LAUREA MAGISTRALE IN FISICA

# Particle Tracking through Machine Learning

Supervisor: prof. Tessera Daniele
Co-supervisor: prof. Borgonovi Fausto

Nicola Vanoli

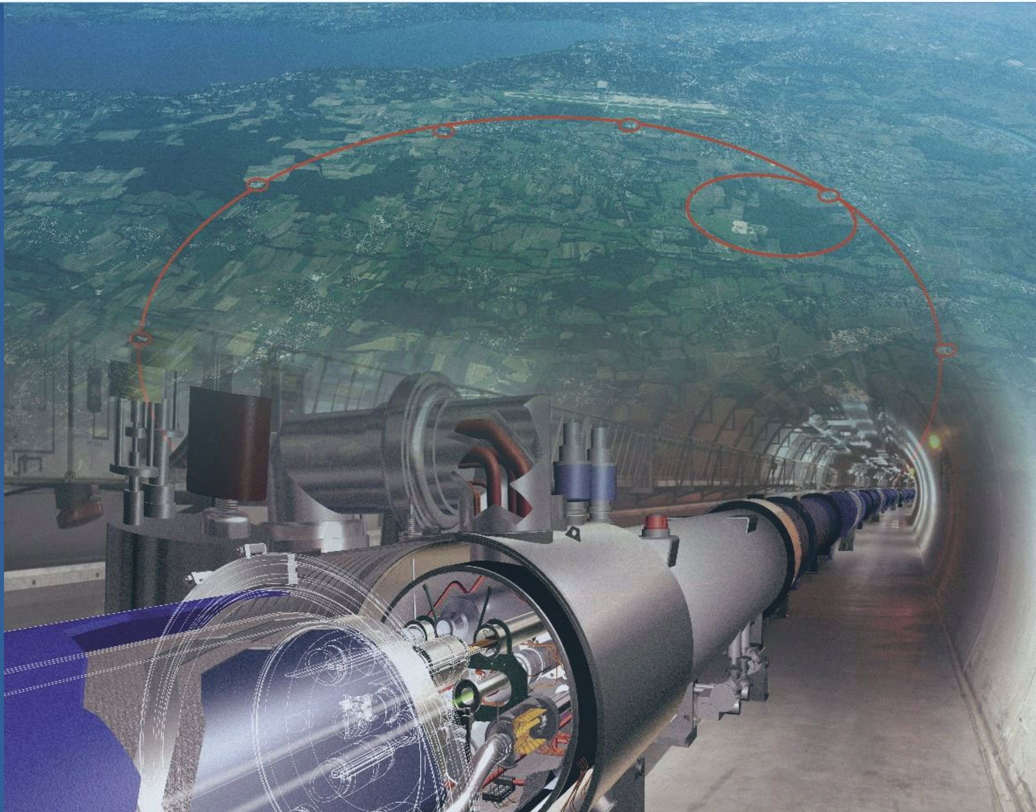
A.A. 2019/2020

# Outline

- Introduction and Motivation
- The ATLAS experiment at CERN
- Deep Neural Network's architecture
- Results and Discussion
- Conclusion

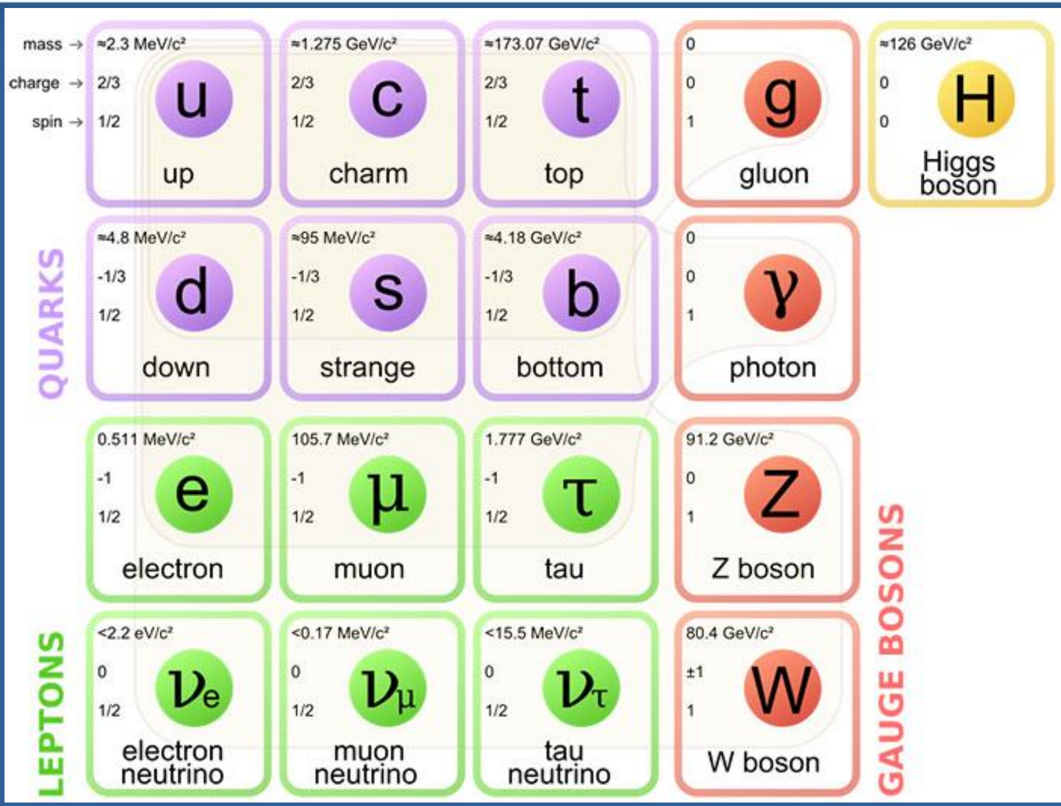
# Introduction and Motivation

- Large Hadron Collider (LHC)
- Huge amount of data
- The tracking problem
- A Machine Learning approach



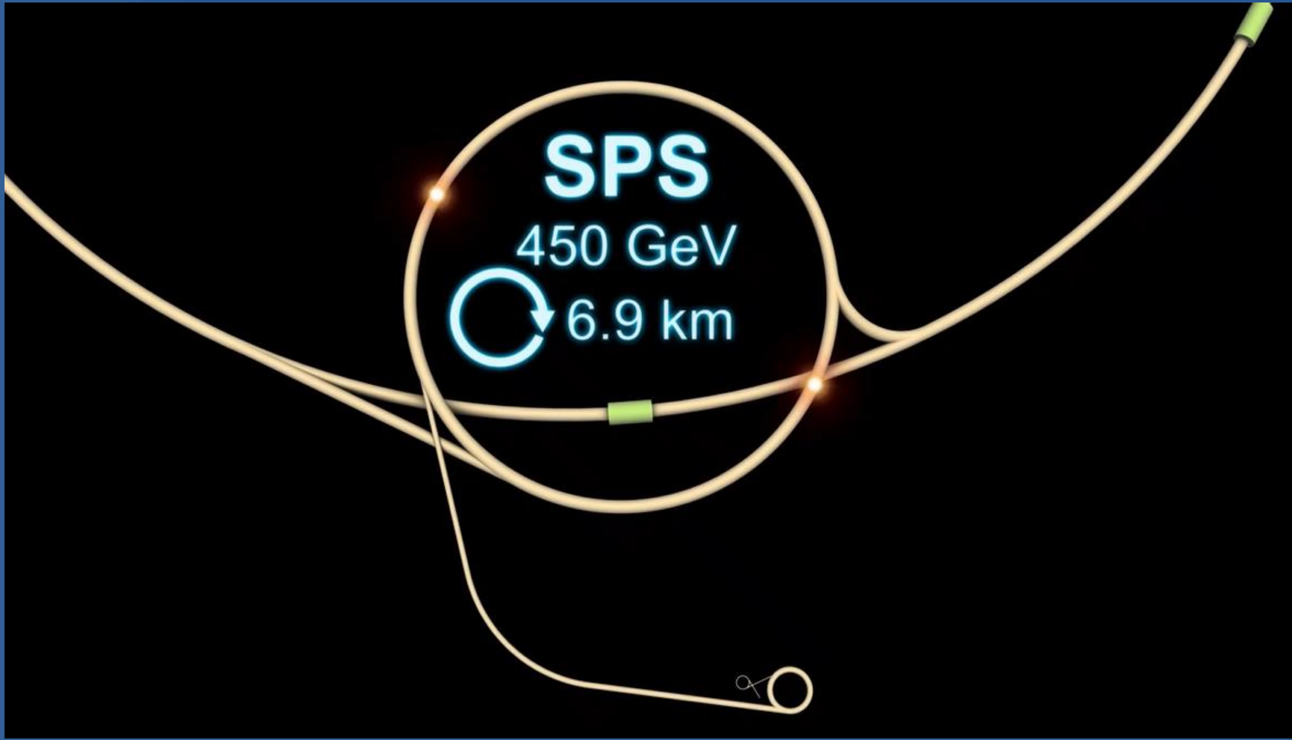
# The Standard Model

- It describes three of the four fundamental forces
- It classifies all known elementary particles
- Protons and neutrons are made up of up and down quarks
- It does not explain gravity and dark matter



# Proton-Proton Collision

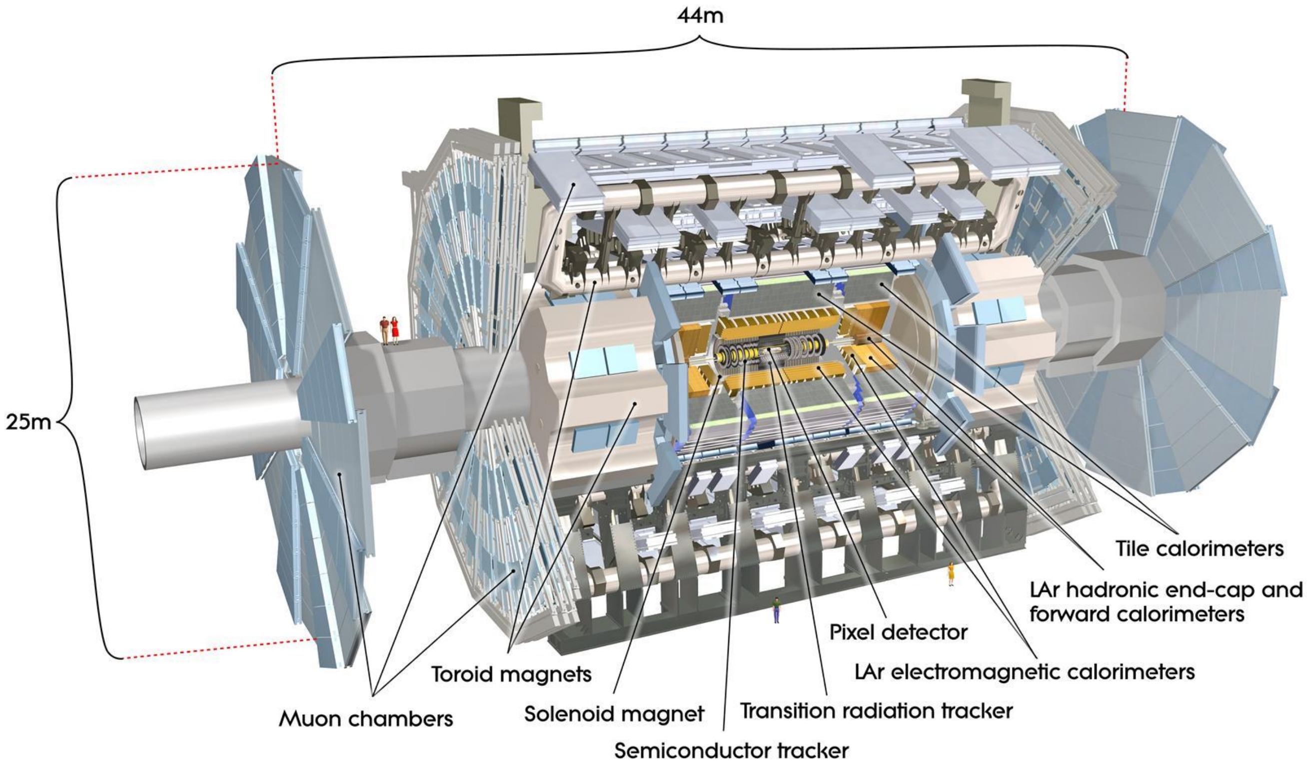
- Protons accelerate through the LHC
- After colliding many different particles are generated
- Particles pull away from the centre



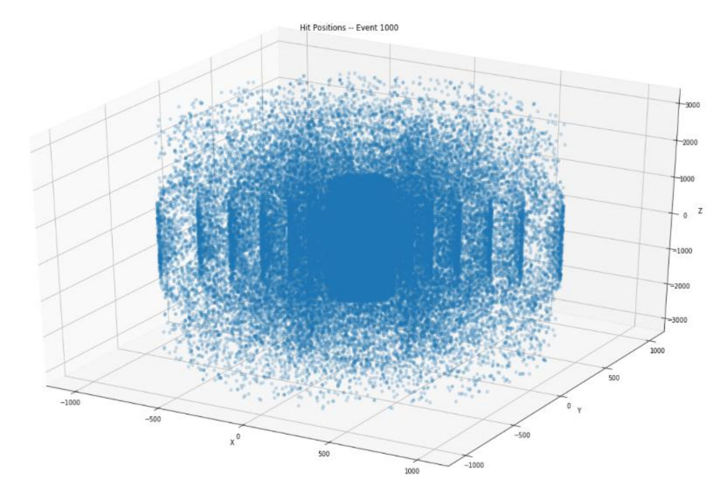

https://www.youtube.com/watch?v=NhXMXiXOWAA&ab_channel=Seeker

# ATLAS

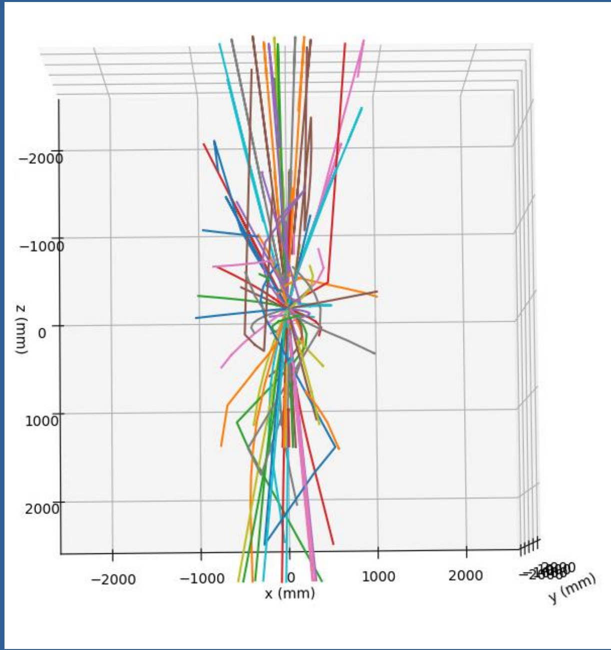
- Protons collide in the middle
- Pixel detectors record the passage of particles
- Pixel detectors record the amount of energy lost by the particle



# Track Particle



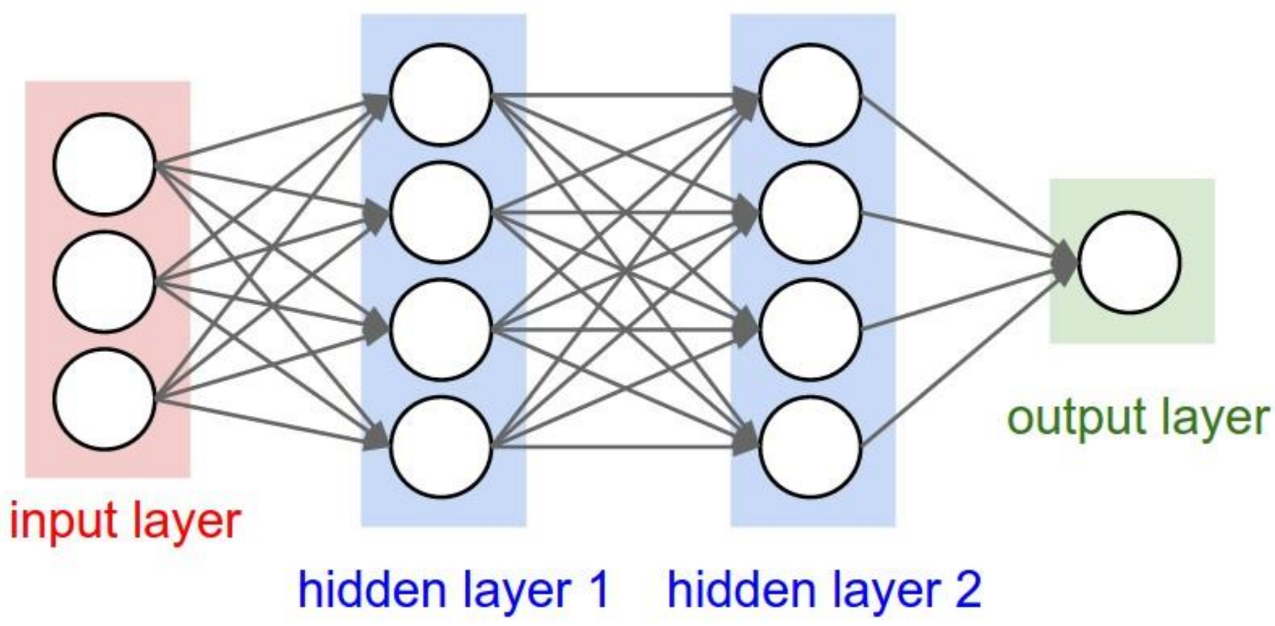

How to reconstruct all trajectories?




Input: dataset recorded by detectors

Output: predicted trajectories by the DNN

# Neural Network




Flow of activation

# Training Process

### Target Function

$$y = f^*(\underline{x}) \ , \ \underline{x} \in \mathbb{R}^d$$

### Approximating Function

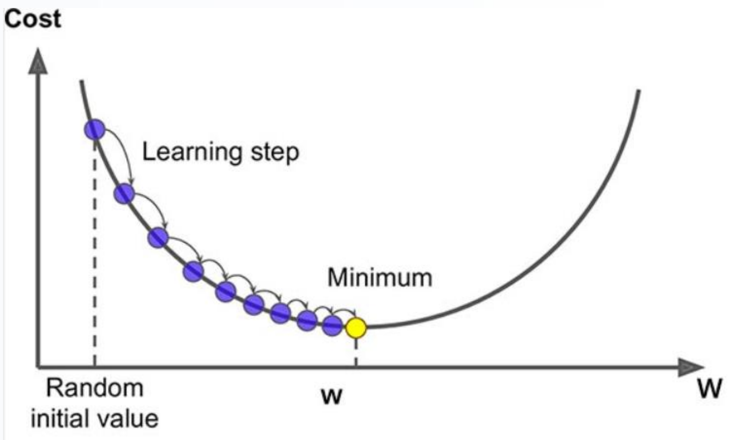
$$\tilde{y}(\underline{x}) = \underline{w} \cdot \underline{x} + b \ , \underline{w} \in \mathbb{R}^d \ , \ b \in \mathbb{R}$$

### Loss Function

$$L(\underline{x}^{(i)}, y^{(i)}) := (\tilde{y}(\underline{x}^{(i)}) - y^{(i)})^2$$

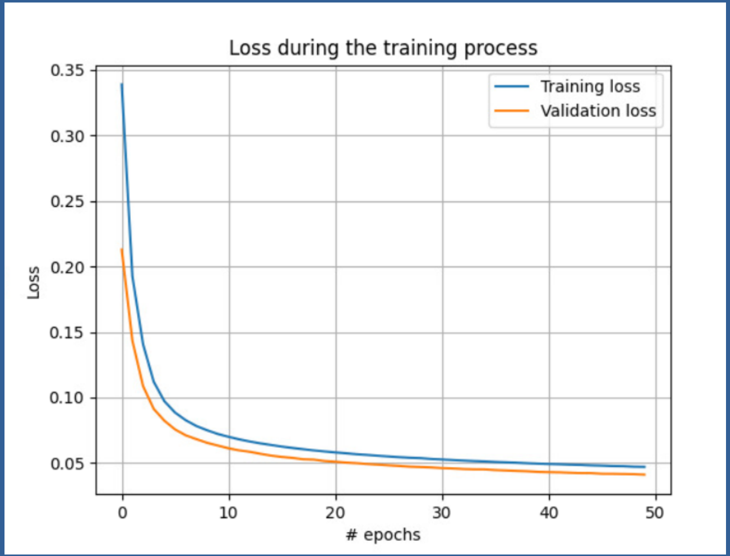
### Best Parameters

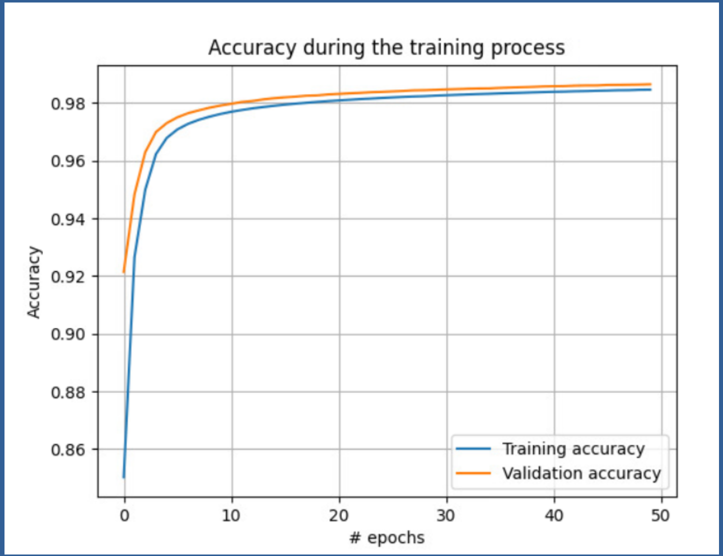
$$\theta^* := argmin_{\Theta} L(D, \theta)$$



# Training vs Validation

Validation set is often used to estimate how well the model has been trained



# Tracking Particles' Neural Network

- Input: couple of points
- Number of neurons: 2200
- Number of layers: 5
- Output: 1 if points belong to the same trajectory, 0 otherwise

Points are gathered in lists

Each list describes a TRAJECTORY

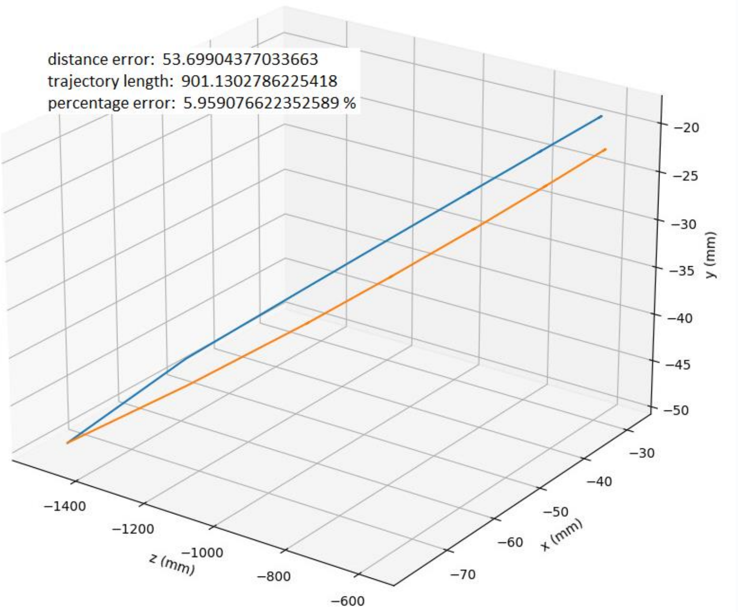
# Error Measure

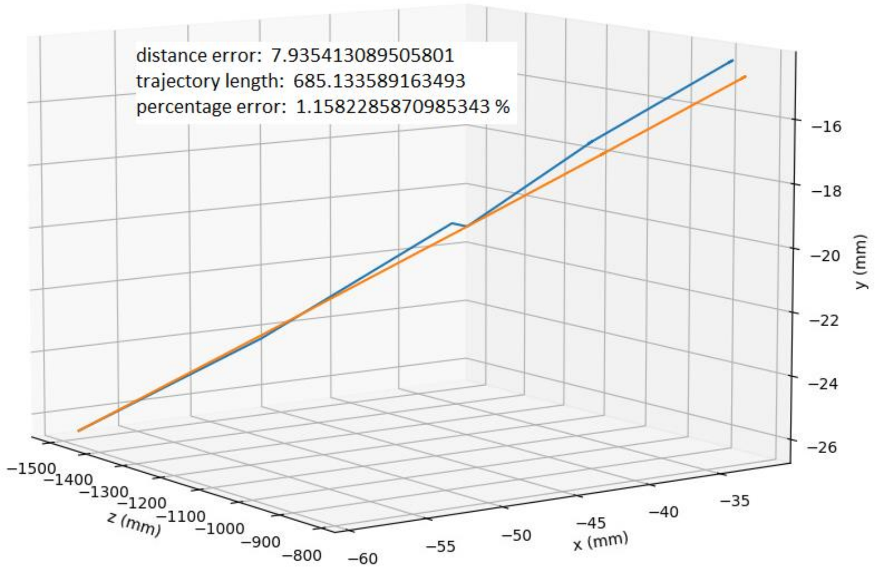
How do we determine if the predicted trajectory is good?

ERROR = sum of all minimum distances between reconstructed points and real trajectory

$$\text{PERCENTAGE ERROR} = \frac{ERROR}{LENGTH\ OF\ TRAJECTORY}$$

# Reconstructed vs Real trajectory




BLUE= reconstructed

ORANGE = real

# Models

| Name | Event A (%) | Event B (%) | Event C (%) | Event D (%) | Average (%) |
|---|---|---|---|---|---|
| **basic10** | 31,19 | 29,02 | 24,03 | 22,52 | 26,69 |
| **100ev_32k** | 4,79 | 12,24 | 7,29 | 5,79 | 7,52 |
| **100ev_2k** | 5,56 | 11,11 | 8,62 | 5,09 | 7,59 |
| **best80** | 2,99 | 4,49 | 6,88 | 1,86 | 4,05 |

4 different model applied to 4 different collisions' events produce the above results

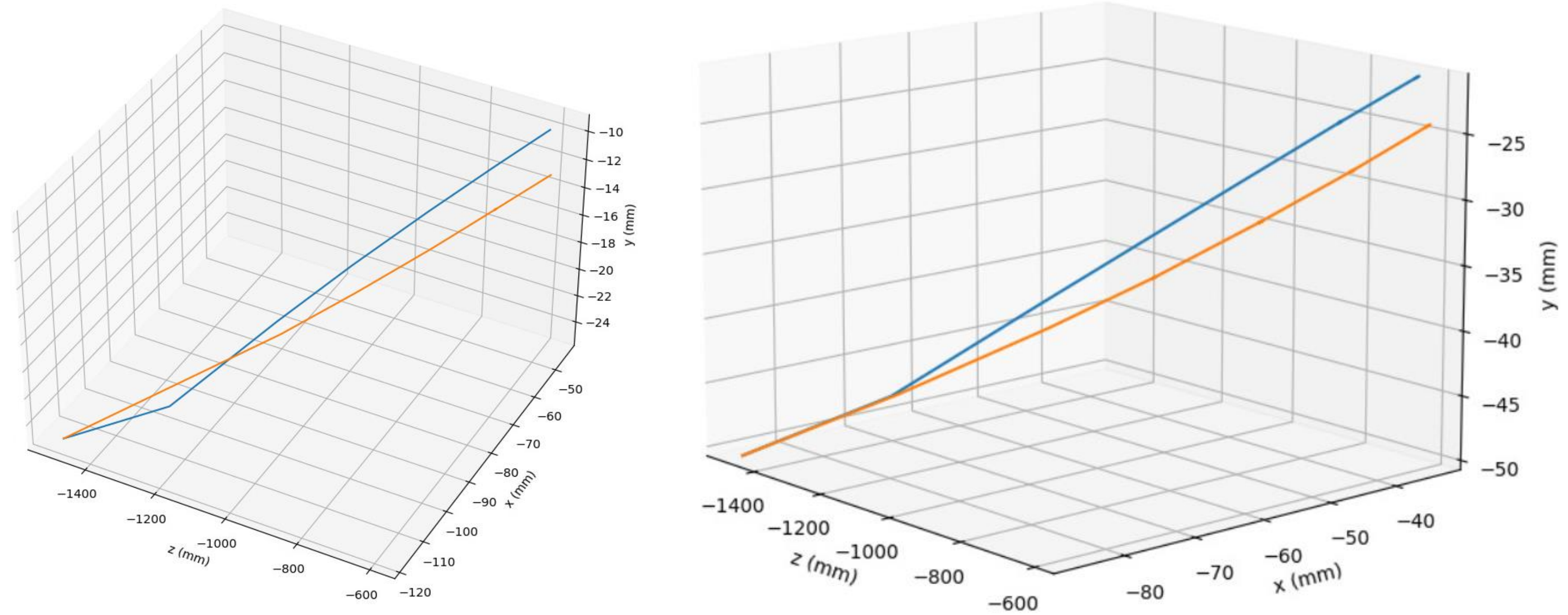
# Models

| Name | Event A (%) | Event B (%) | Event C (%) | Event D (%) | Average (%) |
|---|---|---|---|---|---|
| **basic10** | 31,19 | 29,02 | 24,03 | 22,52 | 26,69 |
| **100ev_32k** | 4,79 | 12,24 | 7,29 | 5,79 | 7,52 |
| **100ev_2k** | 5,56 | 11,11 | 8,62 | 5,09 | 7,59 |
| **best80** | 2,99 | 4,49 | 6,88 | 1,86 | 4,05 |

Why is this the best model?

Difficult to answer; unbalanced dataset could be an explanation

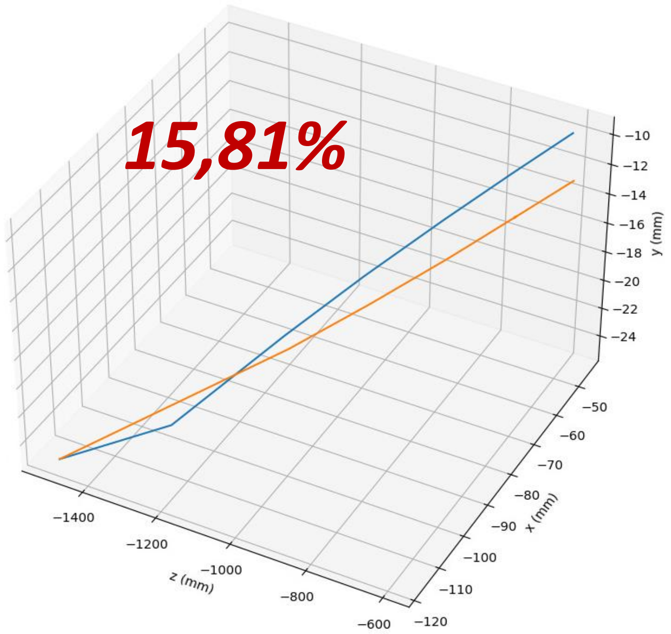
# Which one has the lowest error?

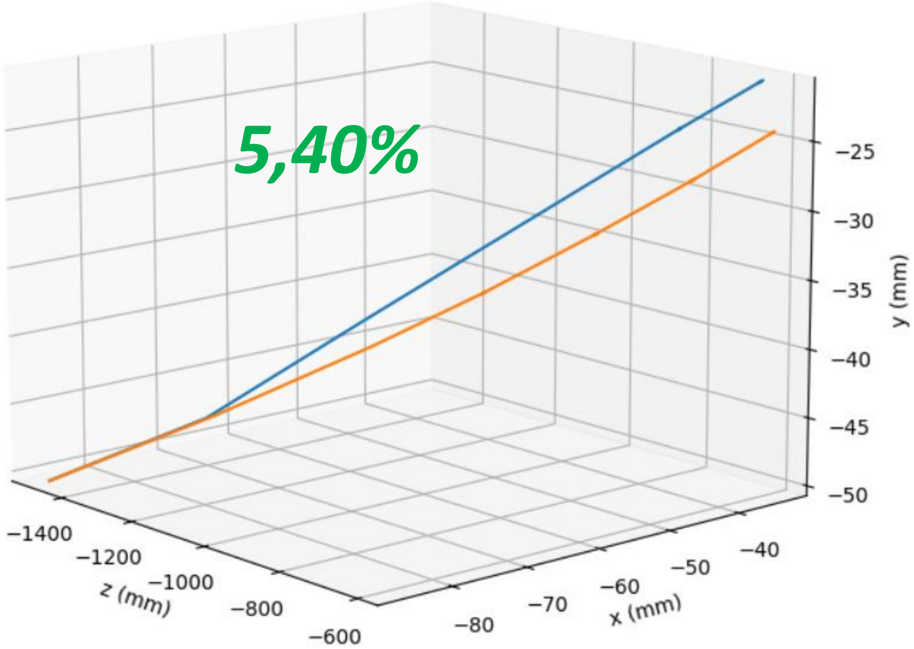



BLUE= reconstructed

ORANGE = real

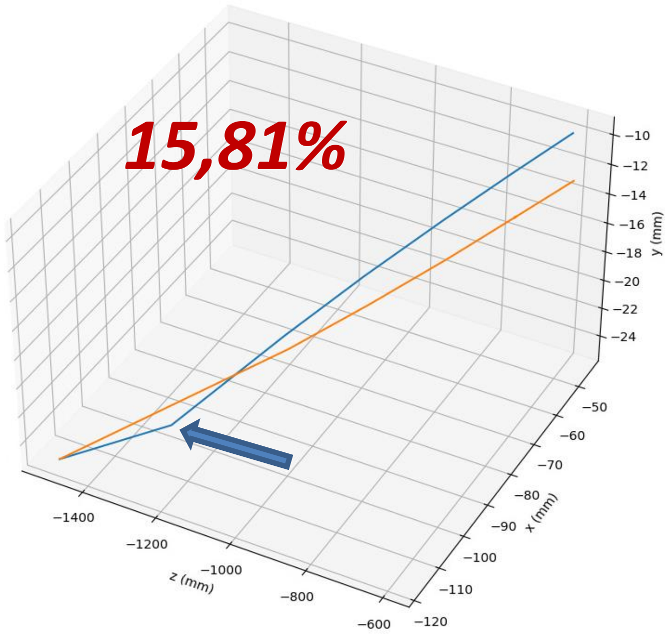
# Average error is 4,05%

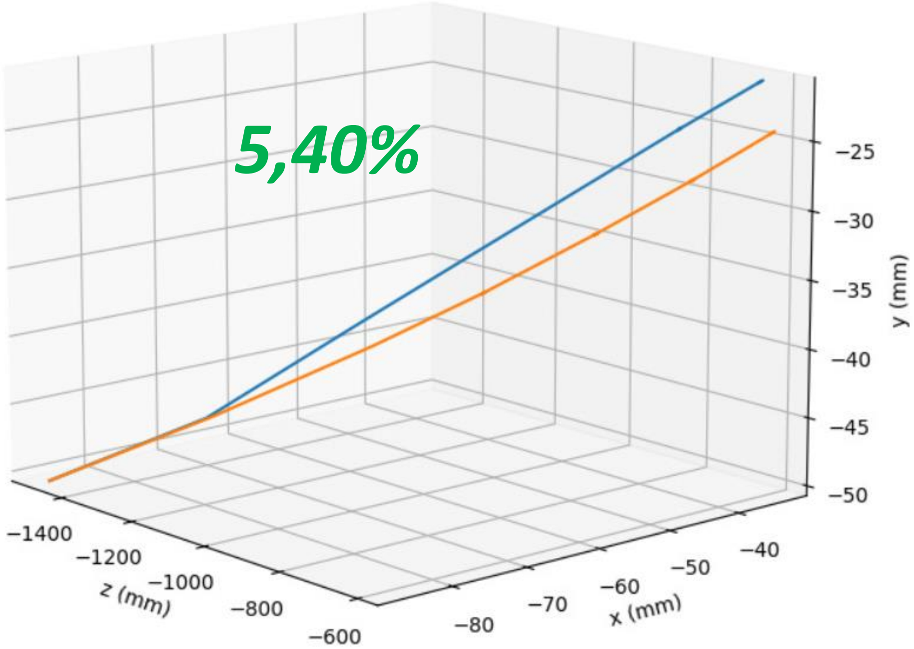



BLUE= reconstructed
ORANGE = real

# Model's predictions




Corner points should not be observed in real trajectories

# Conclusion & Future Works

We demonstrated how DNN can be useful for particles tracking problems, with errors lower than 5%.

Next Steps are:

- Prevent the neural network from violating physical laws
- Assert that each prediction does not violate any conservation law
- Try to estimate the charge of the particles