NEUROMORPHIC READOUT FOR HOMOGENEOUS HADRON CALORIMETERS

Author: Enrico Lupi
Supervisor: Tommaso Dorigo

INTRODUCTION
GOAL OF THE PROJECT

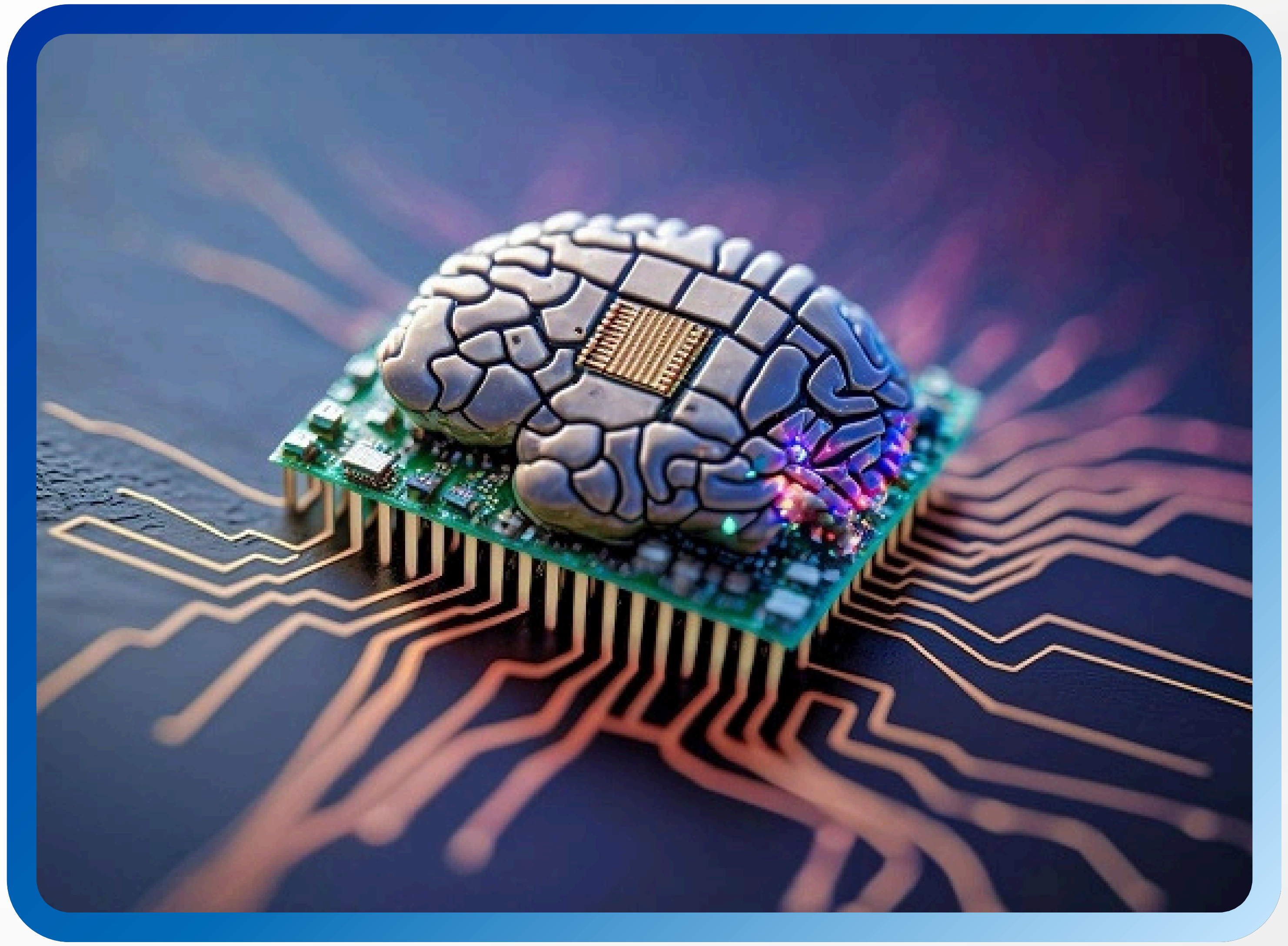
Investigating the readout of light signals from **hadronic showers** in a homogeneous calorimeter by a network of **nanowires** communicating through the time-encoding of **light pulses**.

We aim to offer:

- fast, energy-efficient local computation
- generation of informative high-level primitives

using **neuromorphic computing**.

This work is a **proof of principle**: we want to change paradigm and demonstrate that it is possible to apply this **new technology** efficienctly to calorimetry.



[Carlos Larrechi, via Alamy Stock]

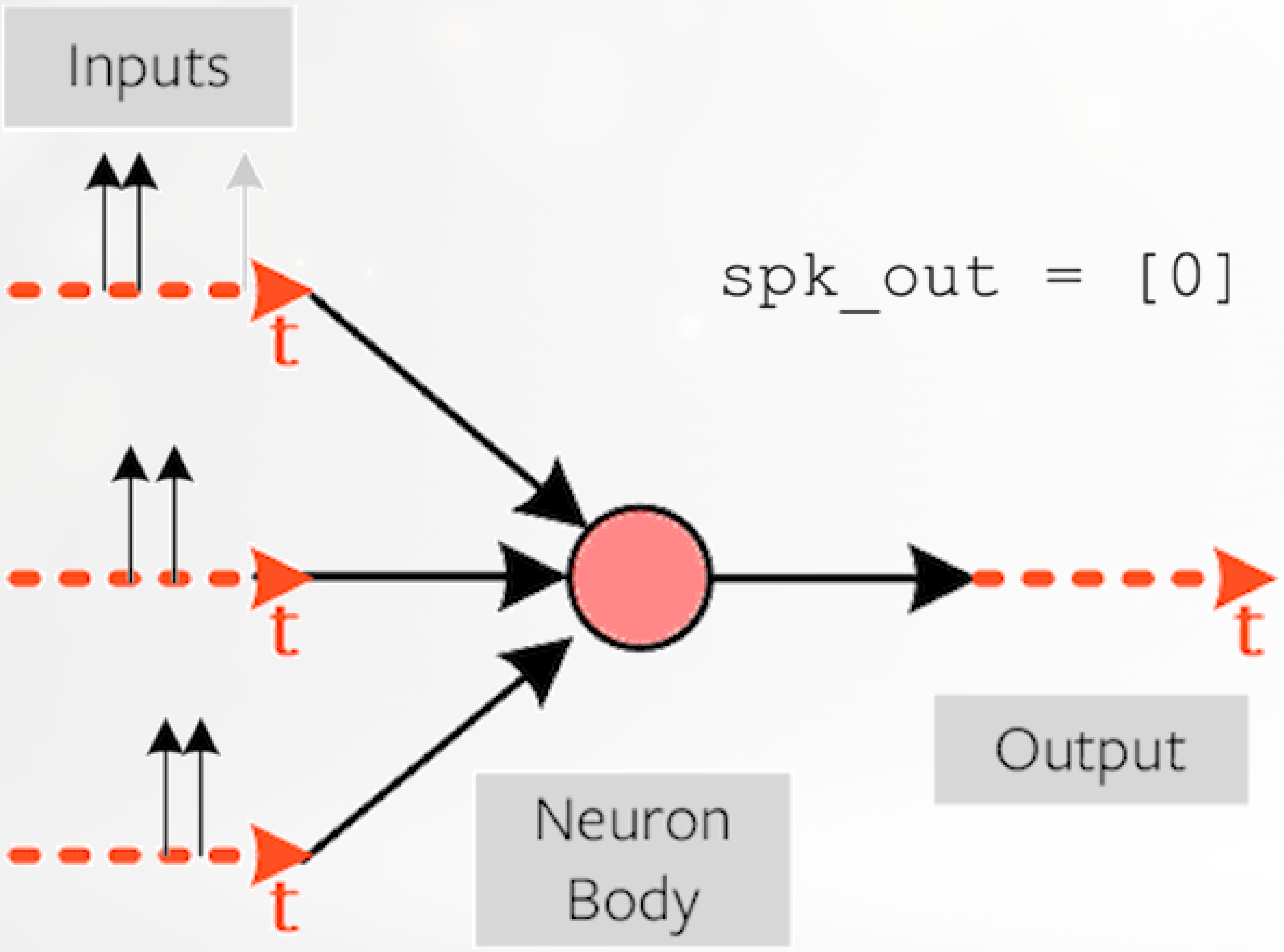
WHAT IS NEUROMORPHIC COMPUTING?
OVERVIEW

- Computing approach that **mimics** the structure and function of the **human brain** using artificial neurons and synapses.
- Studies both new software and hardware solutions
- More difficult to train, but provides multiple advatages...

1. **ENERGY EFFICIENCY:**
   Neuromorphic systems have extremely **low power consumption** and implement an **event-driven processing**, so that they activate only when there is a stimulus
2. **REAL-TIME PROCESSING:**
   Provide **extremely fast** processing times and **reduce latency**, enabling very fast responses
3. **ROBUSTENESS:**
   Inherently **resilient to noise** and faulty signals thanks to brain inspiration

WHAT IS NEUROMORPHIC COMPUTING?
SPIKING NEURAL NETWORKS

SNNs follow the same principle of usual ANNs, but with some key differences...



[J. K. Eshraghian et al. (2023). "Training Spiking Neural Networks Using Lessons From Deep Learning," Proceedings of the IEEE. doi: 10.1109/JPROC.2023.3308088.]

Spikes

Neurons communicate using discrete events called "spikes", which are **binary signals** that represent a neuron's **action potential**.

Temporal Encoding

Incoming data to be analyzed should also be encoded in a **spiketrain**. Information is contained in the **timing or rate** of the spikes

Membrane Potential

Each neuron has a membrane potential, which integrates incoming spikes over time.

When the accumulated potential **crosses a threshold**, the neuron **generates a spike and resets** its membrane potential.

For **LIF neurons**:

$$U[t+1] = \underbrace{\beta U[t]}_{\text{decay}} + \underbrace{WX[t+1]}_{\text{input}} - \underbrace{S[t]U_{\text{thr}}}_{\text{reset}}$$

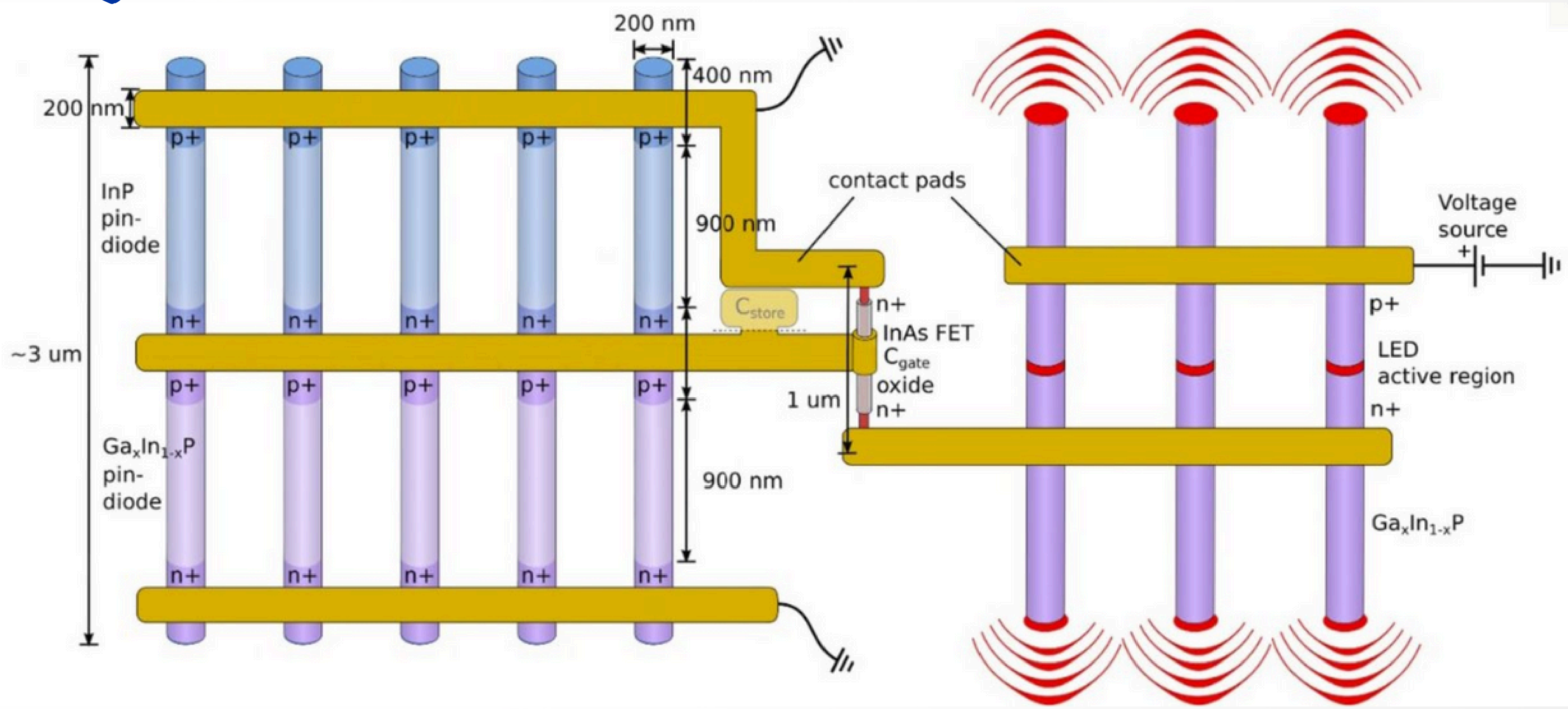
$$S[t] = \begin{cases} 1, & \text{if } U[t] > U_{\text{thr}} \\ 0, & \text{otherwise} \end{cases}$$

HOW WILL IT LOOK LIKE?
HARDWARE IMPLEMENTATION

Implements a **multilayer** approach:

1. Multi-nanowire photo detection layer
2. Waveguiding broadcast signals
3. Learning layer, using photo switchable molecules
4. Output layers, which moves data in external device

All computation happens locally and with **no transduction** (all "photonically")



[David Winge et al. (2023). "Artificial nanophotonic neuron with internal memory for biologically inspired and reservoir network computing." Neuromorph. Comput. Eng. 3 034011, doi:10.1088/2634-4386/acf684]

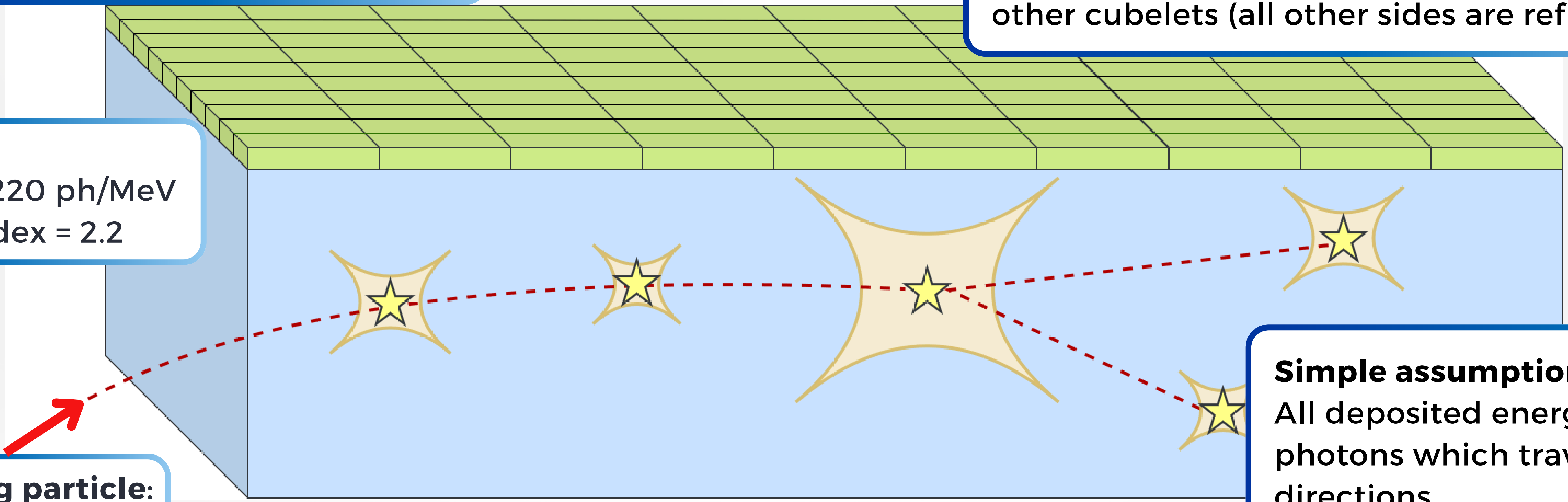
TEST CASE
HADRONIC CALORIMETER

Let's simulate a highly granular calorimeter

Segemetation: 10x10x10 cubelets
Size: 3x3x12 cm³

Segmented readout: 10 x 10 light sensors grid on the upper face of each cubelet.
Sensors are blind to the light coming from other cubelets (all other sides are reflective)

PWO:
- Light Yield ≈ 220 ph/MeV
- Refraction index = 2.2



Simple assumption:
All deposited energy is converted into photons which travel isotropically in all directions

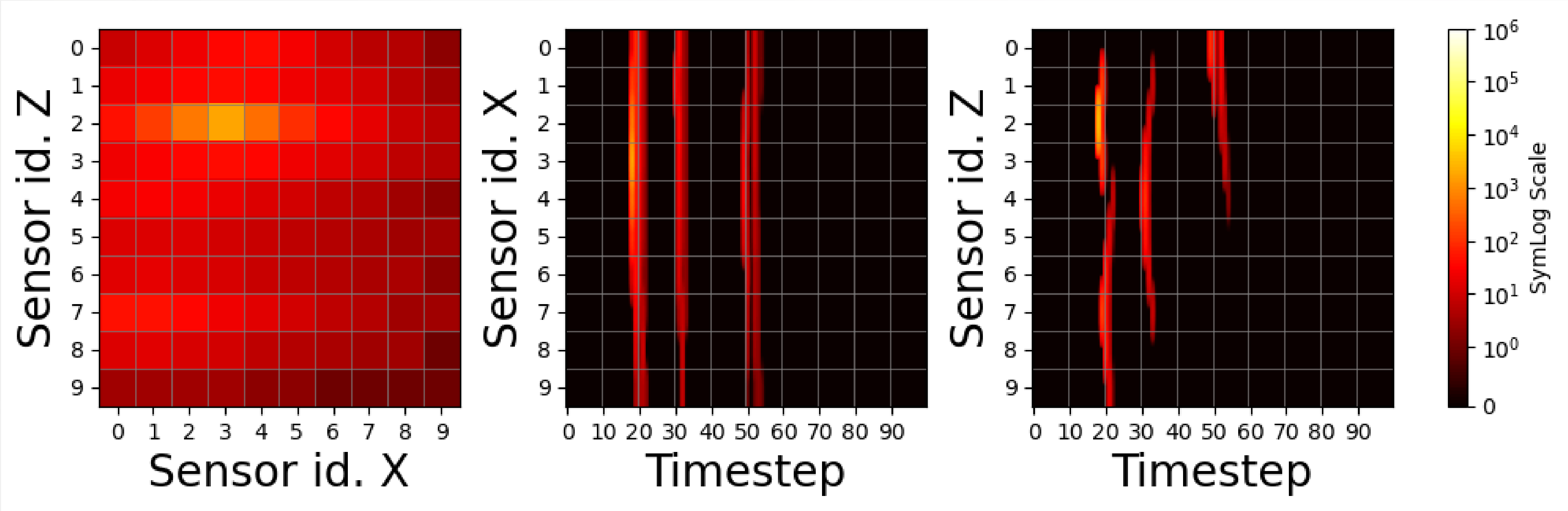
Incoming particle:
p, π or k
E = 100 GeV

TEST CASE
LIGHT SIGNAL PROCESSING

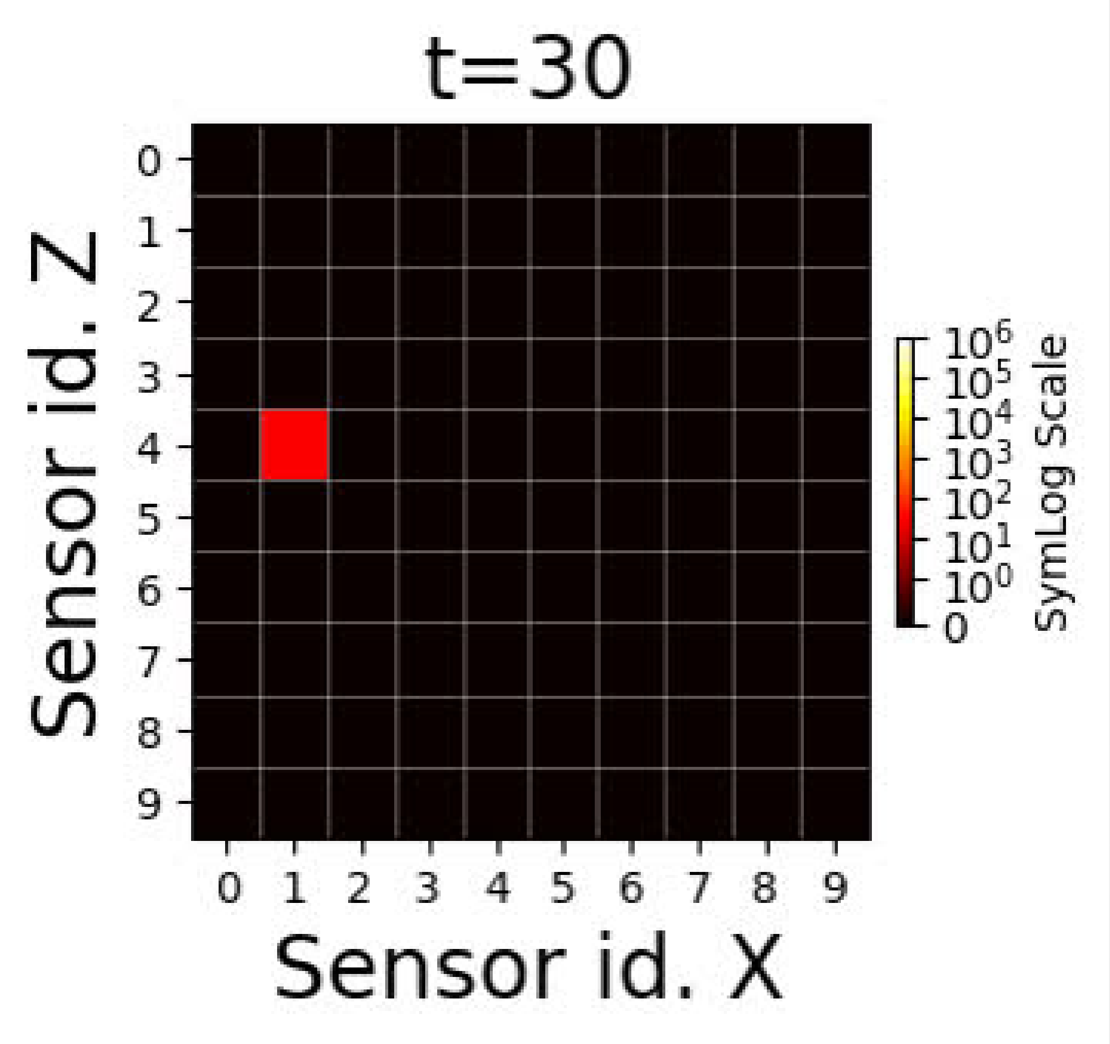
Photons are collected by the light sensors for T_{max} = 20 ns
The signal is discretized into N_t = 100 bins of Δt = 0.2 ns

Here is an example of a recorded event:

Integrated Views

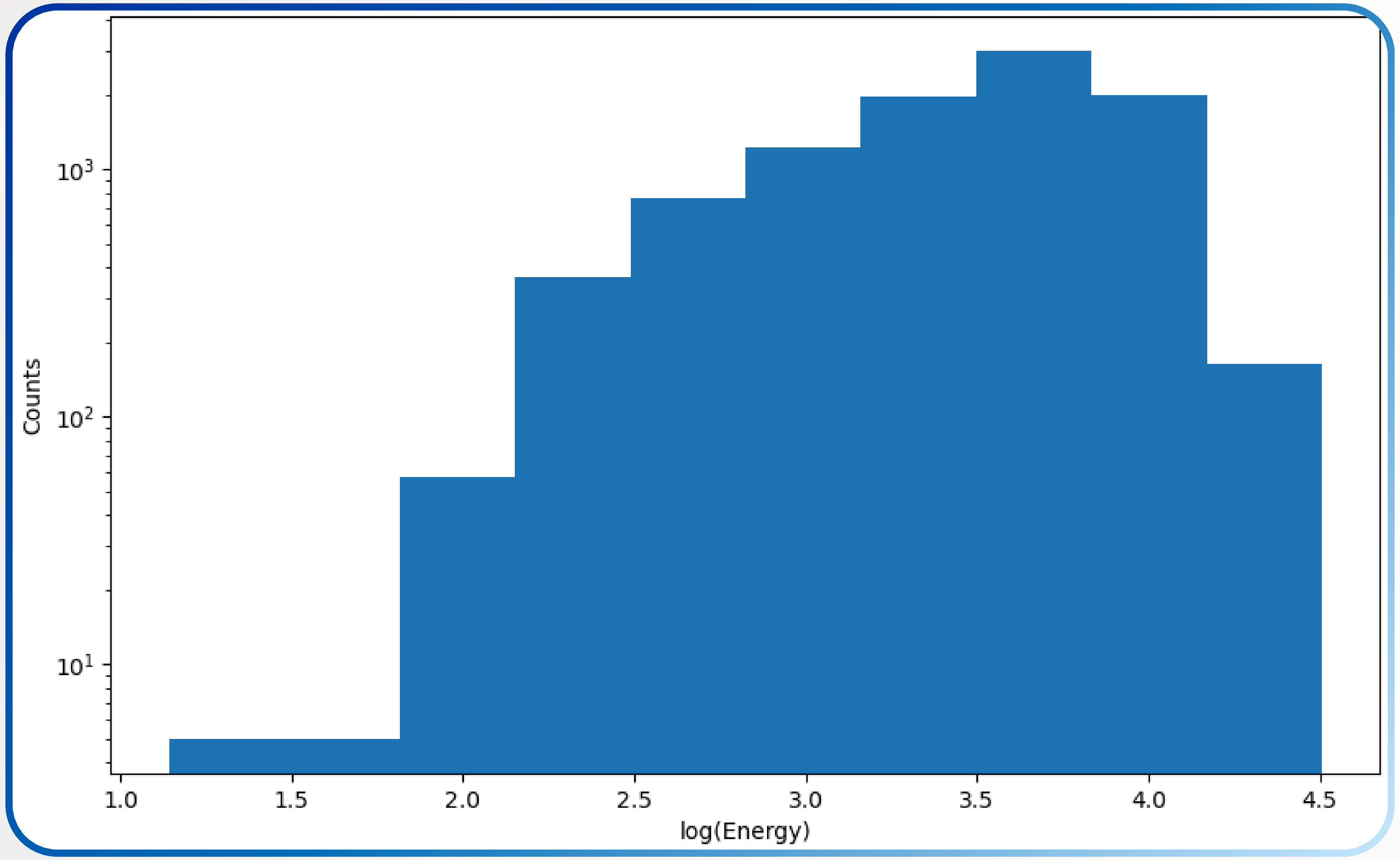


Time Evolution



ENERGY REGRESSION TASK
THE OBJECTIVE

The first task is regressing the **energy released inside one cubelet**.



The values span various orders of magnitude, from a couple of MeV to tens of GeV.

⇒ It is easier to work in **logarithmic scale**

ENERGY REGRESSION TASK
SPIKE ENCODING

As mentioned previosuly, incoming data should be converted into a spiketrain.

Easiest solution:

- **One afferent** (i.e. information-carrying lines). per channel
- Create spike if enough photons reach sensor in that timestep

$$S[t] = \begin{cases} 1, & \text{if } N_{ph}[t] \geq N_{thr} \sim 300 \\ 0, & \text{otherwise} \end{cases}$$

⇒ No way to distinguish intensity!

ENERGY REGRESSION TASK
SPIKE ENCODING

As mentioned previosuly, incoming data should be converted into a spiketrain.

Easiest solution:

- **One afferent** (i.e. information-carrying lines). per channel
- Create spike if enough photons reach sensor in that timestep

$$S[t] = \begin{cases} 1, & \text{if } N_{ph}[t] \geq N_{thr} \sim 300 \\ 0, & \text{otherwise} \end{cases}$$

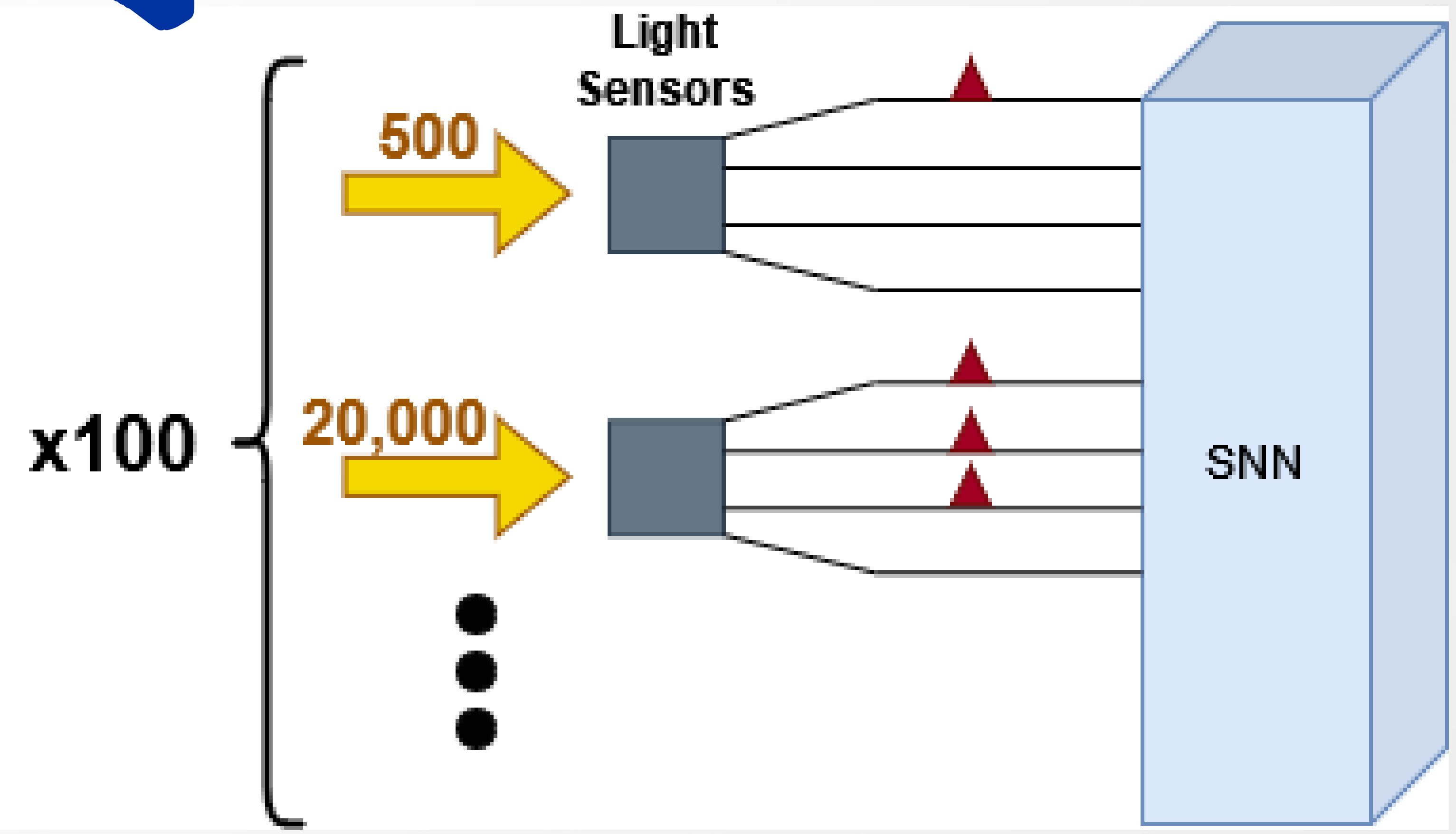
⇒ No way to distinguish intensity!

Four afferents per readout light sensor:

- Create spike if enough photons reach sensor in that timestep with different thresholds per afferent

$$S^{(i)}[t] = \begin{cases} 1, & \text{if } N_{ph}[t] \geq 10^{i+2} \\ 0, & \text{otherwise} \end{cases}, \quad i = 0, 1, 2, 3$$

⇒ Spikes carry information about **time and intensity**



ENERGY REGRESSION TASK
OUTPUT DECODING

The network does not give us a numerical value like ANNs... How do we obtain the energy value?
We can exploit two quantities:

Membrane Potential

Use the value of the output neurons's membrane potential at the last timestep

$$\hat{E} = U[N_t]$$

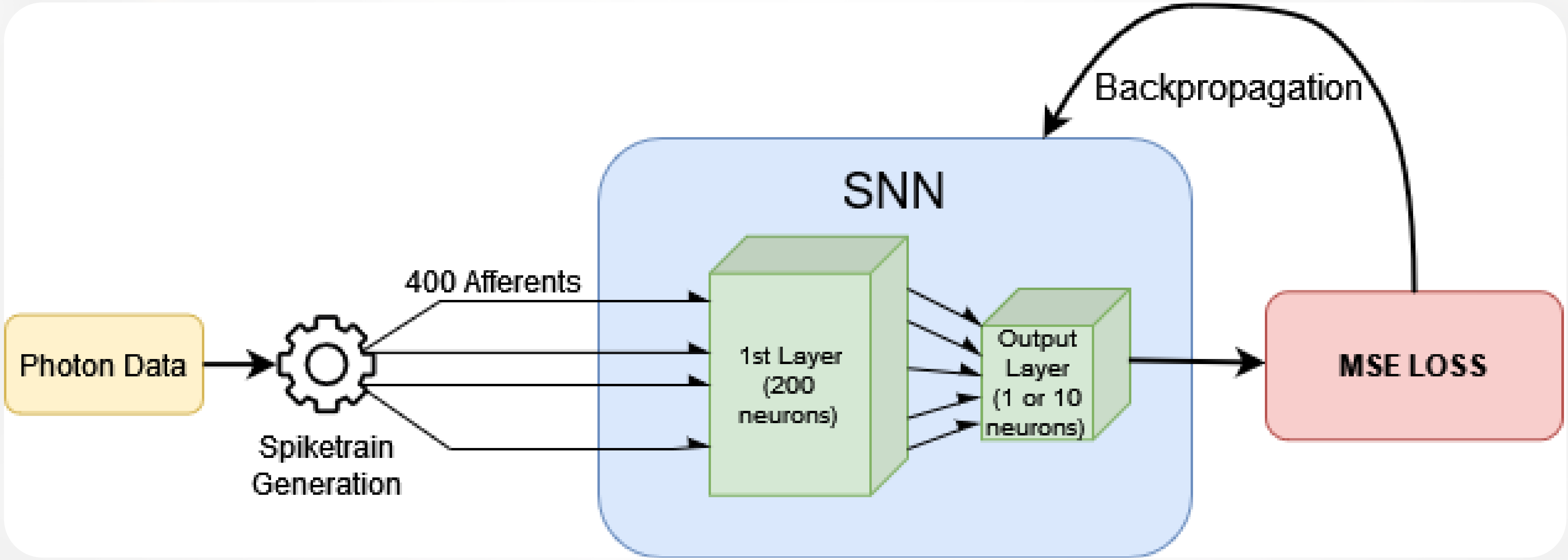
Output Spikes

Use the mean firing rate of a population of output neurons across the inference time

$$\hat{E} = \frac{1}{N_t \cdot N_{pop}} \sum_{t=1}^{N_t} \sum_{i=1}^{N_{pop}} S^{(i)}[t]$$

ENERGY REGRESSION TASK
NETWORK DESCRIPTION

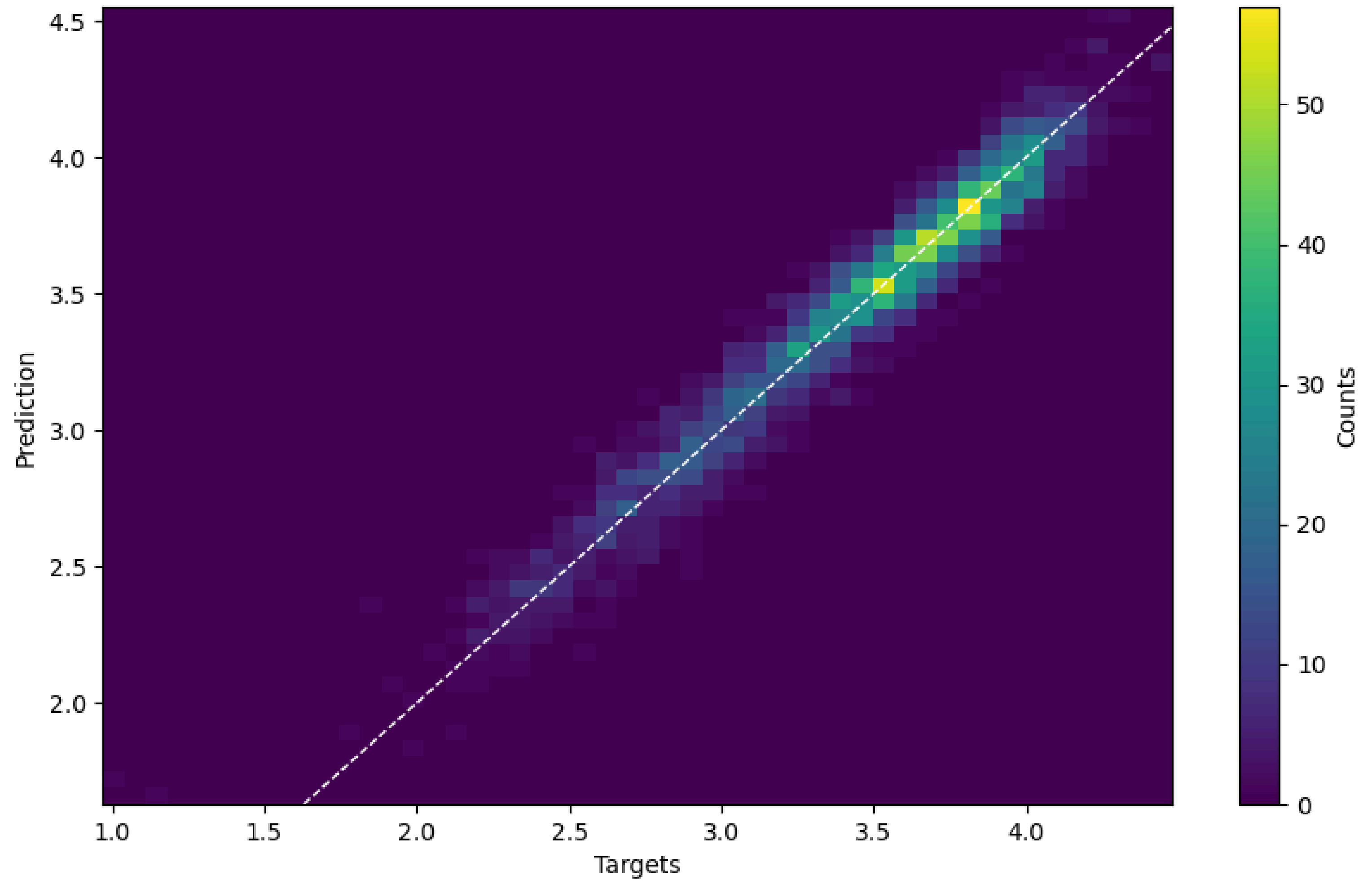
Here is an overview of the overall pipeline.
Each layer uses LIF neurons with learnable U_{thr} and β parameters.



ENERGY REGRESSION TASK
RESULTS

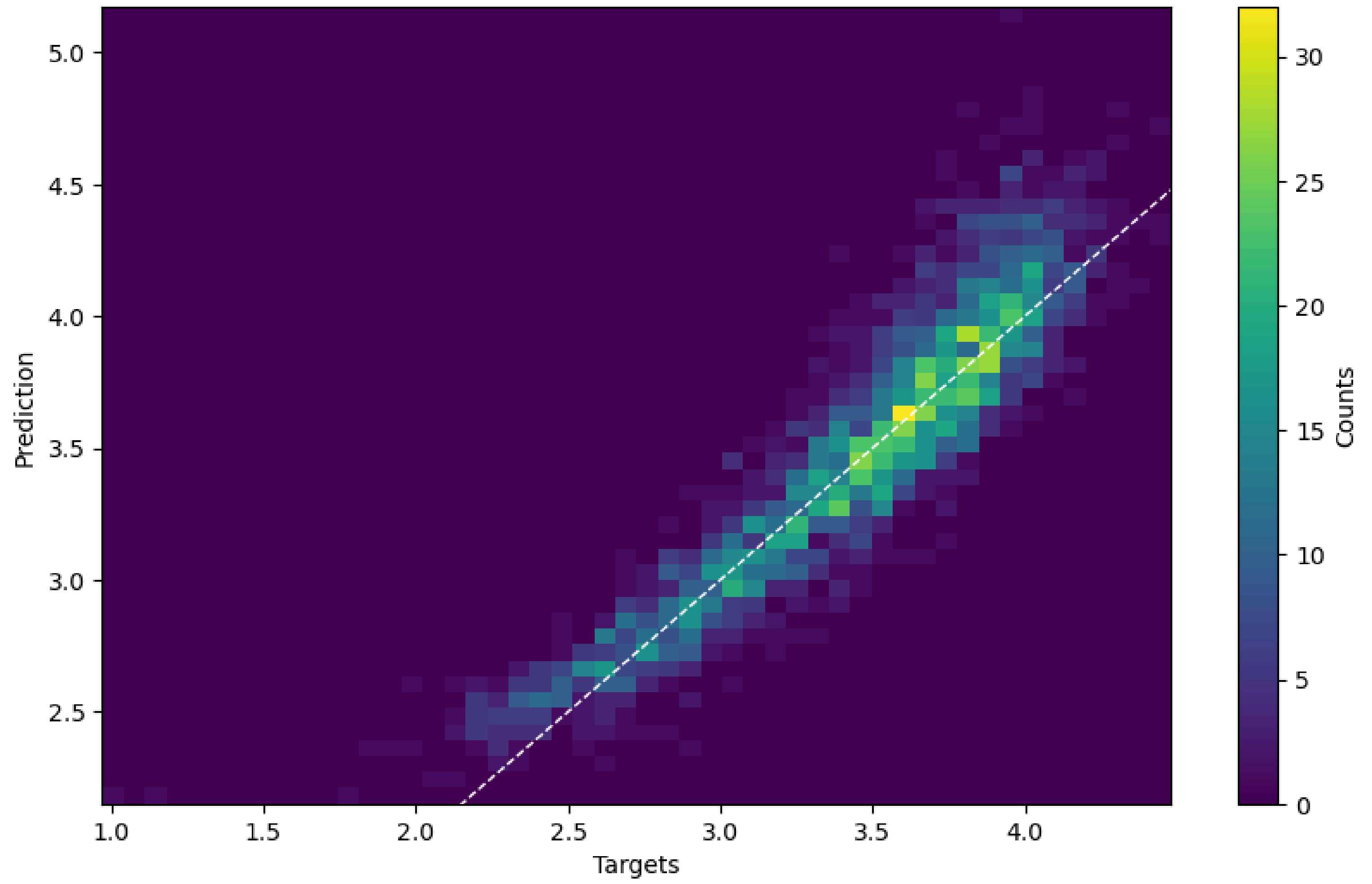
Membrane Potential

Loss: 0.0122 Relative Error: 2.632%



Spike Frequency

Loss: 0.0408 Relative Error: 4.511%



WHAT NOW?
OUTLOOK

This project is the **first ever attempt** to use neuromorphic solutions for calorimetry readout!

It is still in its infancy, there are many more steps ahead:

- Next task: **regression of the energy dispersion** inside one cubelet.
  In general, we aim to discover how much information it is possible to recover using NC.

- Implementing **hyperparametrs optimization**: testing new network configurations, neuron models, encoding schemes...   The possibilities are endless!

THANK YOU!

Contacts:
Enrico Lupi: enrico.lupi@studenti.unipd.it

BACKUP SLIDES

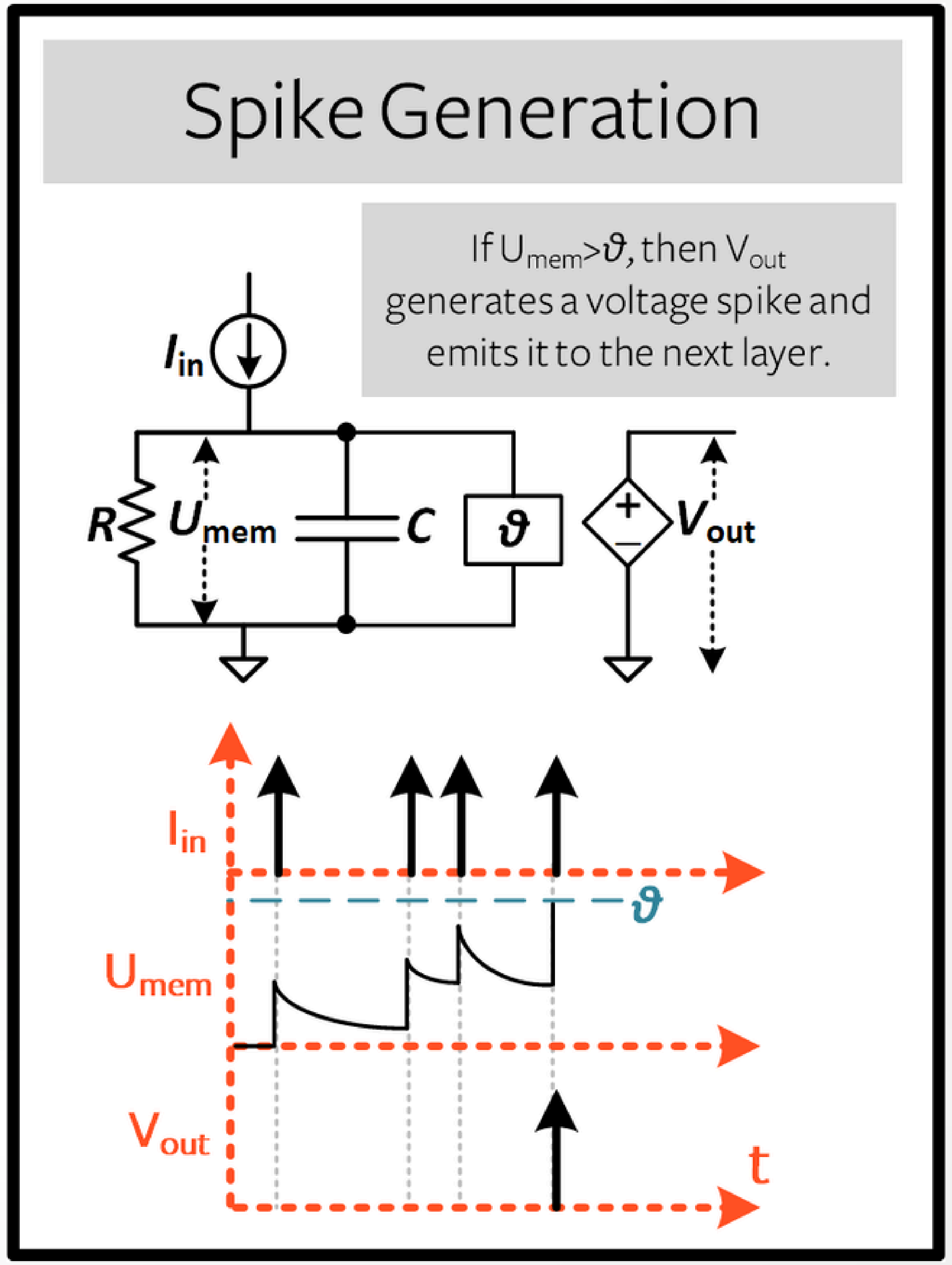
WHAT IS NEUROMORPHIC COMPUTING?
THE LIF NEURON MODEL

There are multiple neuron models that mostly regulate how the membrane potential evolves over time.

In this work we will adopt the **Leaky-Integrate-and-Fire** (LIF) Neuron:

- Simplest neuron model
- Based on the **RC circuit**, simplified and discretized over time to obtain: $$U[t+1] = \underbrace{\beta U[t]}_{\text{decay}} + \underbrace{WX[t+1]}_{\text{input}} - \underbrace{S[t]U_{\text{thr}}}_{\text{reset}}$$

$$S[t] = \begin{cases} 1, & \text{if } U[t] > U_{\text{thr}} \\ 0, & \text{otherwise} \end{cases}$$



[J. K. Eshraghian et al. (2023). "Training Spiking Neural Networks Using Lessons From Deep Learning," Proceedings of the IEEE. doi: 10.1109/JPROC.2023.3308088.]