ν_e AND ν_μ PARTICLE ID WITH MACHINE LEARNING

Stefano Calí, Jonas Eschle, Martina Ferrillo, Jack Jenkins, Kevin Keefe, Edward Millard, Brant Rumberger, Yicong Sui

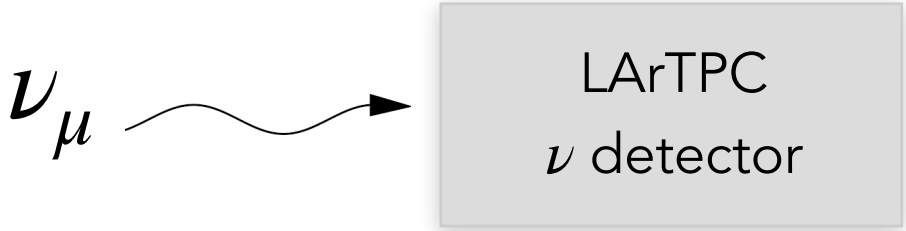
SLAC Summer Institute 2019 | 12th - 23rd August 2019

Motivation

Particle Identification (**PID**) is a crucial challenge in experimental particle physics, ranging from high energy to astro-particle experiments

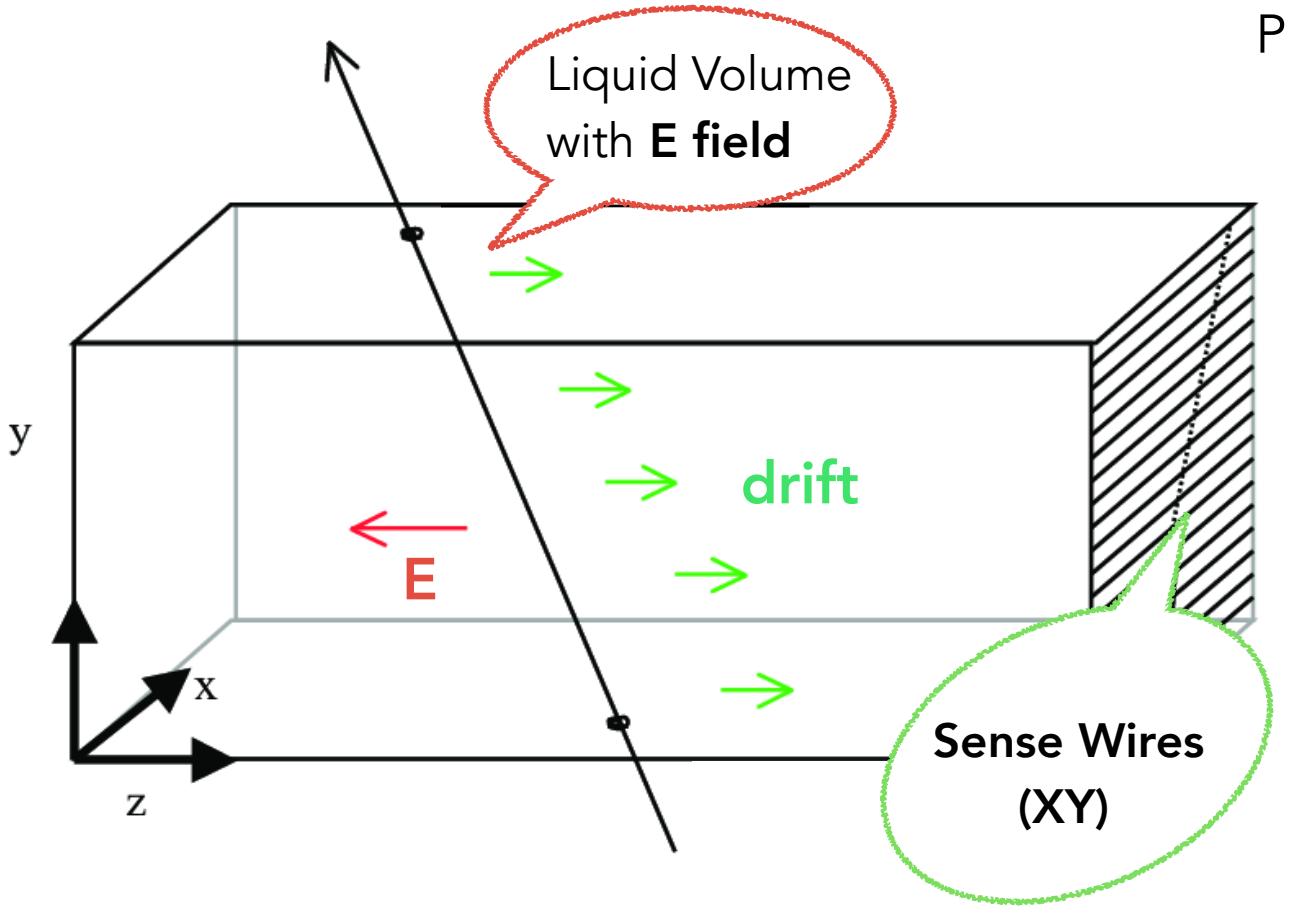
Aim of the project

In a neutrino oscillation experiment ($\nu_\mu \rightarrow \nu_e$) determine the **number** of **detected** ν_e interactions on top of 1M of ν_μs to get a 5 σ **significance Z**.



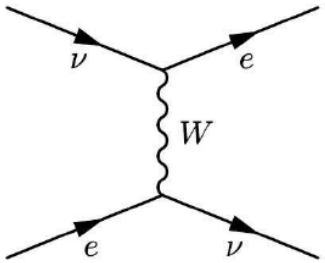
$$Z = \frac{n_s}{\sqrt{n_s + n_b}}$$

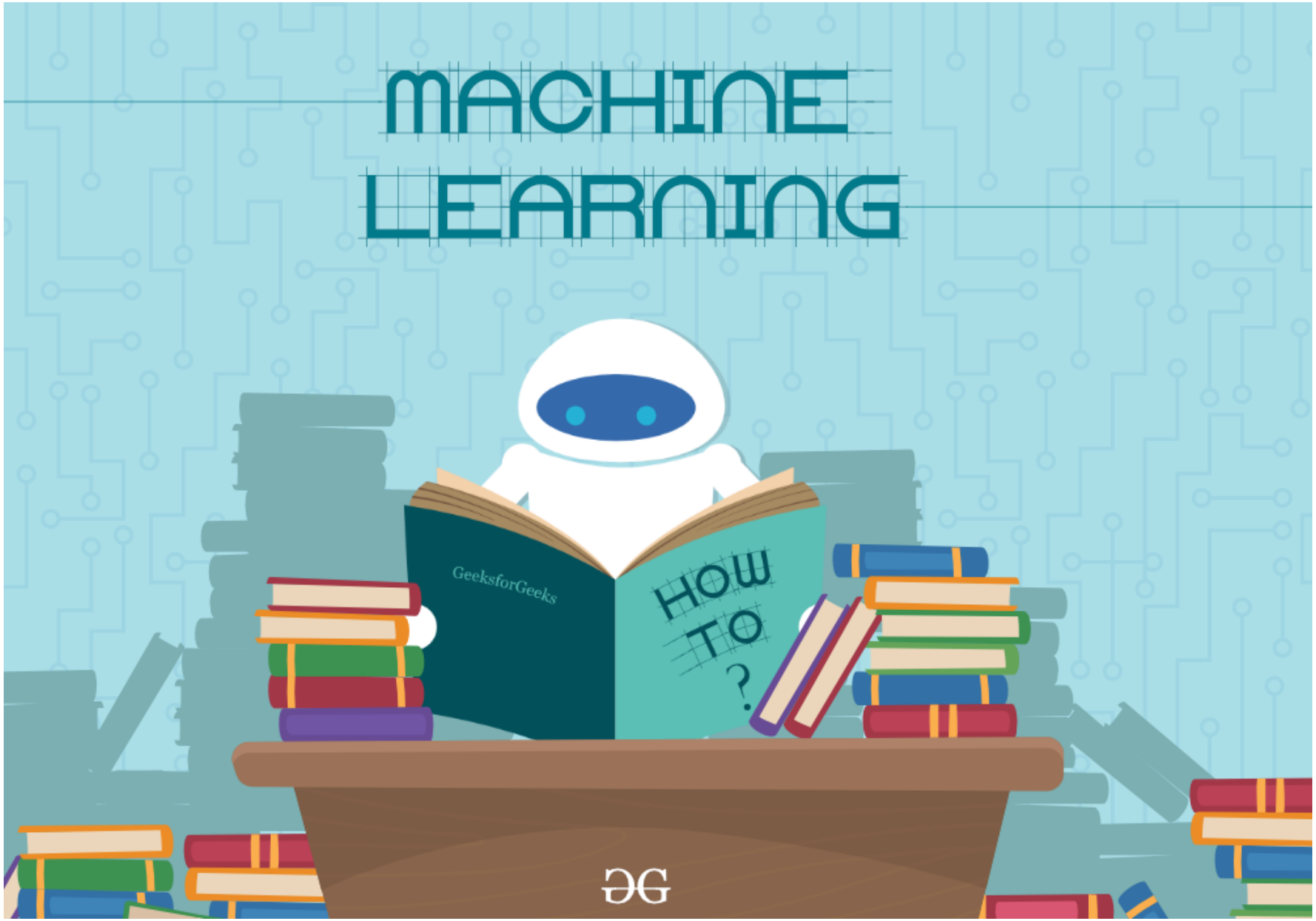
LArTPC - Liquid Argon Time Projection Chamber



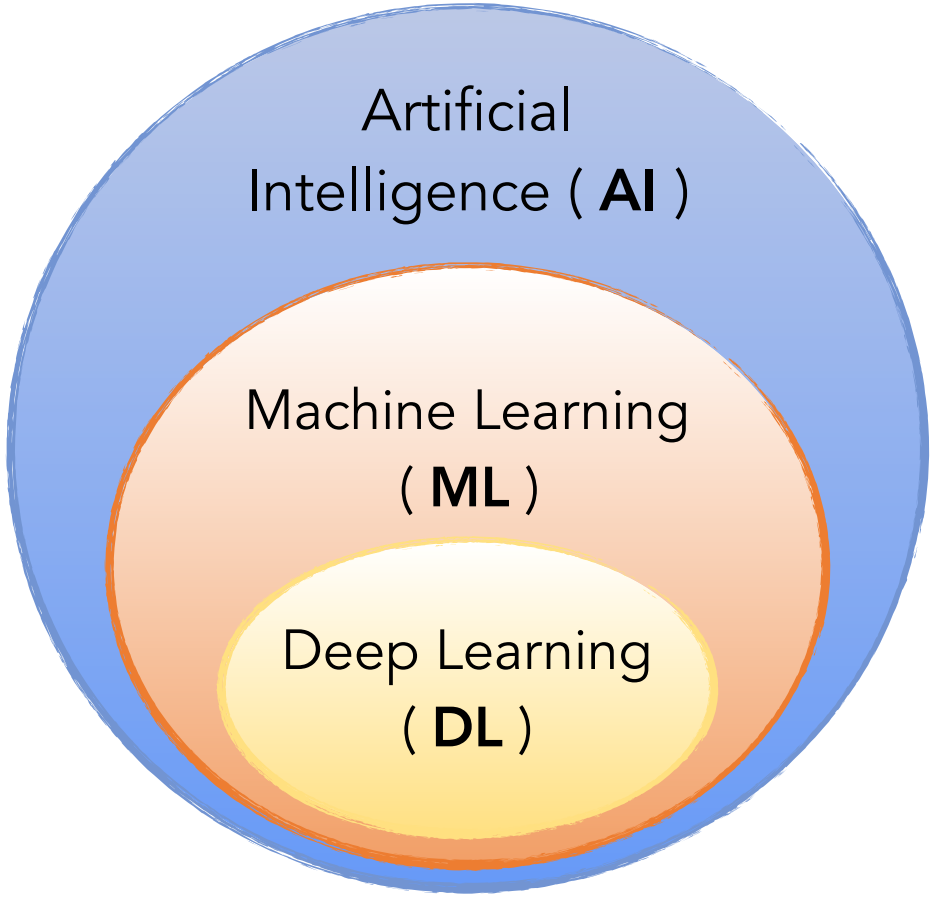
Provides 3D tracking and dE/dx measurement:

1. **Charged particle** enters fiducial volume and **ionises** medium
2. Primary ionisation **drifts** in uniform electric field to **readout plane** (**XY**)
3. Drift component (**Z**) reconstructed using drift velocity and trigger time



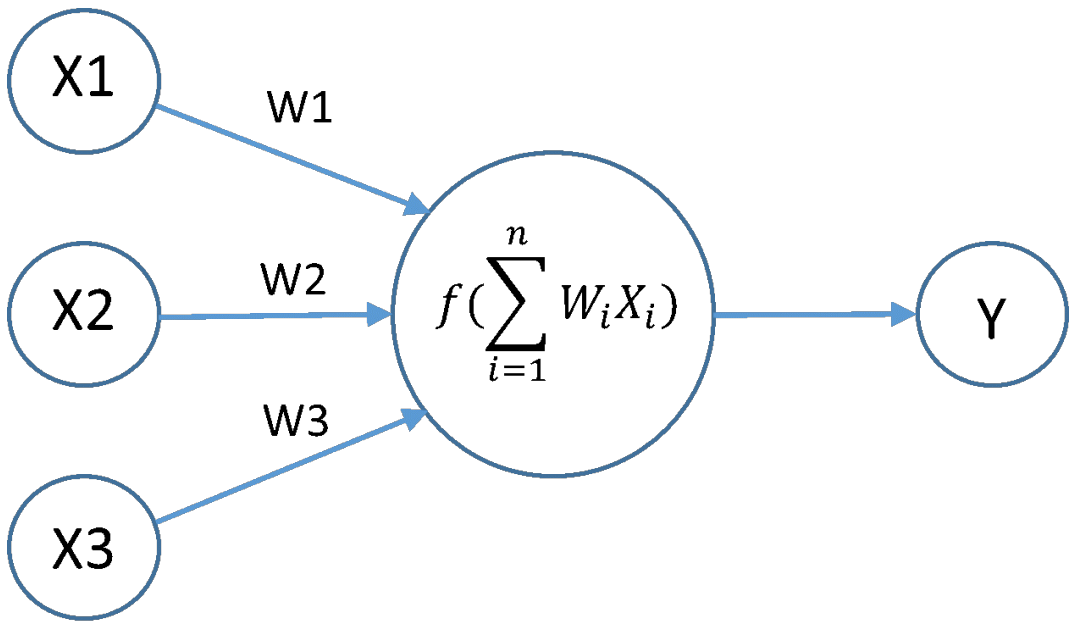

MACHINE
LEARNING
GeeksforGeeks
HOW
TO
?

Machine Learning in a nutshell (I)



> ***"[…] What we want is a machine that can learn from experience"***
>
> Alan Turing, 1947

Machine Learning in a nutshell (II)



- Input data X_i - *features*
- **Train** a model - f
- Make ***predictions*** Y on new data

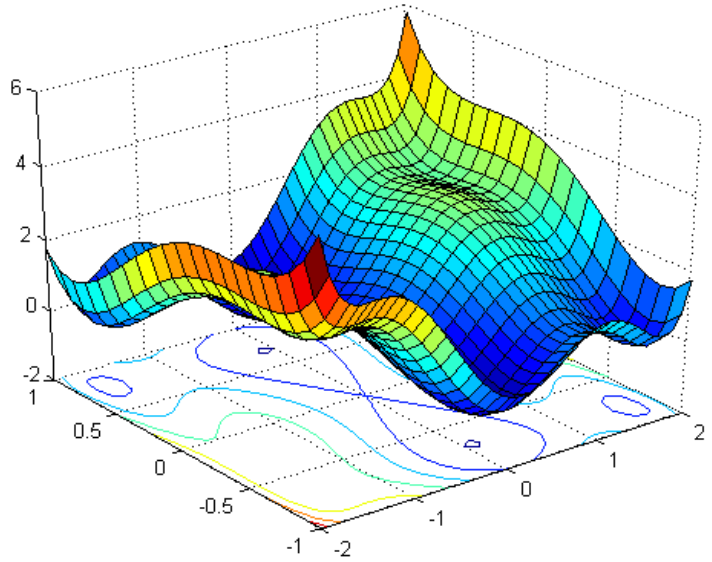
THE TRAINING PROCESS STEP BY STEP

iterate

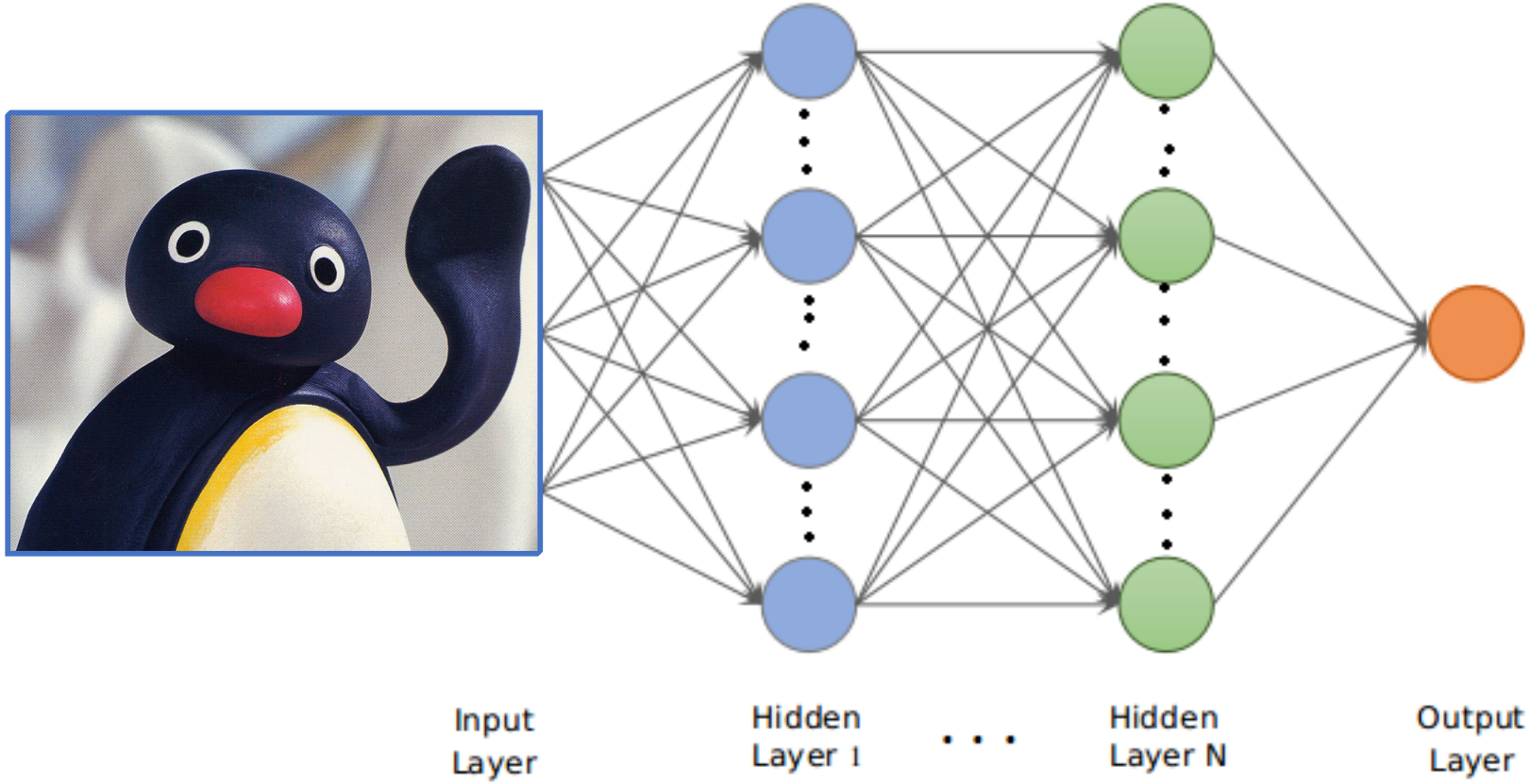
- Provide the input X_i to the neural network
- Calculate how far is the prediction $\hat{Y}$ from the truth Y - ***Loss function*** $\mathscr{L}$
- Determine the gradient of $\mathscr{L}$ with respect to the parameters w_i
- Update the parameters w_{i+1}

$$w_{i+1} = w_i - \alpha \cdot \nabla_w \mathscr{L}$$

Optimisation algorithm, *e.g.* Stochastic Gradient Descent evaluated on batches of the input data



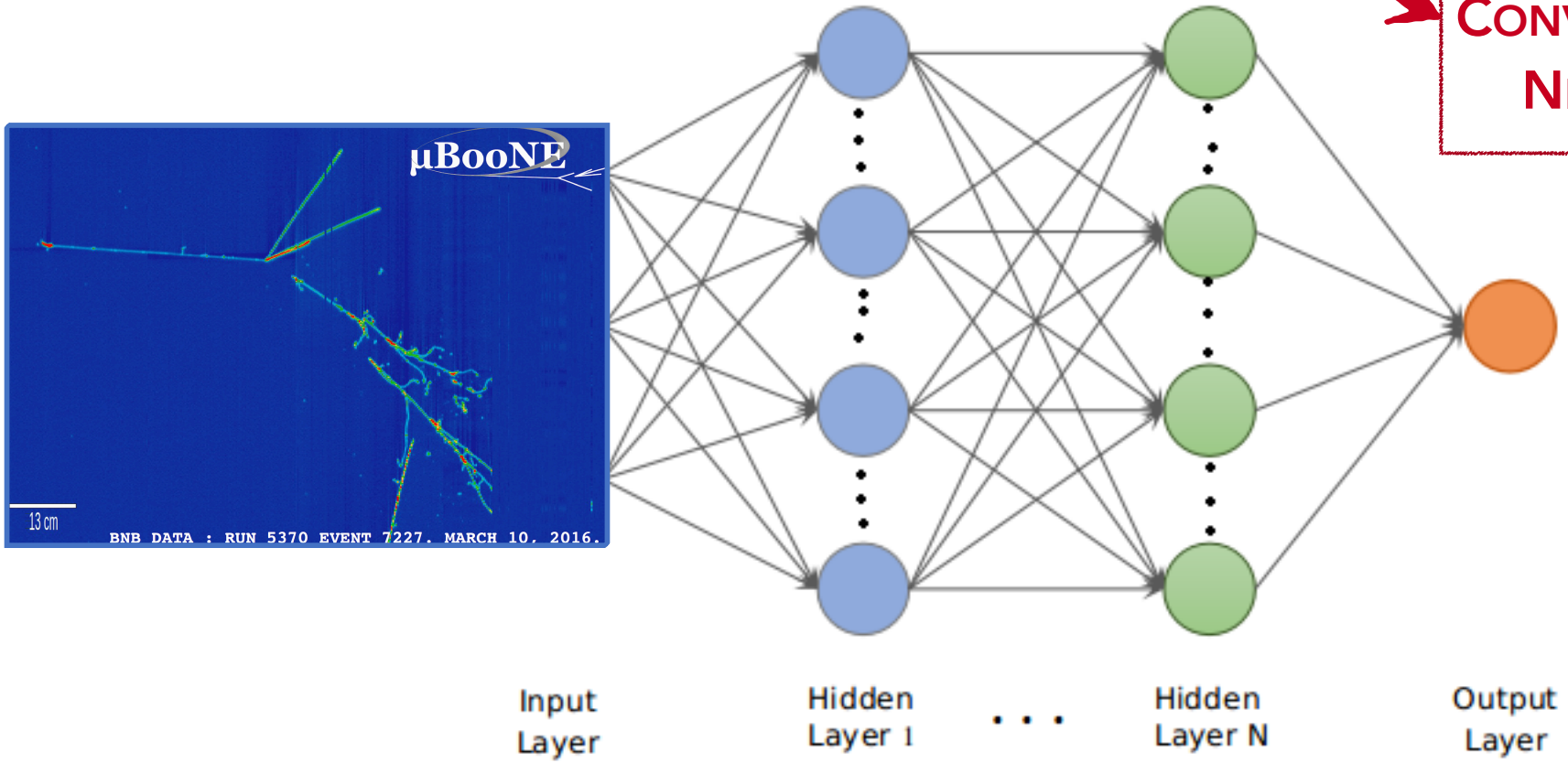
The Problem: Image Classification with ML approach



P(Pingu)

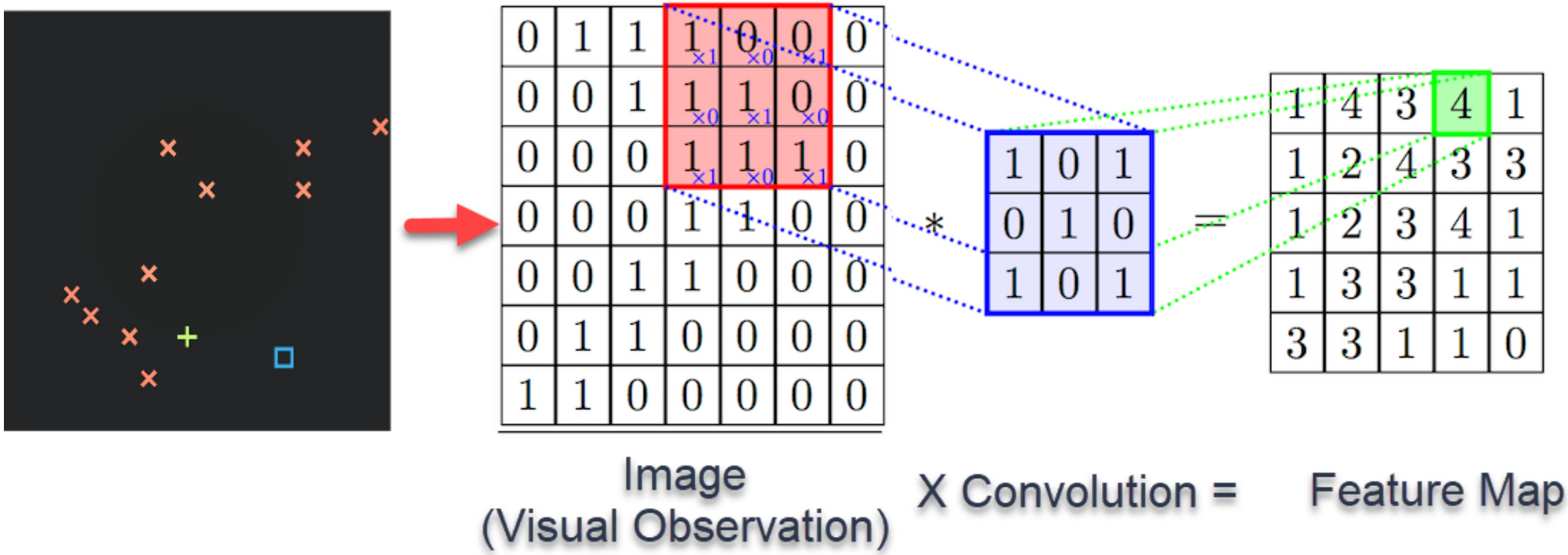
The Problem: Image Classification with ML approach

Convolutional Neural Networks (CNNs)



$P(\nu_e)$

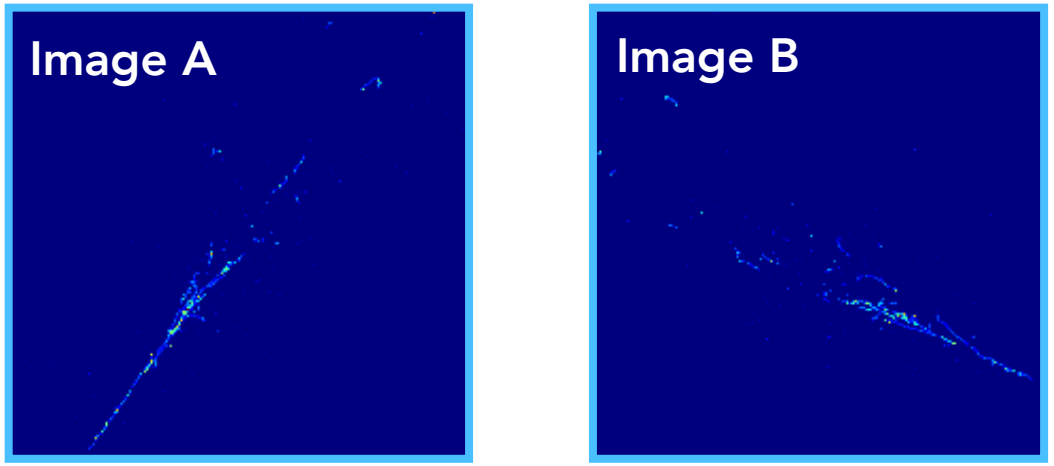
Convolutional Layers



Deep Neural Network Optimisation and Results

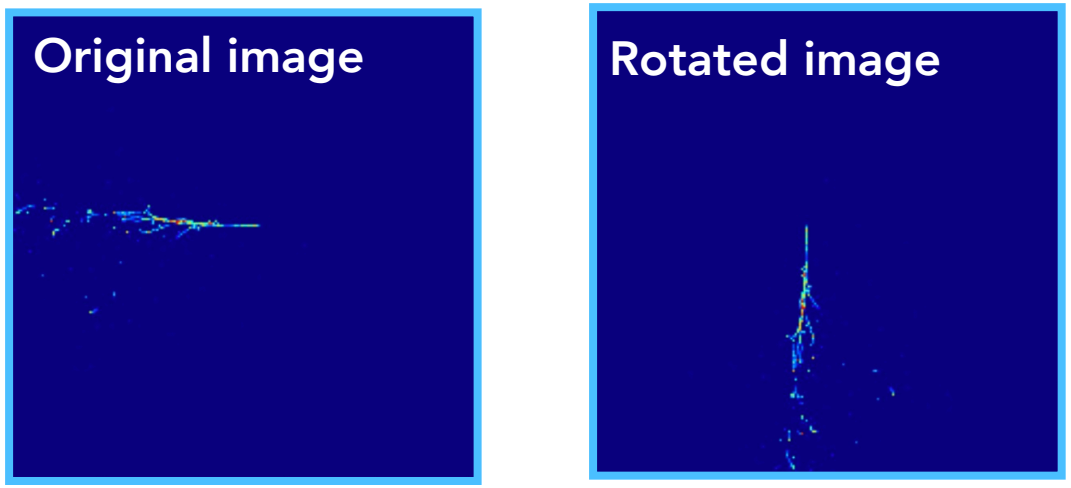
DNN Architecture Modifications

- Expand the training sample set by including a second 2D projection of the same track — more **features**



- **Hyper-parameters** tuning, *e.g.* number of layers, batch size etc.

- **Data augmentation**, *i.e.* artificially increase the training sample size

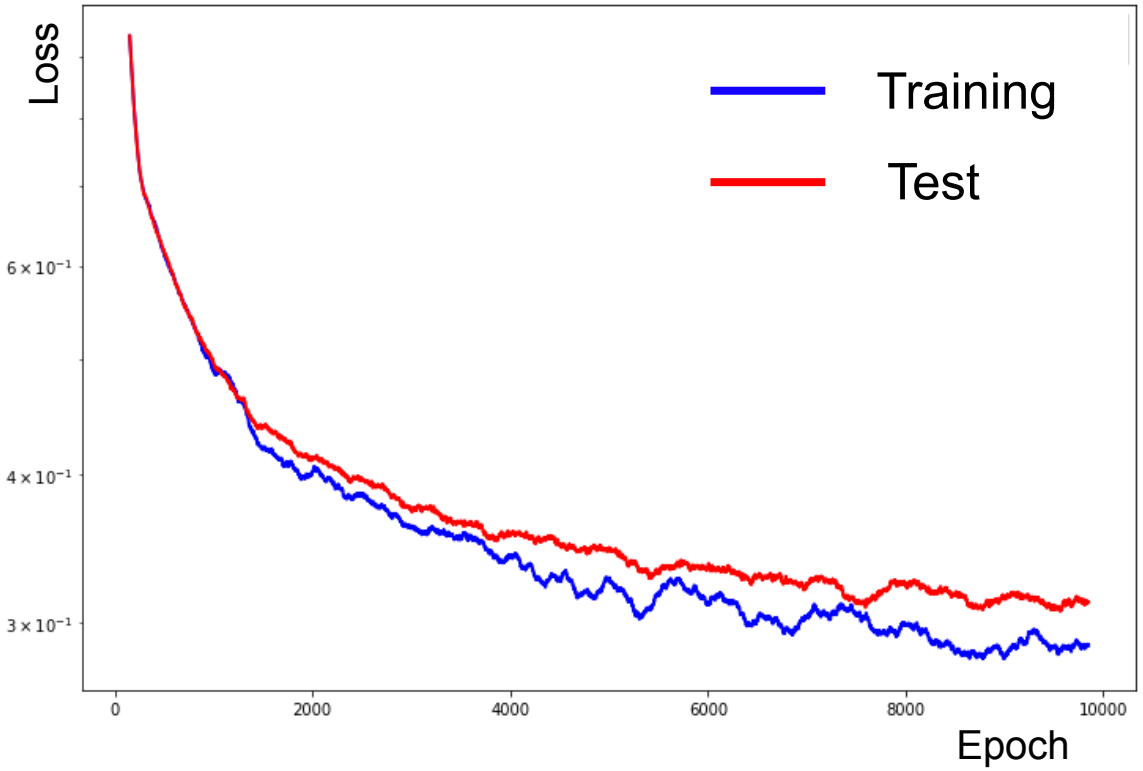


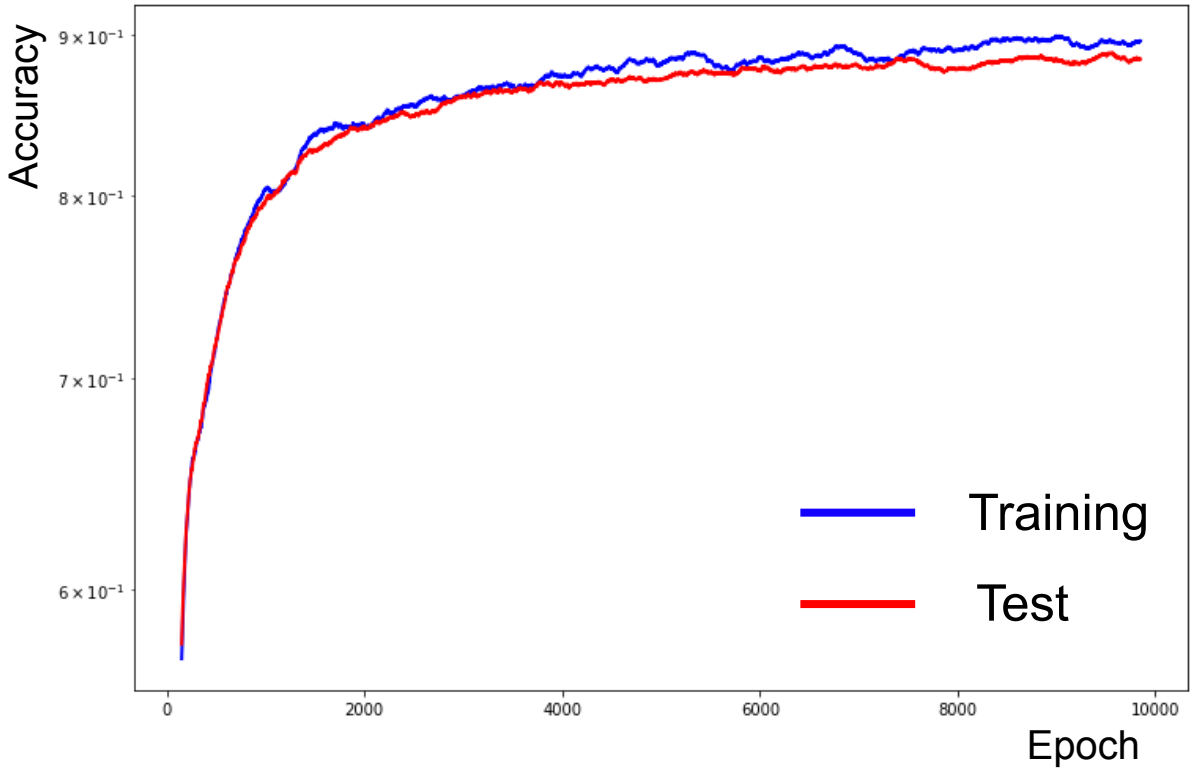
- **Spatial Transformer Network**

Loss and Accuracy

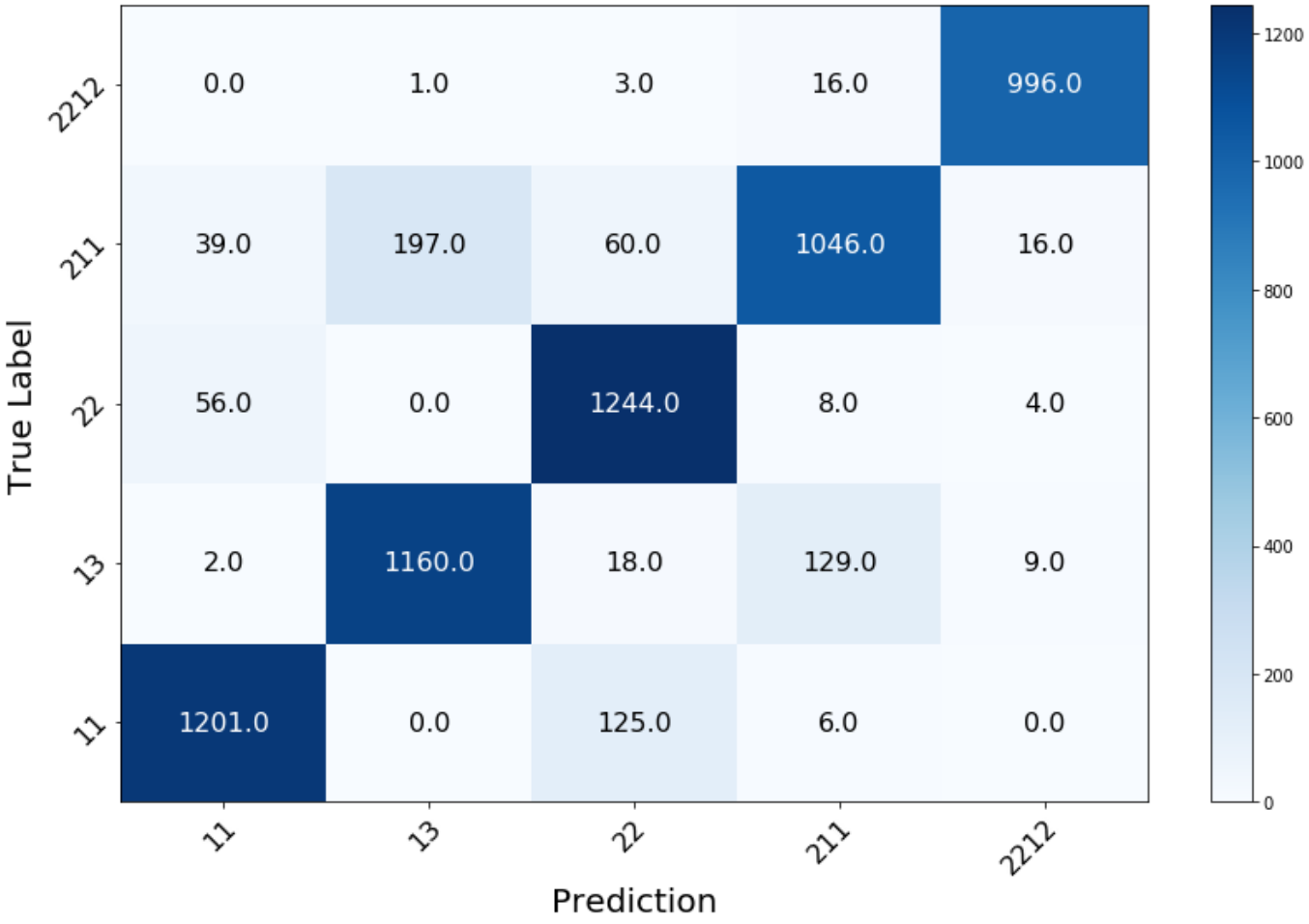
Cross entropy Loss: $$-\sum_{i=1}^{N} Y_i \log(\hat{Y}_i)$$

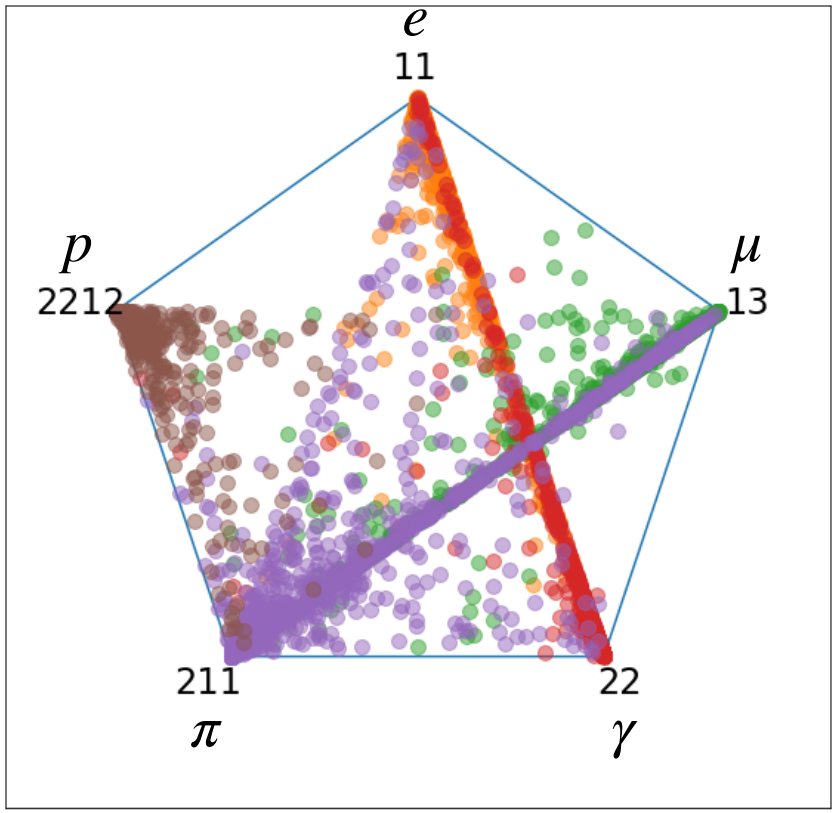
Accuracy: $$\frac{\text{Nr . Correct predictions}}{\text{Tot . nr . predictions}}$$



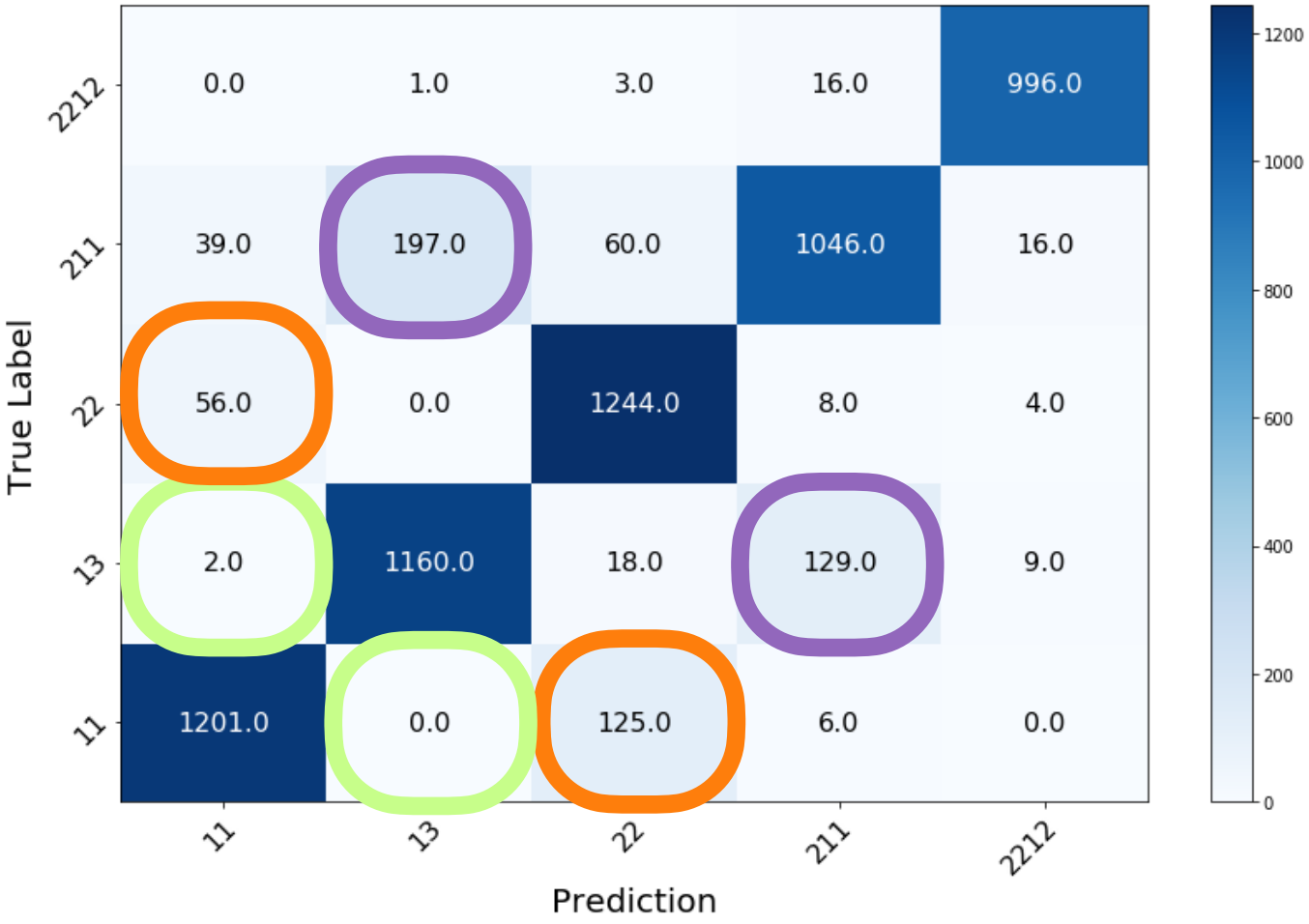


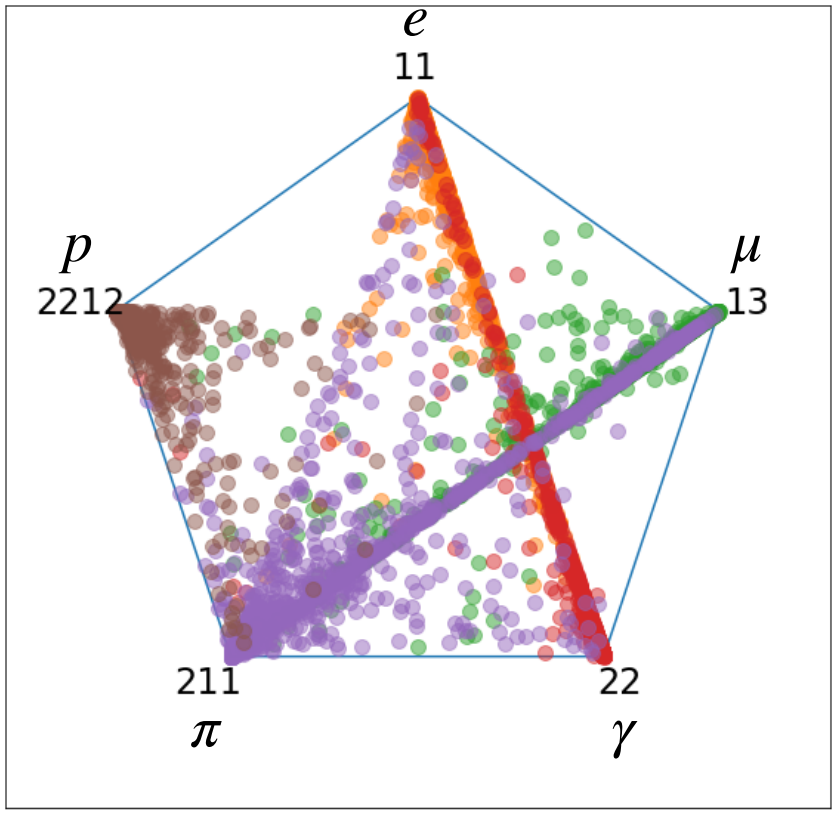
Performances and Results



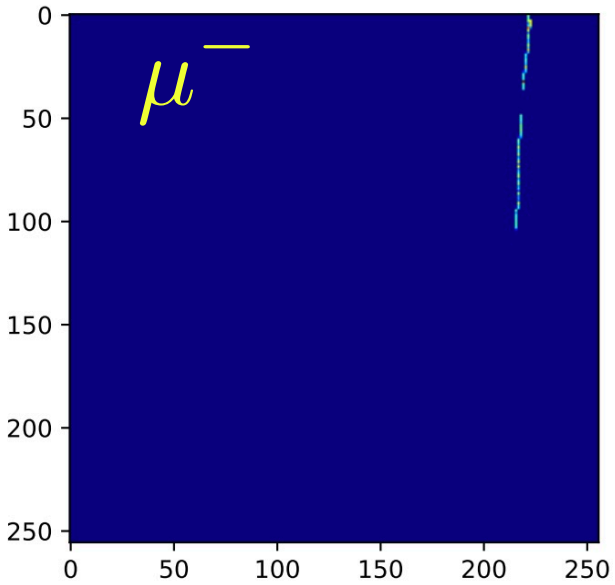


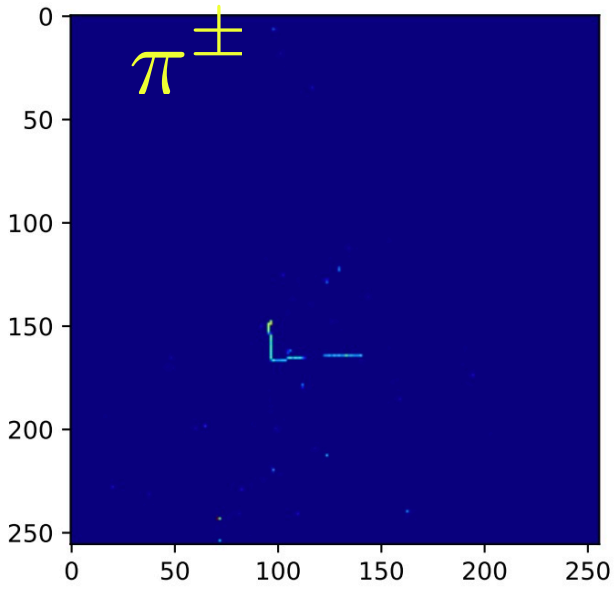
Performances and Results

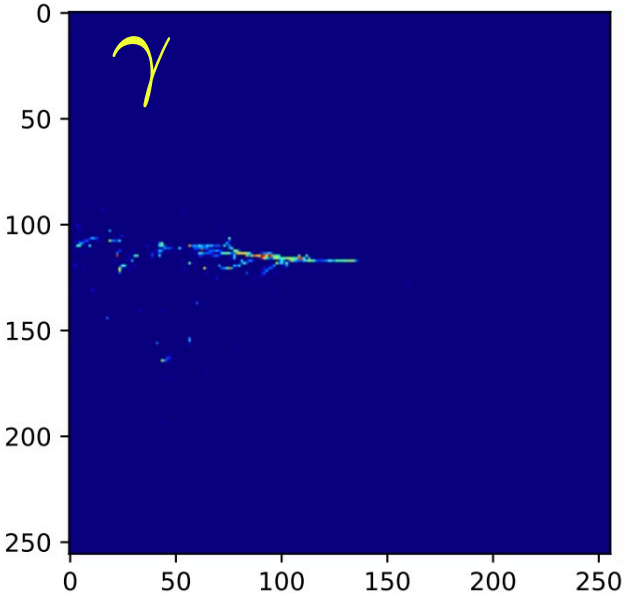


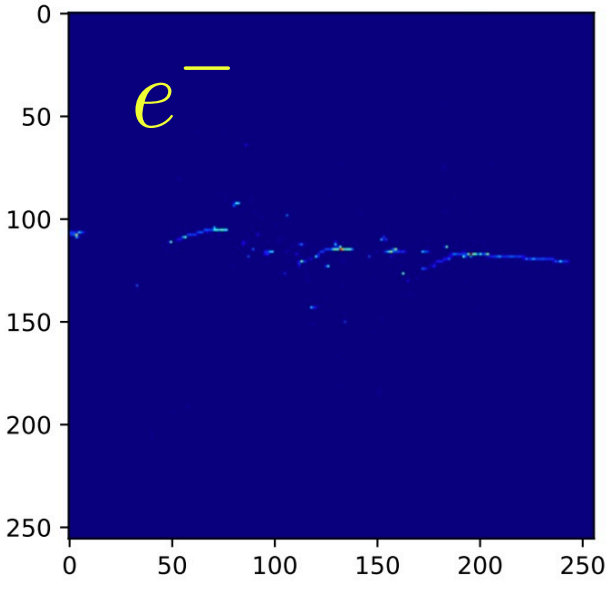


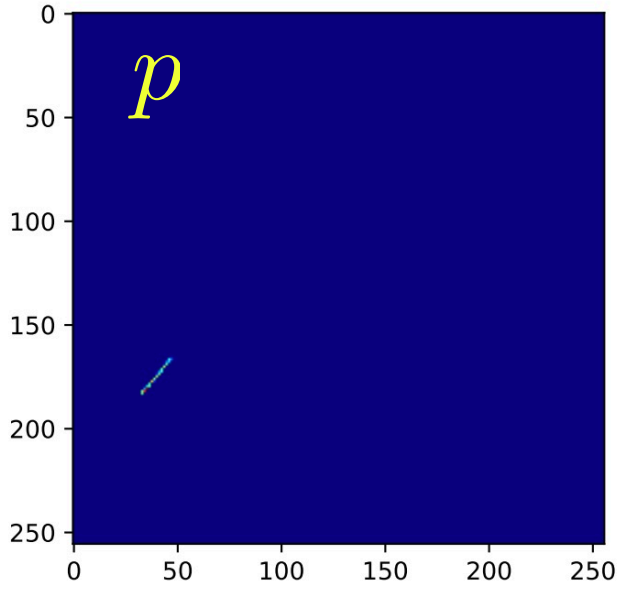
Training data examples



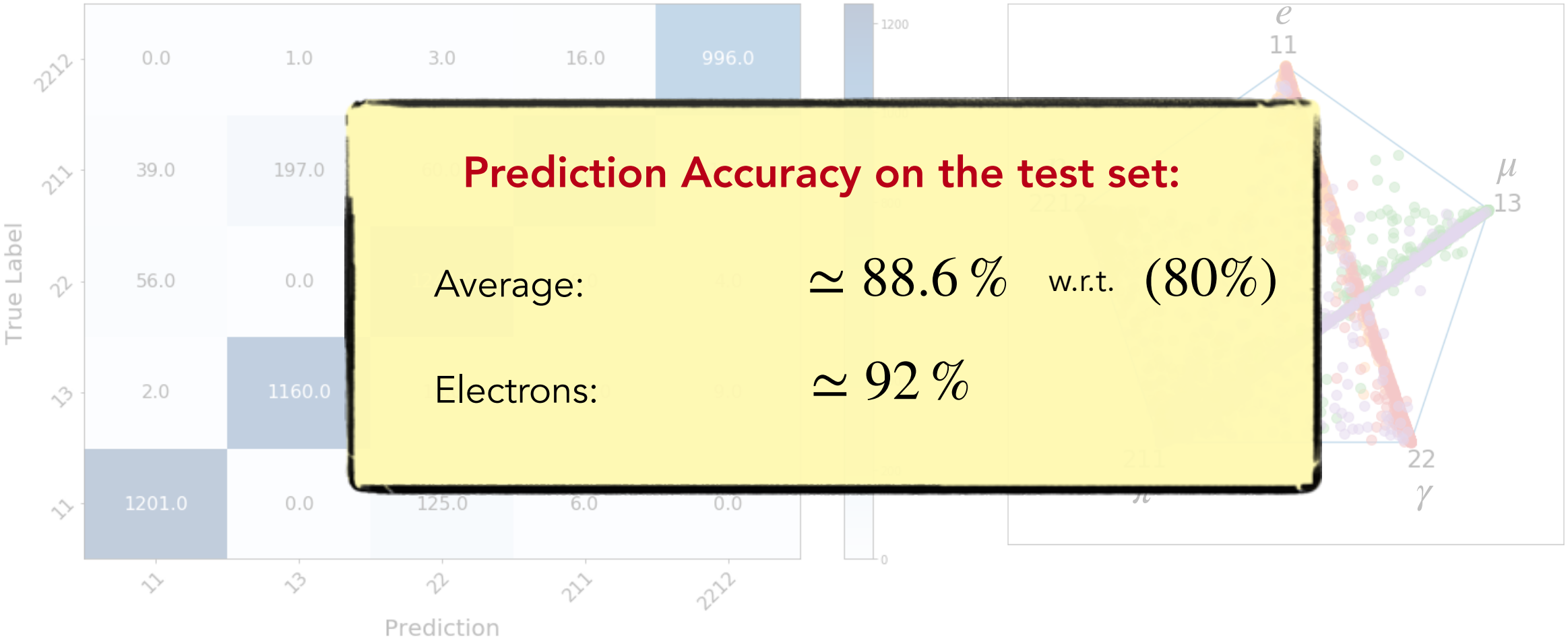








Performances and Results



Prediction Accuracy on the test set:

Average: $\simeq 88.6\,\%$ w.r.t. (80%)

Electrons: $\simeq 92\,\%$

...Claiming a $\nu_\mu \to \nu_e$ Oscillation Discovery!

Assuming: Electron events only contaminated by mis-identification of muon events

Given $N^{tot}_{\nu_\mu} = 10^6$ events, a 5σ observation requires:

Significance $$Z = \frac{n_s}{\sqrt{n_s + n_b}} \geq 5$$

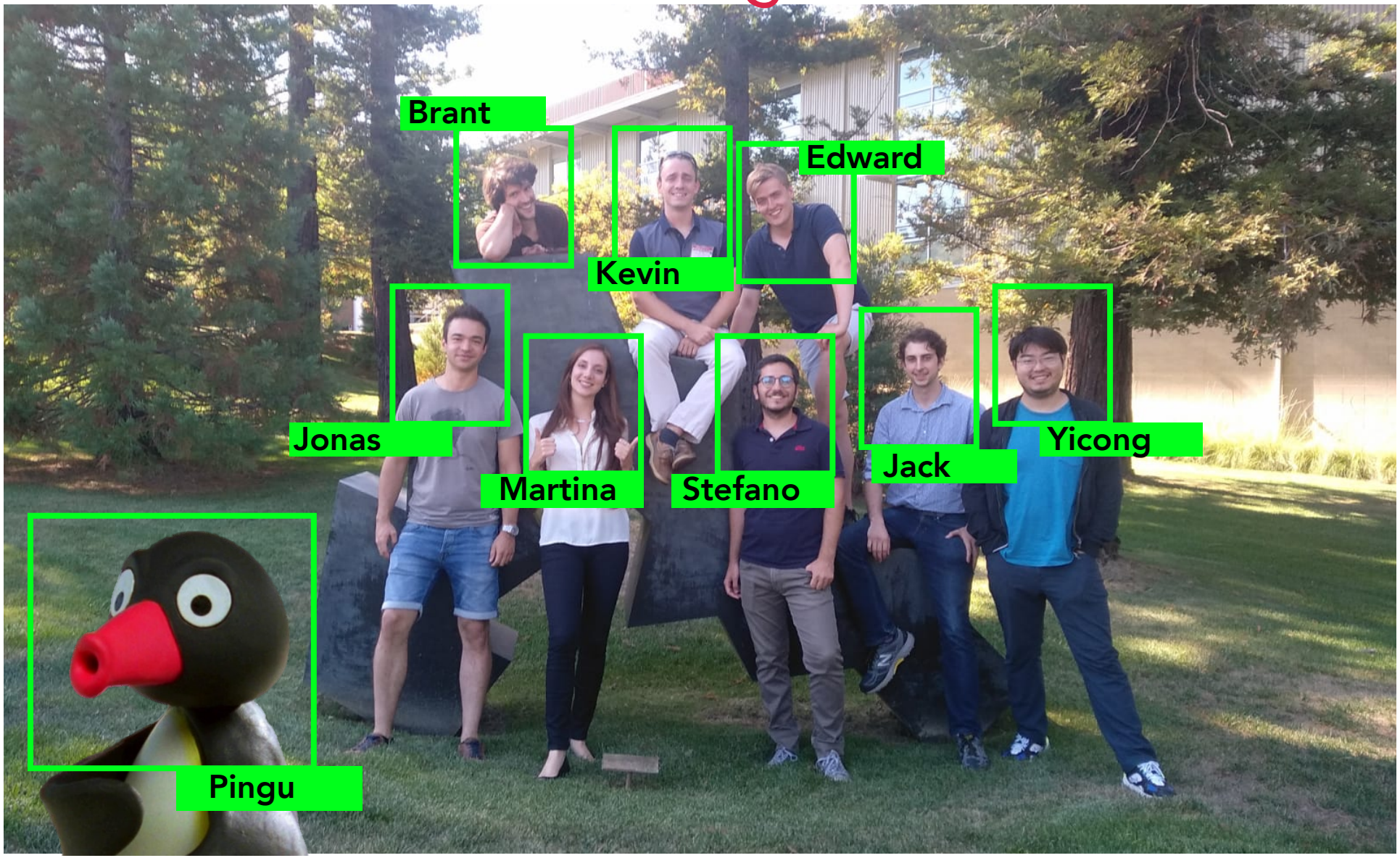
where $n_s = N^{tot}_{\nu_e} \cdot \epsilon_{ee}$ $n_b = N^{tot}_{\nu_\mu} \cdot \epsilon_{\mu e}$ **from our DNN**

→ For $\epsilon_{ee} \approx 0.92$ and $\epsilon_{\mu e} \approx 0.001$, a 5σ *oscillation discovery* requires **128 observed ν_e** events

Summary and Conclusions

- Used a **Deep Neural Network** for **PID** in a LArTPC simulation
- **Optimised** the DNN by improving the **architecture** and tuning the **hyper-parameters**
- Significant **<accuracy> improvement**: 80% to 88.6%
- Determined the corresponding **number** of ν_e events required for a 5σ **discovery** in neutrino oscillation experiment: **128**

Thank you!


Brant
Kevin
Edward
Jonas
Martina
Stefano
Jack
Yicong
Pingu

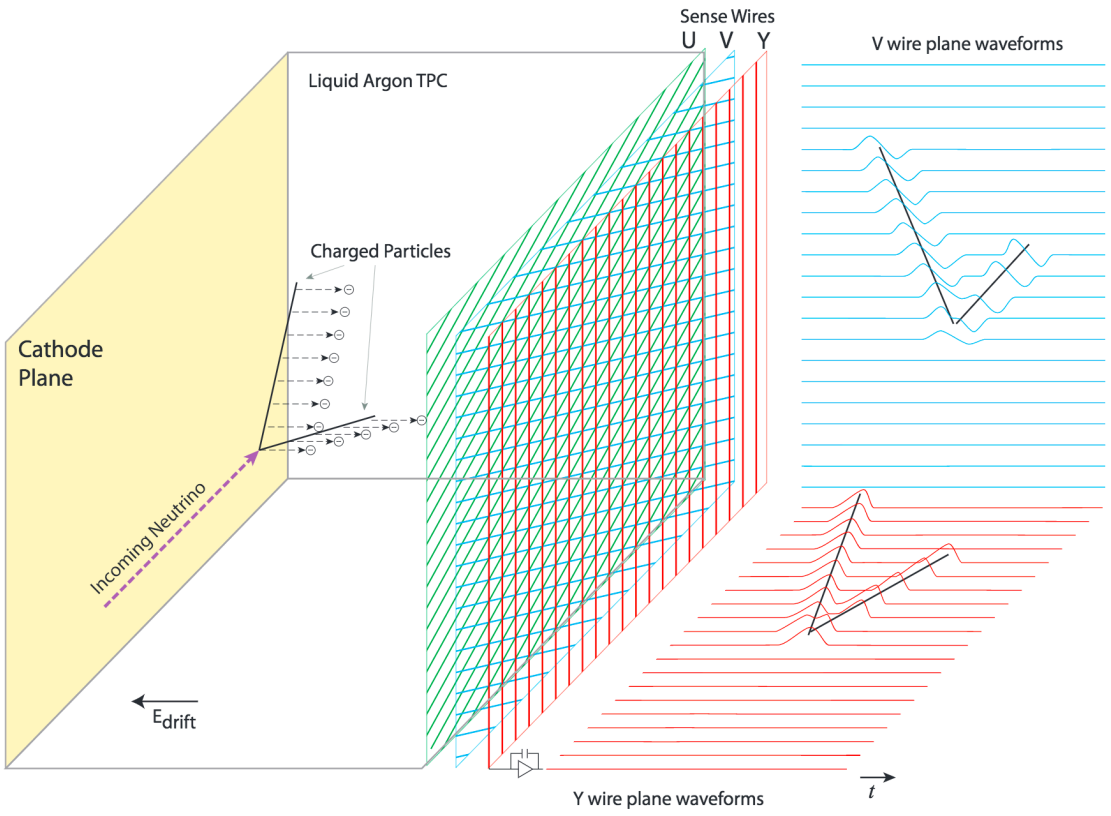
Backup

LArTPC - Liquid Argon Time Projection Chamber

- High-mass detector for neutrinos
- Provides 3D tracking and dE/dx measurement
- High-density medium makes amplification unnecessary
- Used by MicroBooNE and to be used as part of the DUNE experiment

Basic concept:

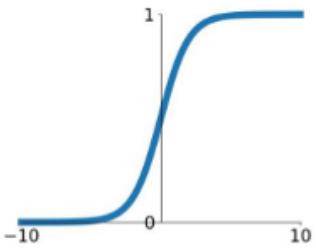
1. Particle enters fiducial volume & ionizes medium
2. Primary ionization drifts in uniform electric field to readout plane
3. Readout plane stores ionization magnitude & position in 2D (xy projection)
4. Drift dimension (z) reconstructed using drift velocity and trigger time



Machine Learning Activation Functions

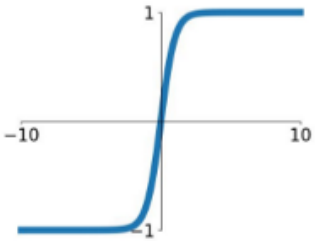
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



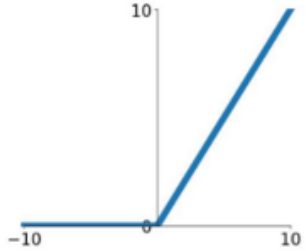
tanh

$$\tanh(x)$$



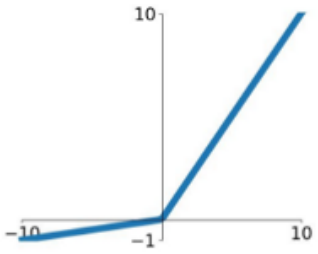
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

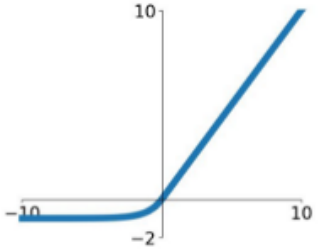


Maxout

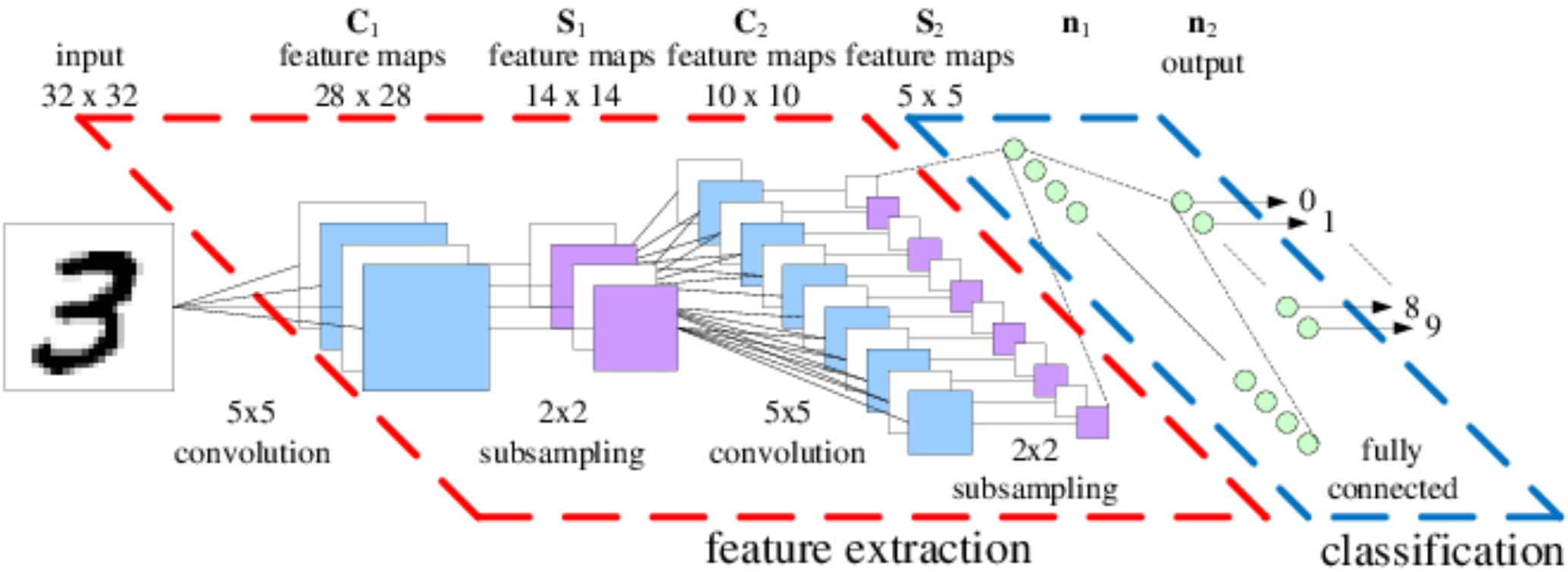
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

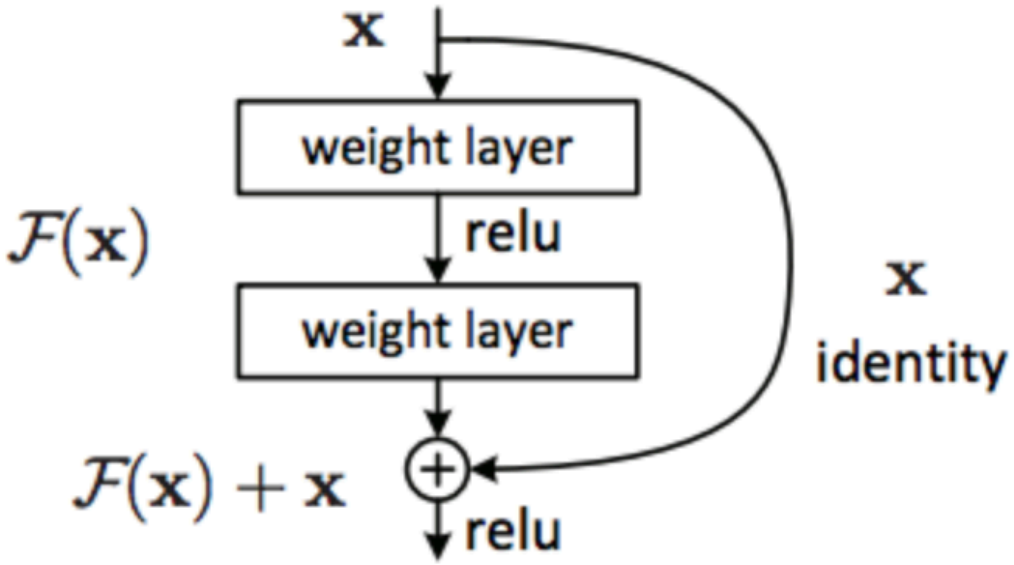
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Convolutional Neural Networks (CNNs)



Residual Networks



$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}.$$