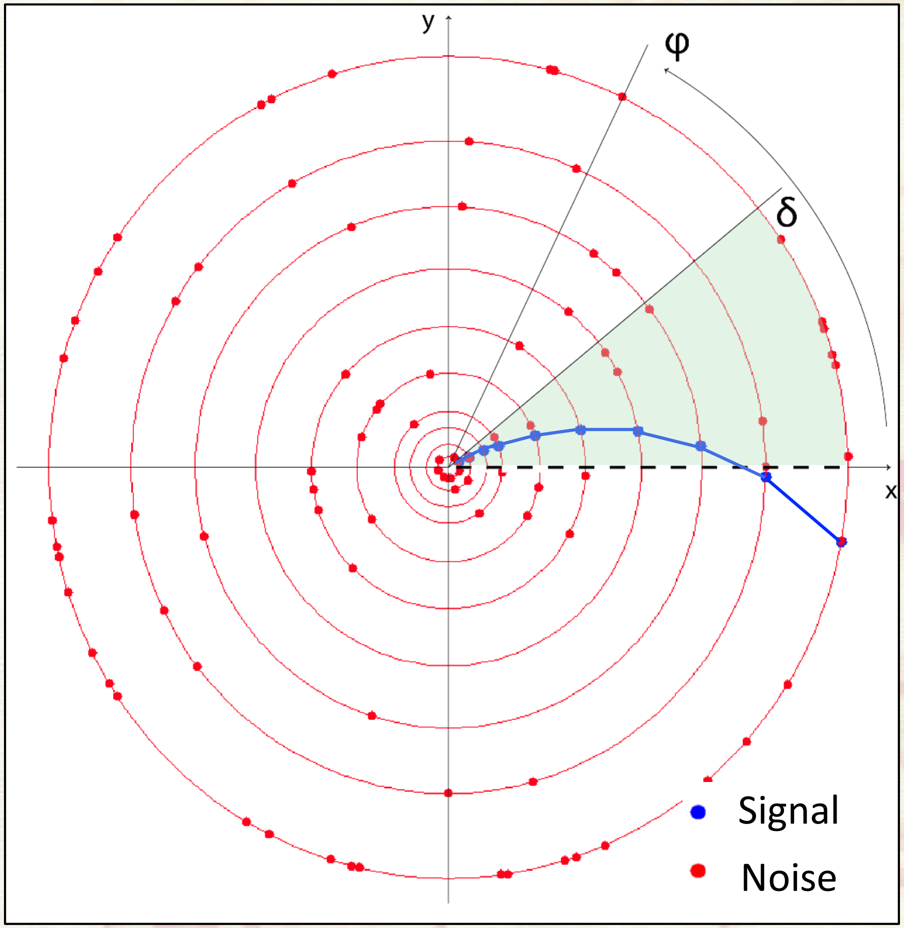


Sketch of the Phase2 tracker in the transverse plane

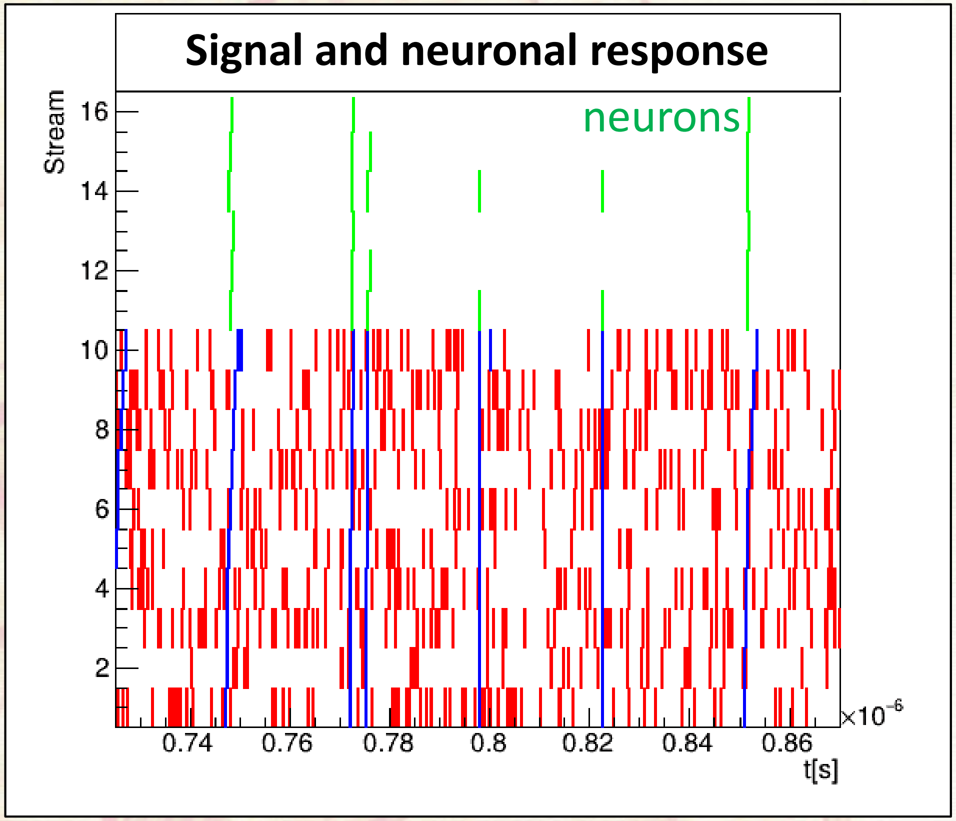


Simulated event, muon with $p_T = 1\ GeV$

Information encoding



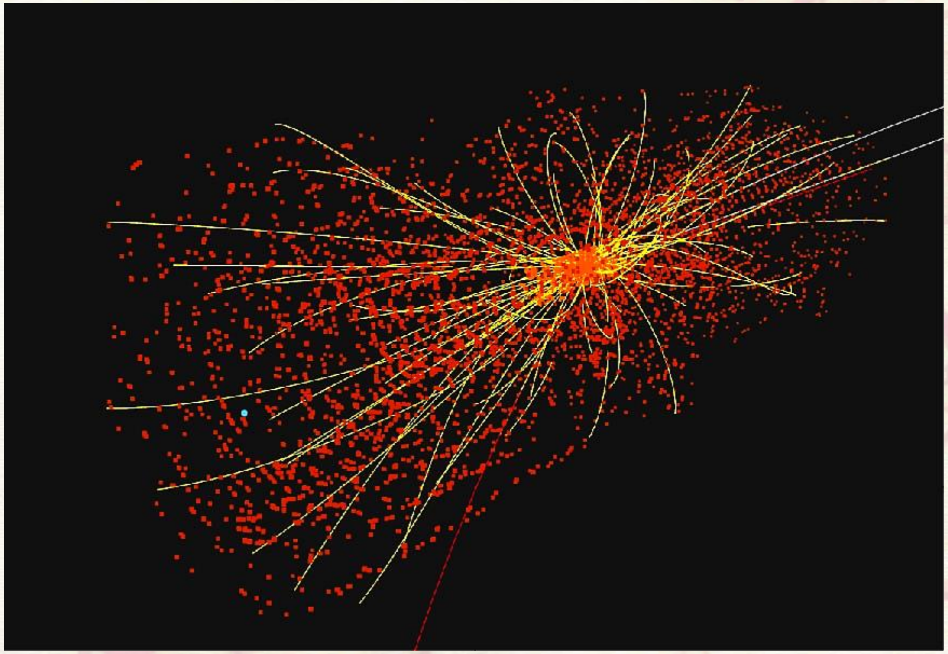
Simulated event, muon with $p_T = 1\ GeV$



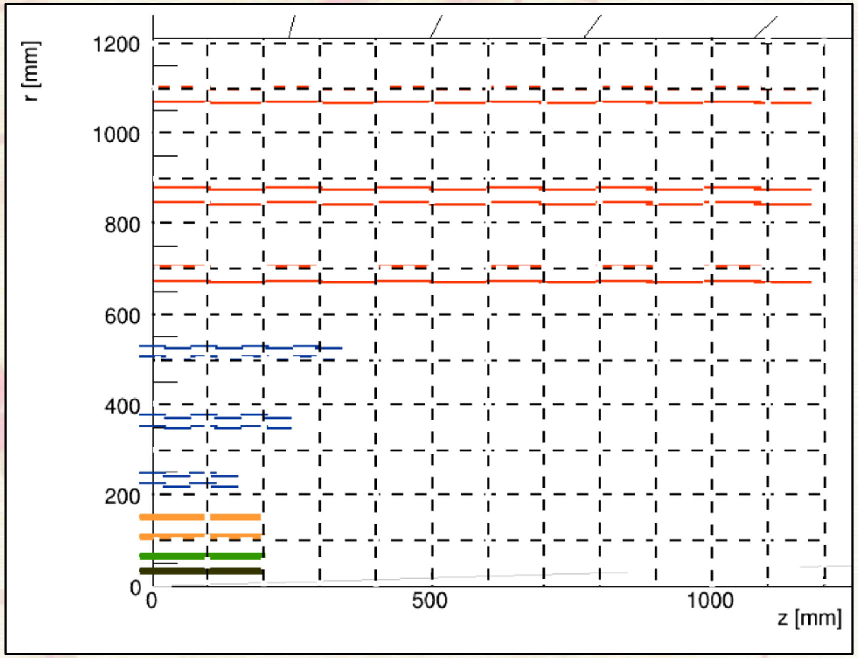
3D information encoding

- Restriction to the central area (*Barrel*)
- r sections: $N_r = 50$
- z sections: $N_z = 50$

- Afferents: $N_a = N_r \cdot N_z = 2500$
- Encoding all the information



3D vision of an event in the CMS tracker

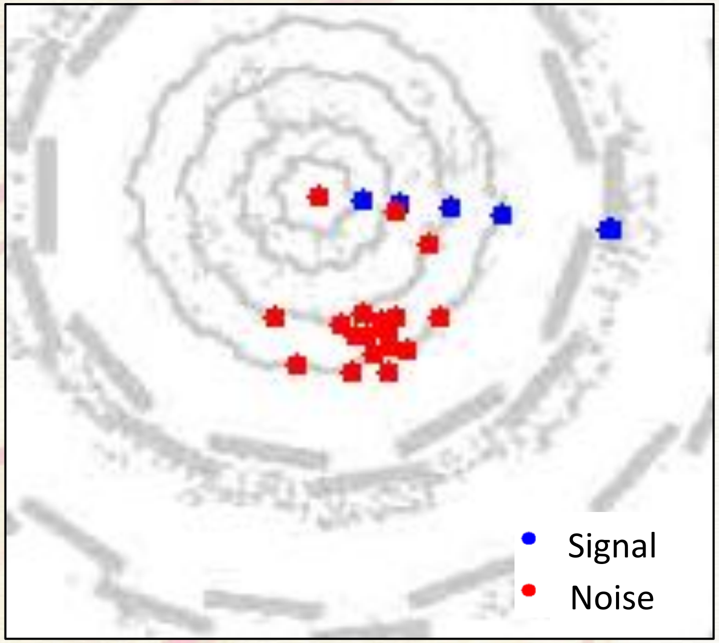


Sketch: layout of the tracker Barrel for Phase2

Datasets

Monte Carlo simulations

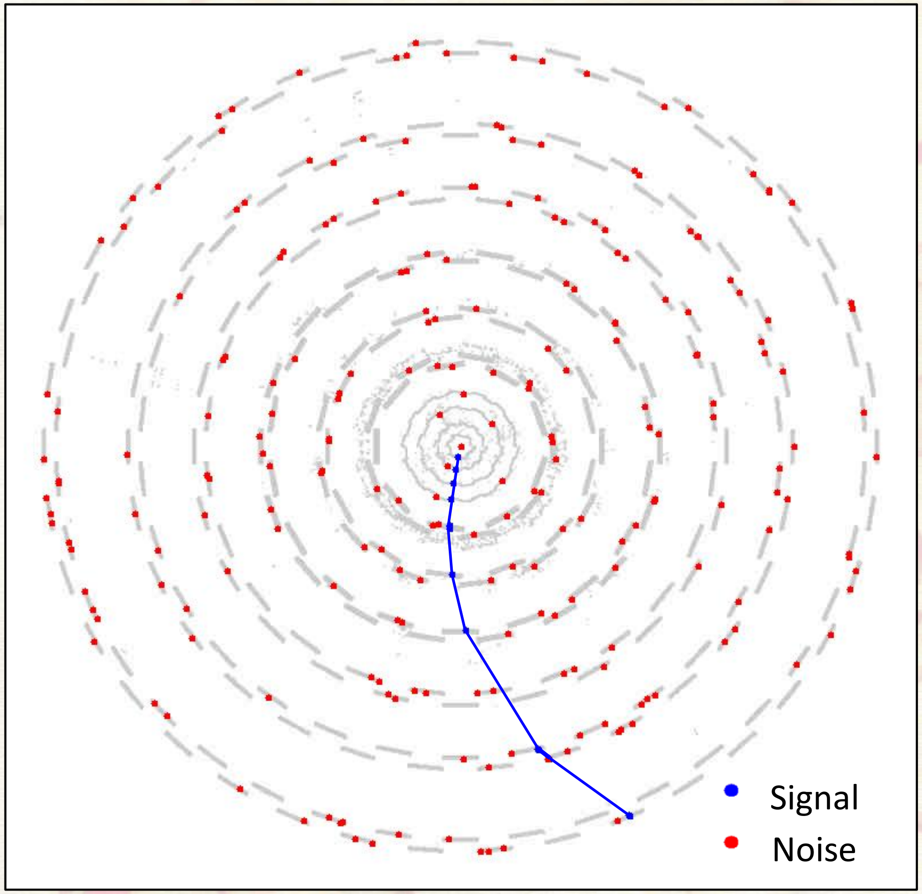
- **1 particle** per event:
  - Muons: $q = -1,\ p_T \in \{1, 3, 10\}\ GeV$
  - Antimuons: $q = +1,\ p_T \in \{1, 3, 10\}\ GeV$
- Contains some **interactions with the tracker material**



Detail of an event in which the impact with the tracker material caused the emission of an electron

Datasets

Background



Transverse plane projection of an event containing an antimuon, $p_T = 1\ GeV$ with $\bar{N}_{hit} = 200$

- We superimpose a Poissonian backround
- Background hits randomly extracted from the signal hits
- $\bar{N}_{hit} = 50, 100, 200$
- $N_{hit}^{signal} \sim 10$

3D model results

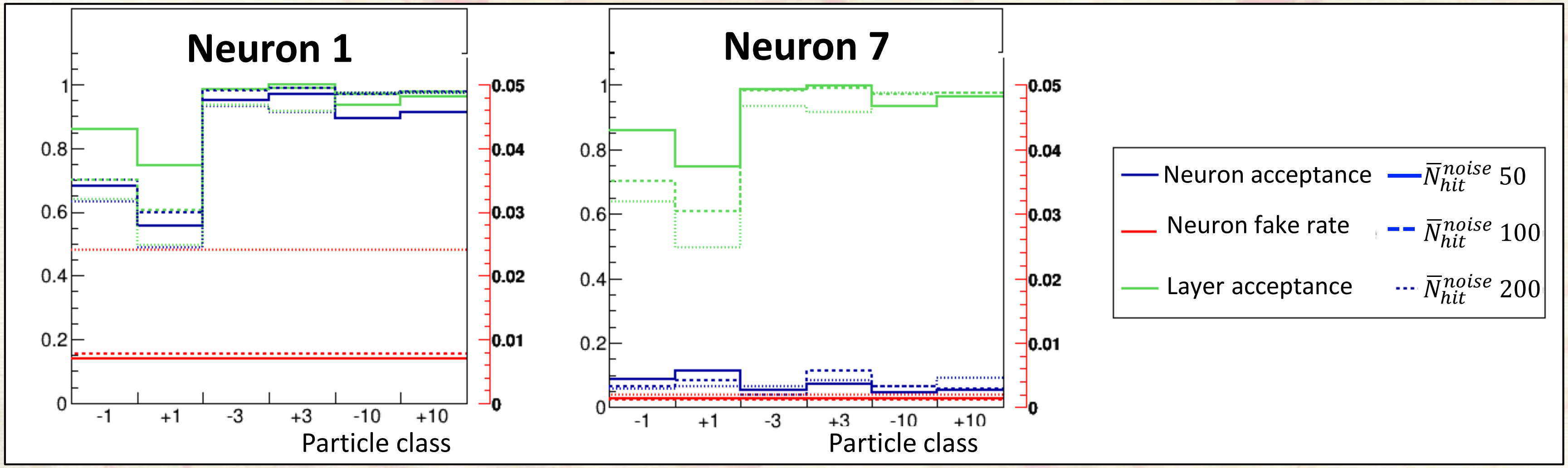
- Acceptance $A_{q,p_T} = \dfrac{\#Events(q,p_T)\ in\ which\ L1\ was\ activated}{\#Total\ events(q,p_T)}$
- Fake rate $E = \dfrac{\#Background\ events\ in\ which\ L1\ was\ activated}{\#Total\ background\ events}$

$\bar{N}_{hit}$	A_{-1}	A_{+1}	A_{-3}	A_{+3}	A_{-10}	A_{+10}	E
50	86.0%	75.7%	**98.6%**	**99.9%**	93.5%	96.4%	**2.2%**
100	70.1%	60.7%	98.4%	98.9%	97.0%	97.6%	**2.1%**
200	**63.9%**	**49.4%**	**93.5%**	**91.6%**	97.5%	97.4%	**3.9%**

✓ Better results in high p_T patterns

✓ Low fake rate < 5%

x Poor specialization of the neurons



Achievements and challenges

First implementation and proof of work of a **Spiking Neural Network** for the **identification** of particle trajectories produced in high-energy collisions

Successes

- ✓ The network **learns autonomously** to recognize tracks from noise
- ✓ Acceptance > 90% for particles with $p_T \geq 3\ GeV$
- ✓ Low false positive rate (<5%)

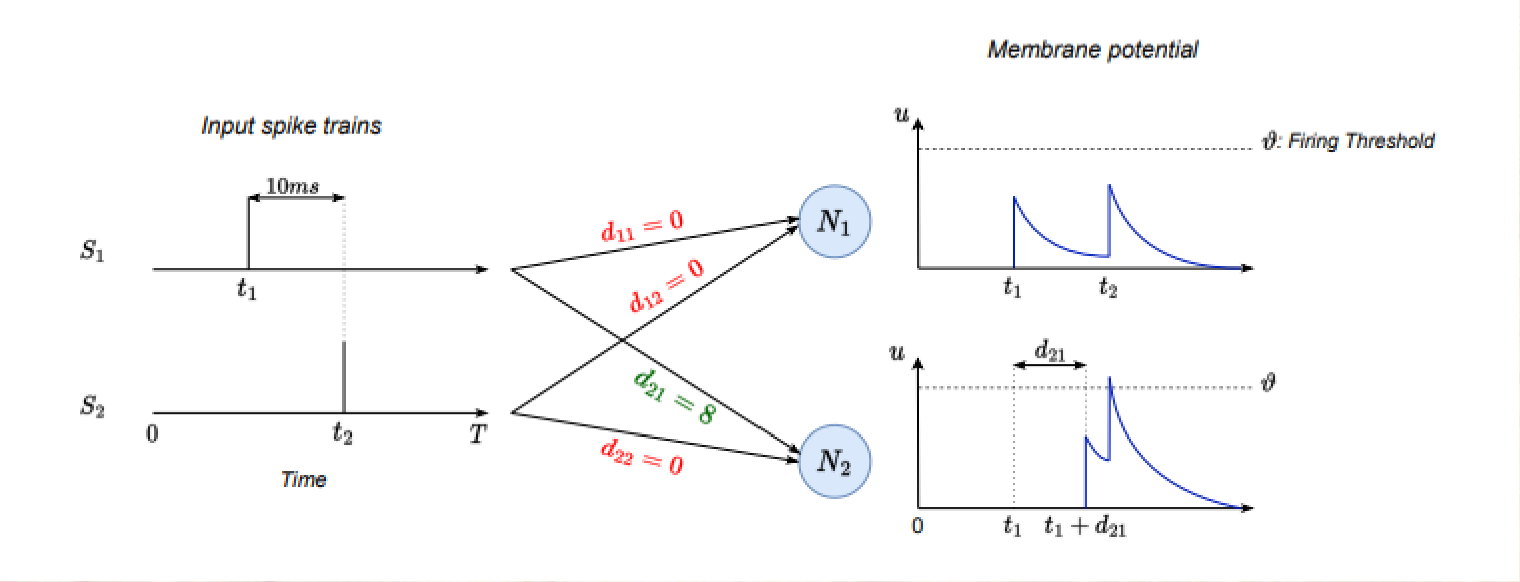
Limitations and future challenges

- x Poor neuron specialization
- x Multi-track event management
- x Difficult optimization of hyperparameters
- x Better management of 3D information

Active fields

Unsupervised delay learning

- Synaptic delays are another degree of freedom that we could exploit
- Delay adaptation to different signals → improve the specialization

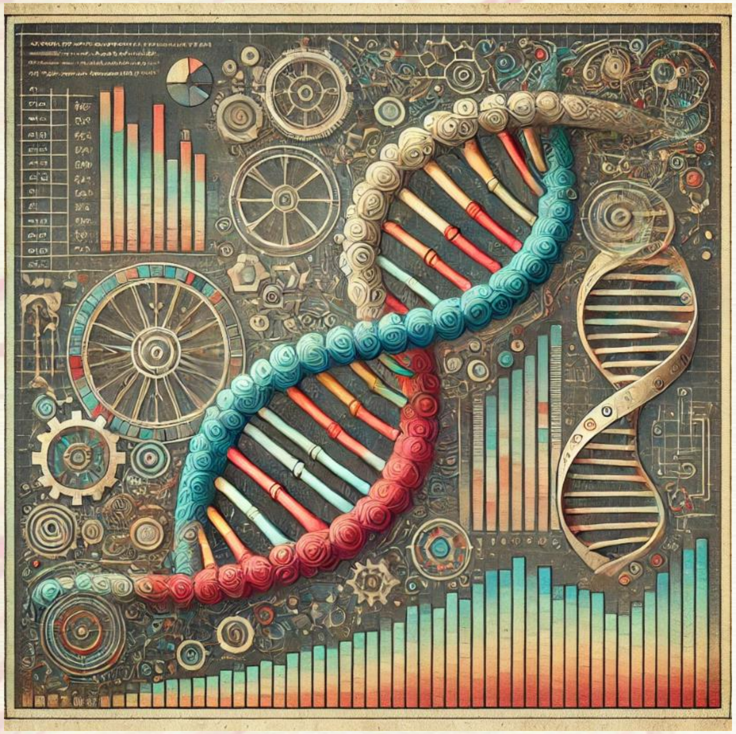


I. Hammouamri et al., "Learning Delays in Spiking Neural Networks using Dilated Convolutions with Learnable Spacings", arXiv preprint, 2023

Active fields

Genetic Algorithm

- Genetic algorithms are inspired by natural selection
- Application for hyperparameters tuning:
  - Can handle **large spaces** effectively.
  - Works well with **non-differentiable**, **discontinuous**, and **noisy** search spaces.
  - Can **avoid** getting stuck in **local minima**, unlike some traditional methods.



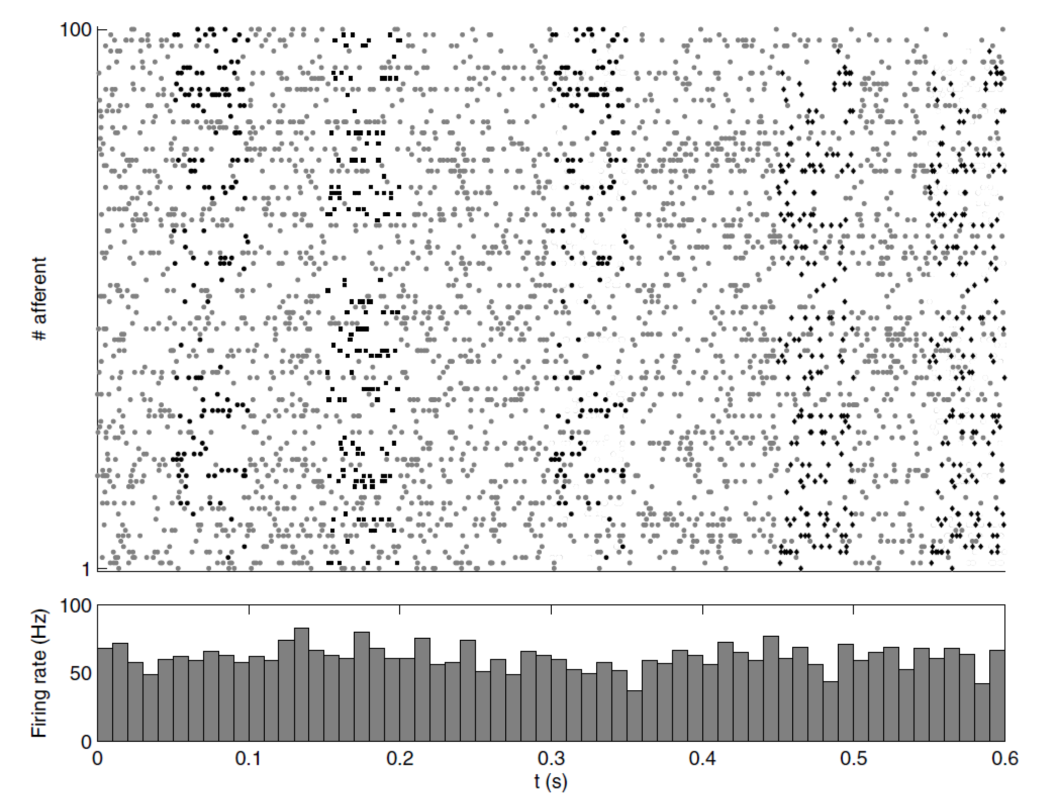
4th MODE workshop Valencia
23 september 2024

Thanks for your attention

Emanuele Coradin
Muhammad Awais, Fabio Cufino, Tommaso Dorigo,
Enrico Lupi, Jinu Raji, Fredrik Sandin, Mia Tosi

Spiking neural network

Reference model: Masquelier et al.



T. Masquelier et al., "Competitive STDP-based spike pattern learning", Neural Computation, may 2009

Technical features

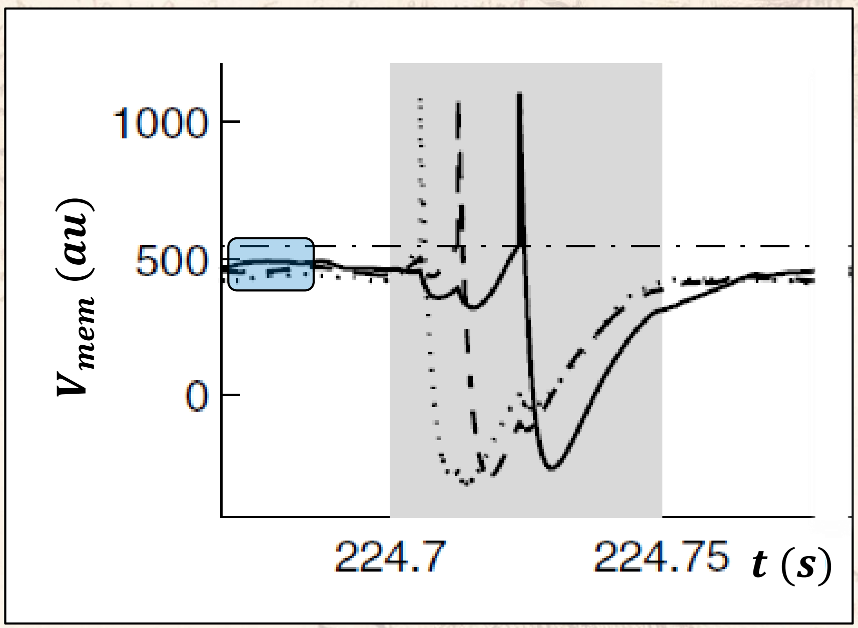
- Spike-timing-dependent plasticity
- Leaky integrate-and-fire neurons
- Single layer network

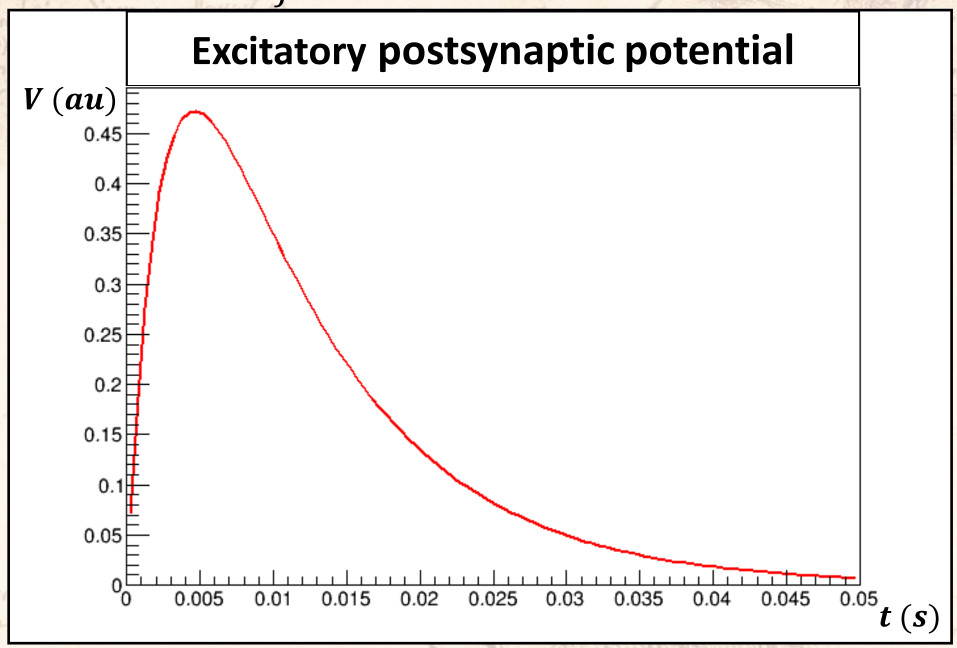
Achievements

- A simple SNN learns to **recognize complex patterns** in a **noisy environment**
- Poisson noise with embedded patterns
- **Patterns repeat randomly** in a **continuous regime**

Leaky integrate-and-fire neuron (LIF)

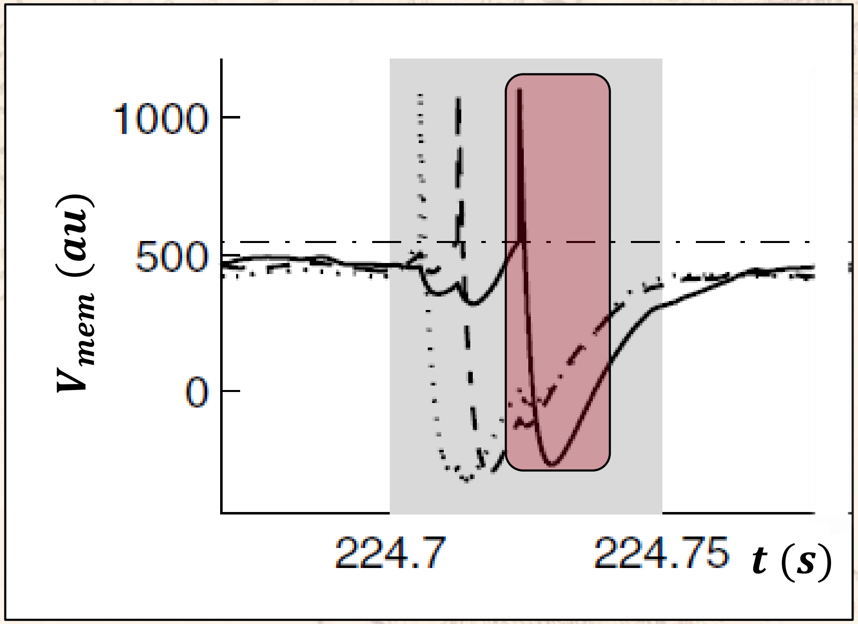
- Input pulse at the synapse → **Excitatory postsynaptic potential** (EPSP)
- $\varepsilon(t-t_j) = K \cdot \left[exp\left(-\frac{t-t_j}{\tau_m}\right) - \exp\left(-\frac{t-t_j}{\tau_s}\right)\right] \cdot \theta(t-t_j)$
- $\boldsymbol{\tau_m}$: Membrane characteristic time
- $\boldsymbol{\tau_s}$: Synapse characteristic time
- **K** multiplicative constant
- t_j pulse arrival time



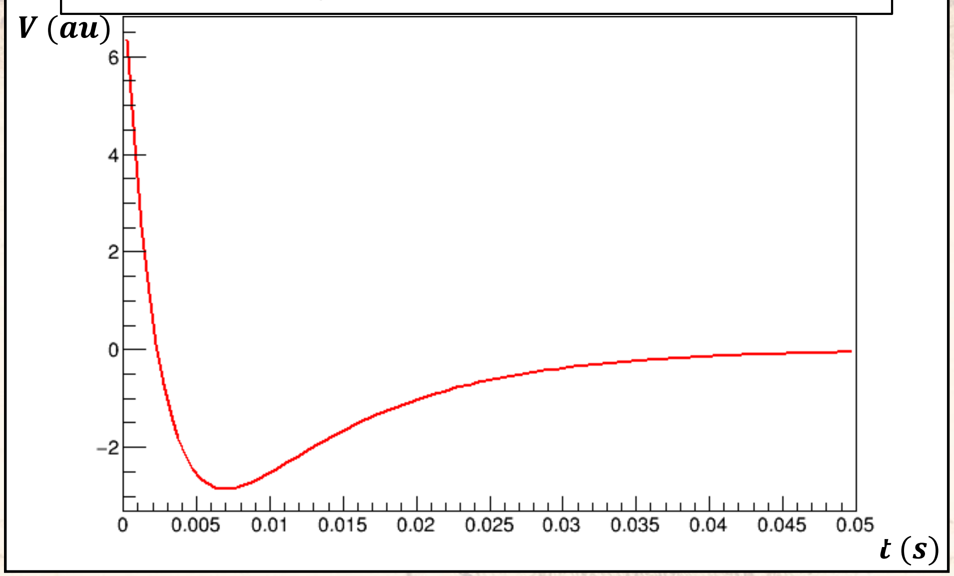


Leaky integrate-and-fire neuron (LIF)

- Potential exceeds threshold **T** → **Neuron activation** → output pulse
- $\eta(t-t_i) = T \cdot \left\{K_1 \cdot exp\left(-\frac{t-t_i}{\tau_m}\right) - K_2 \cdot \left[exp\left(-\frac{t-t_i}{\tau_m}\right) - exp\left(-\frac{t-t_i}{\tau_s}\right)\right]\right\} \cdot \theta(t-t_i)$
- $\boldsymbol{\tau_m}$: Membrane characteristic time
- $\boldsymbol{\tau_s}$: Synapse characteristic time
- $\boldsymbol{K_1, K_2}$ multiplicative shape constants
- t_i activation time



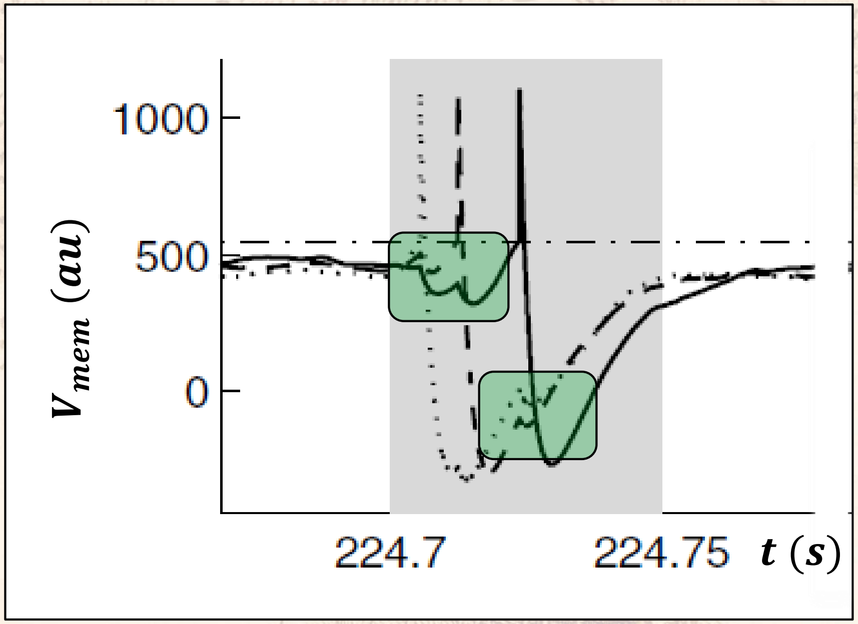
Membrane potential after the activation

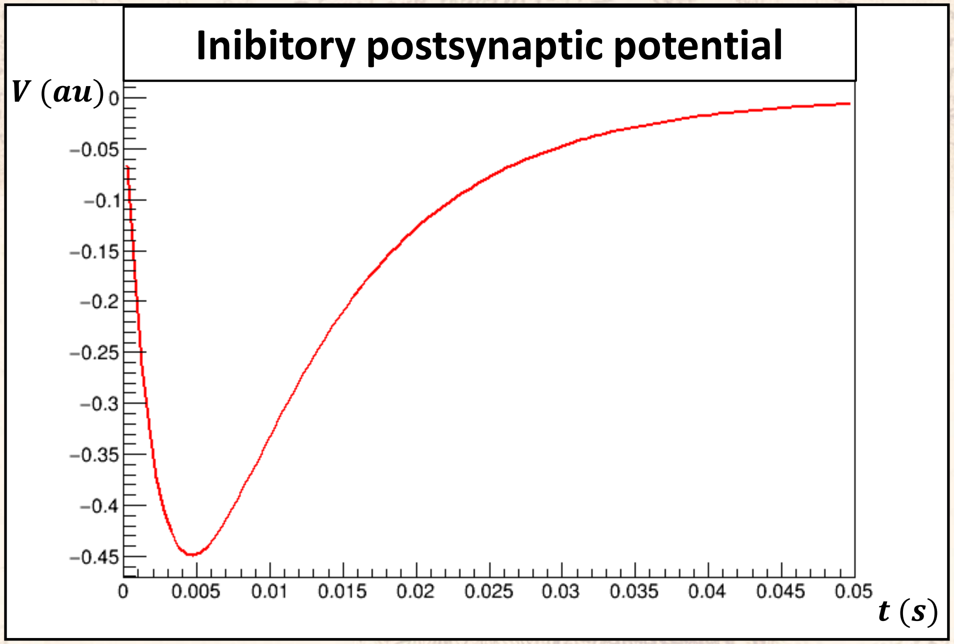


Leaky integrate-and-fire neuron (LIF)

- Neuron activates → It inhibits neurons in its layer → **Inhibitory postsynaptic potential** (IPSP)
- $\mu(t - t_k) = -\alpha \cdot T \cdot \varepsilon(t - t_k)$
- $\boldsymbol{\alpha}$ multiplicative intensity constant

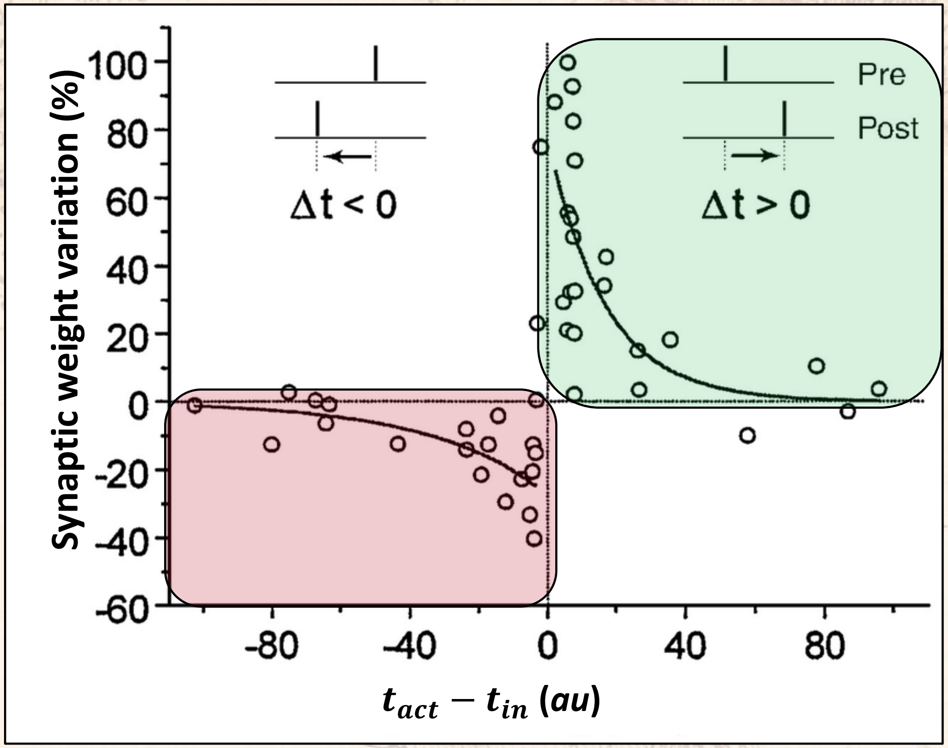
Competition among neurons





Learning algorithm: STDP

- **Spike timing-dependent plasticity** (STDP):
- Synaptic weights are adjusted only according to the arrival **time** of **input** spikes ($\boldsymbol{t_{in}}$) and the **activation time** of a neuron $\boldsymbol{t_{act}}$
- If $t_{in} < t_{act}$ → casual effect → potentiation of the synaptic weight
- If $t_{in} > t_{act}$ → anti-causal effect → depression of the synaptic weight



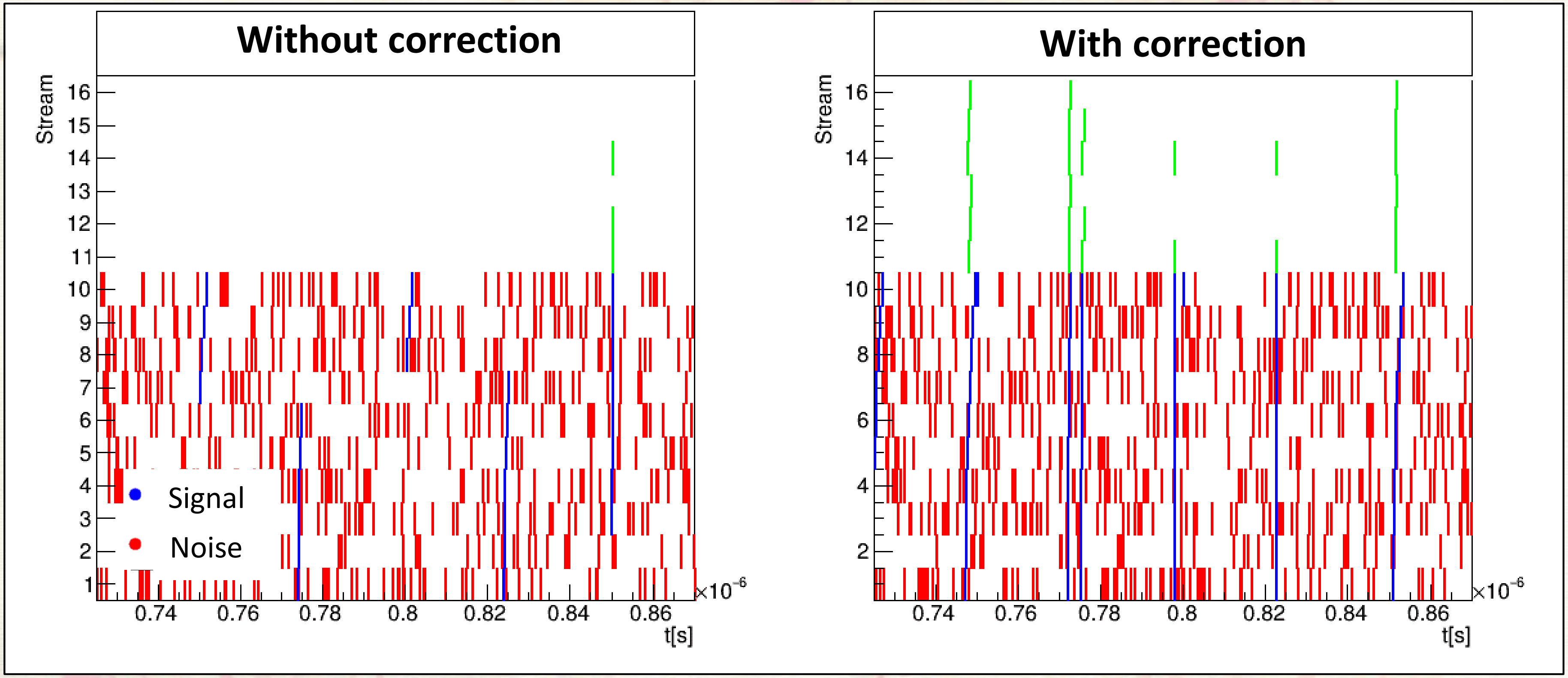
- Completely **unsupervised learning**
- Parameters:
  - τ^+, τ^- Characteristic times
  - a^+, a^- Learning constants

$$\Delta\omega = \begin{cases} \boldsymbol{a}^+ \cdot exp\left(\frac{t_{act} - t_{in}}{\boldsymbol{\tau}^+}\right) & if\ t_{act} \le t_{in} \\ -\boldsymbol{a}^- \cdot exp\left(-\frac{t_{act} - t_{in}}{\boldsymbol{\tau}^-}\right) & if\ t_{act} > t_{in} \end{cases}$$

Border effects

- Discontinuous signal encoding
- Increase in learning difficulties

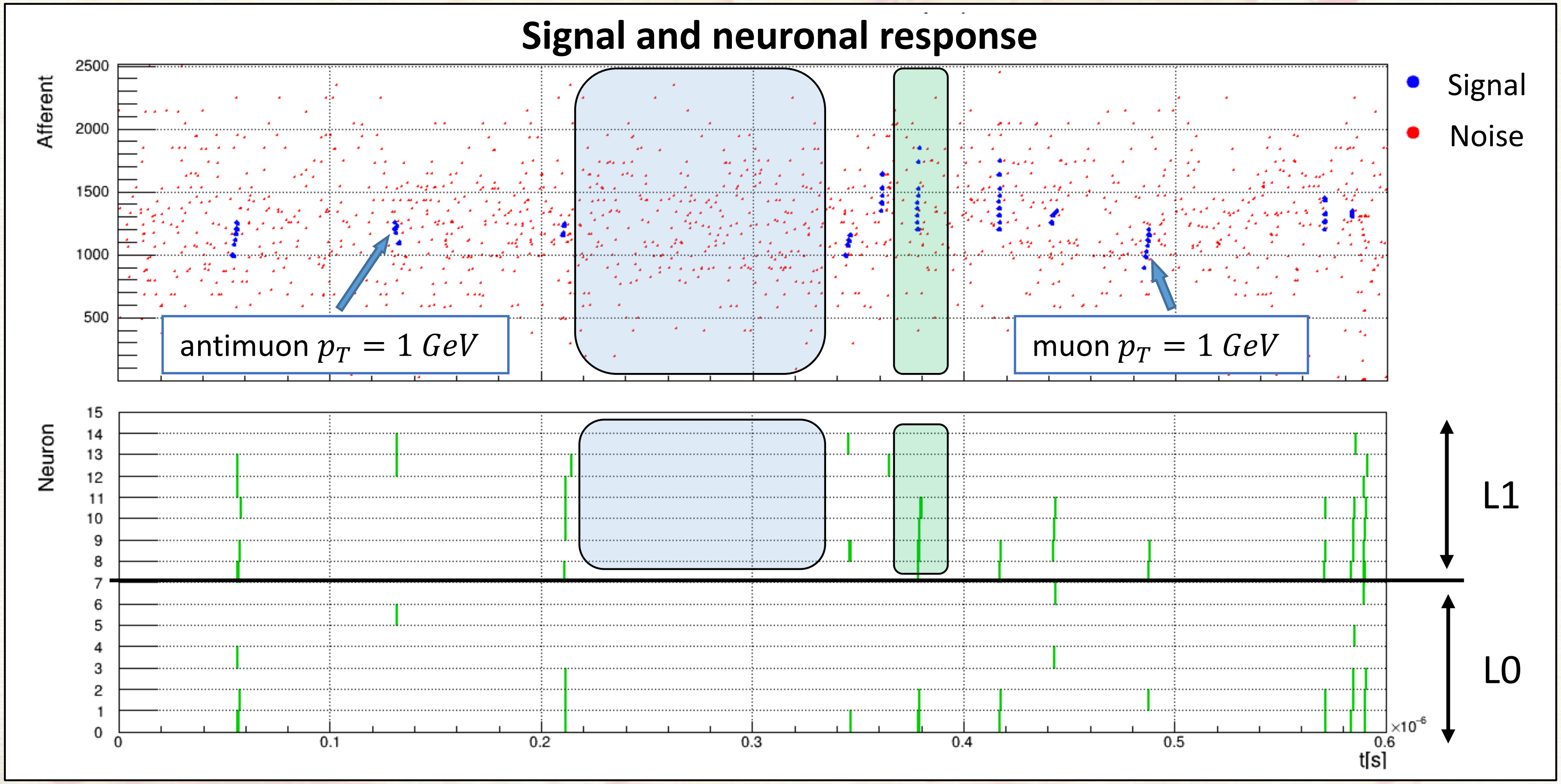
- Continuous signal encoding
- Better performance
- Presence of duplicates



3D model results

- Activation of neurons at **signal pulses**
- Inactive neurons at **noise pulses**

The network has **learnt autonomously** to recognize tracks from noise



3D model results

- Possible problems:
  - ➢ Activation during **noise events** → False positive
  - ➢ Falure to recognize **signal track**