Reconstruction of electromagnetic showers in calorimeters using Deep Learning

Polina Simkina

Fabrice Couderc, Julie Malclès, Mehmet Özgür Şahin

CEA IRFU

26ème Congrès Général de la SFP
06/07/2023

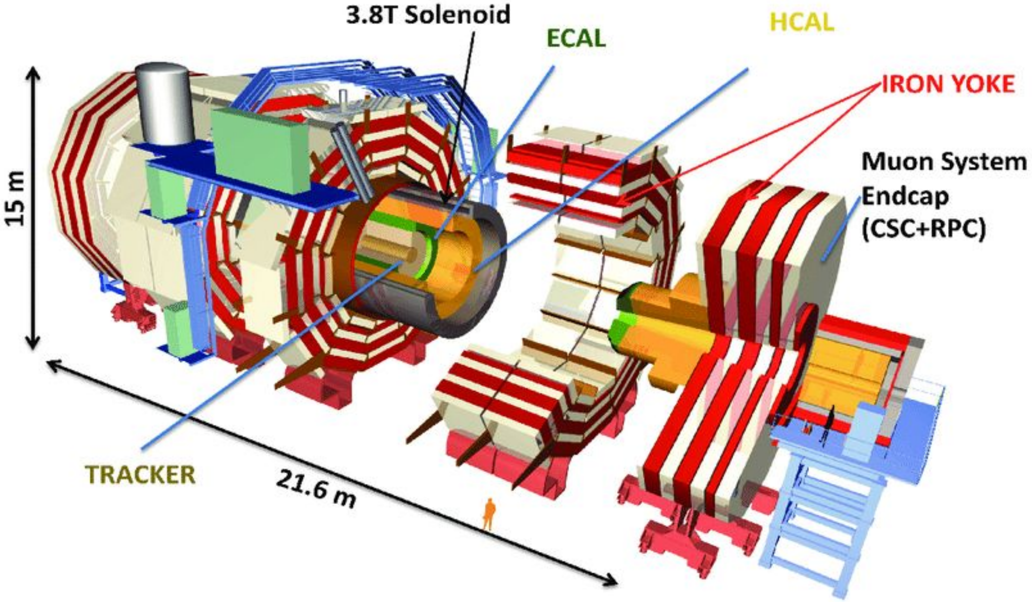
Introduction

CMS experiment

Discovery of the **Higgs boson** in 2012 (along with ATLAS).

Physics scope: probe **standard model** and search for **physics beyond standard model**.

Uses **proton-proton collisions** at the center of mass energy from 7 TeV to 13.6 TeV.



Electromagnetic CALorimeter

Homogeneous calorimeter.

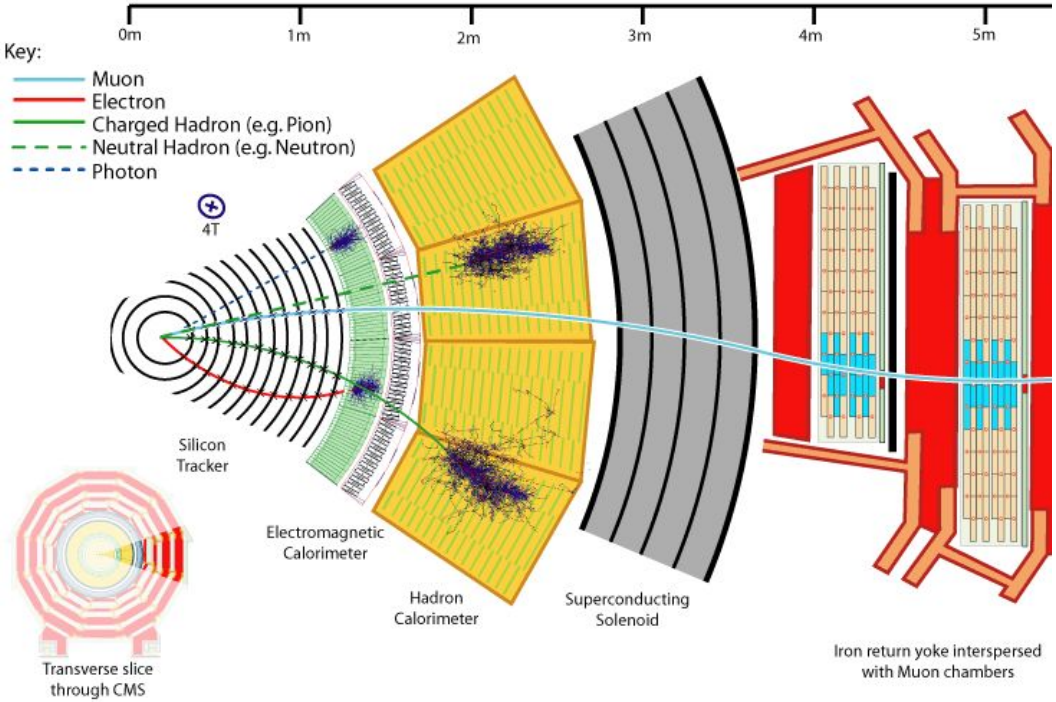
Around 76 000 **$PbWO_4$ crystals.**

Mainly used for the reconstruction of **electrons** and **photons.**

Plays crucial role for **all physics analysis**, e.g. for Higgs decay channels:

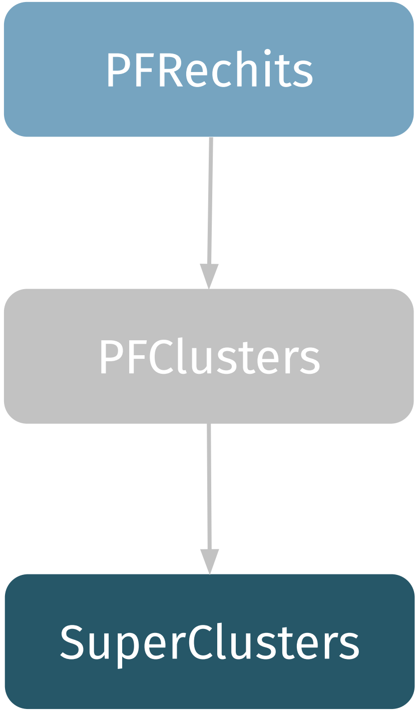
$$H \to \gamma\gamma$$

$$H \to ZZ^{*} \to 4\ell$$



0m
1m
2m
3m
4m
5m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon
4T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
Transverse slice through CMS

EM object reconstruction in ECAL



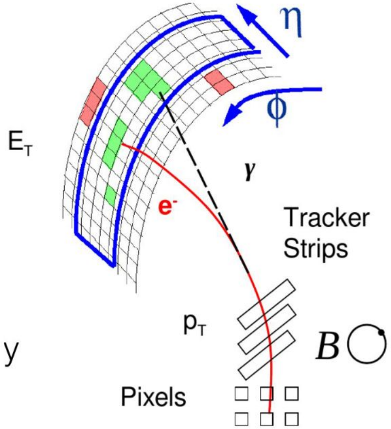
PFRechits

PFClusters

SuperClusters

- **Energy deposits** left by particles in the PbWO4 crystals of the calorimeter.

- **PFRechits** are gathered together to form a PFCluster that represents a **single particle**.
- **The subject of this talk!**

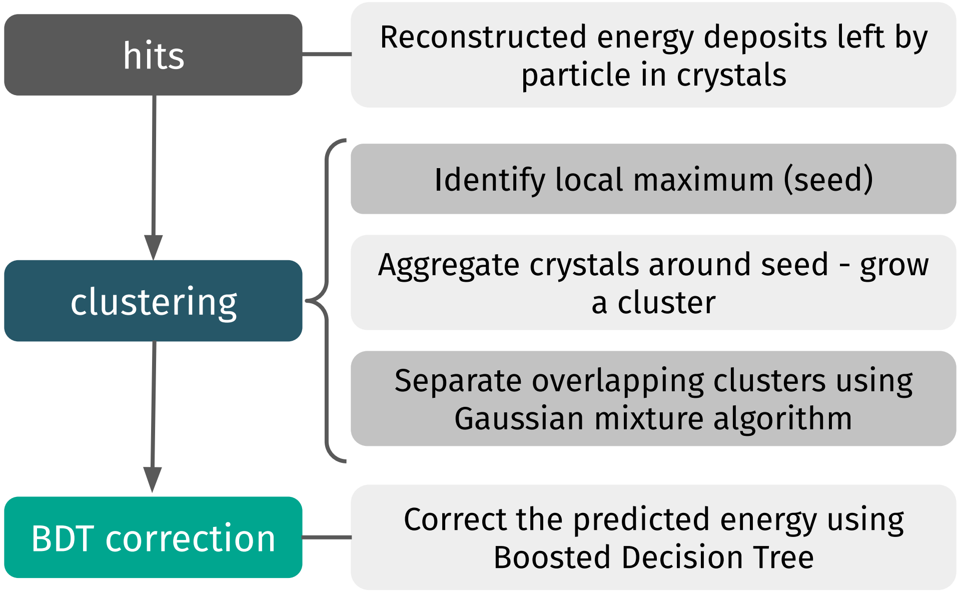


- Because of bremsstrahlung or photon conversion before the ECAL, PFClusters have to be combined to form a **SuperCluster**.
- Currently a geometrical algorithm (Mustache) is used, a Boosted Decision Tree is applied for energy correction.
- New ML-based **DeepSC algorithm** was developed and is currently tested in CMSSW.

PFCluster reconstruction

Reconstruct **position and energy** of electrons and photons from **electromagnetic showers**.

PFClustering algorithm:



hits

Reconstructed energy deposits left by particle in crystals

clustering

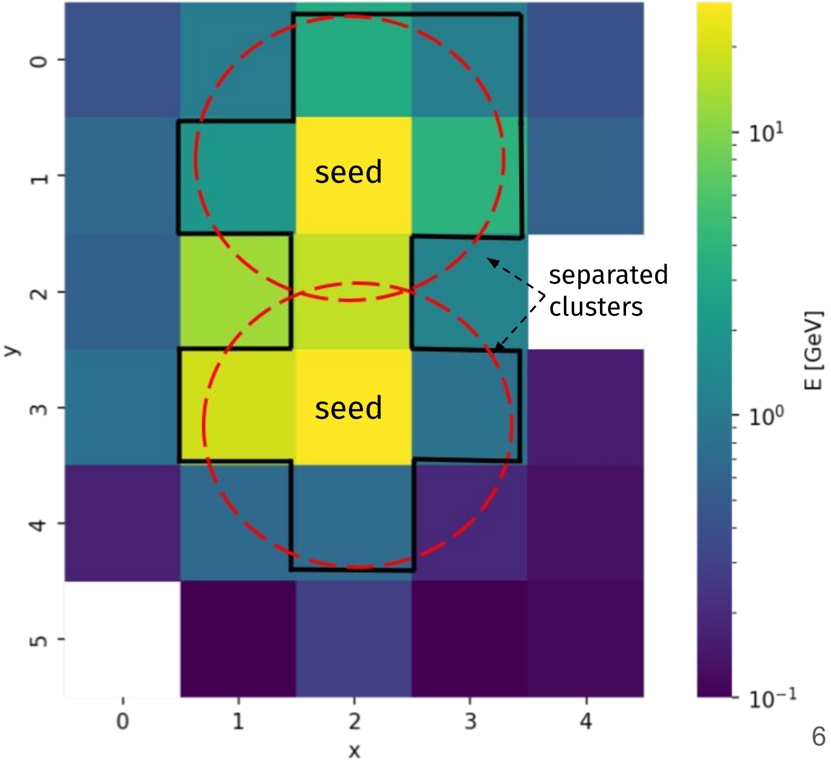
Identify local maximum (seed)

Aggregate crystals around seed - grow a cluster

Separate overlapping clusters using Gaussian mixture algorithm

BDT correction

Correct the predicted energy using Boosted Decision Tree



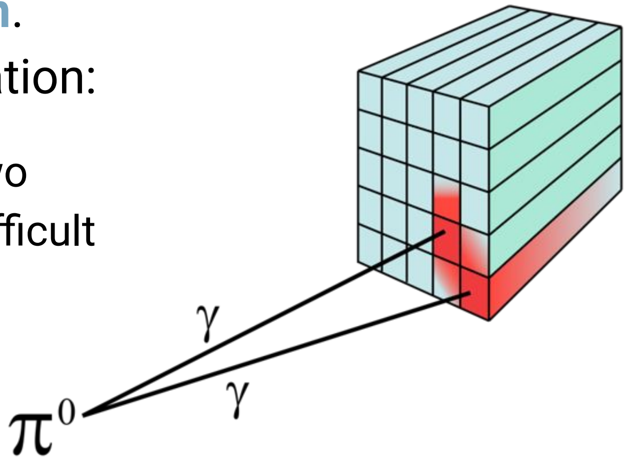
Motivation

Creating a novel **ML-based algorithm** for ECAL PFClustering reconstruction.

Main objectives:

- ➔ Improving **energy** and **coordinates resolution**.
- ➔ Improving **photon vs. neutral pion** discrimination:

  Photons coming from neutral pion decay create two overlapping clusters in the calorimeter, which is difficult to discriminate from a single photon's signature.



Source: science 2.0

Simulation

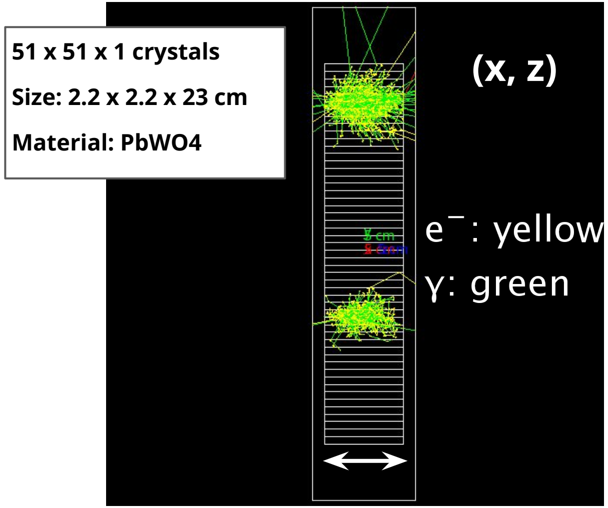
Detector simulation

Simplified calorimeter simulated in Geant4 to test the performance of the algorithms.

Same crystal characteristics as in real ECAL (but not tilted), no magnetic field or material in front of the calorimeter.

The crystals of the calorimeter detect the deposited energy and an **energy smearing** is applied to emulate the readout of the real detector.

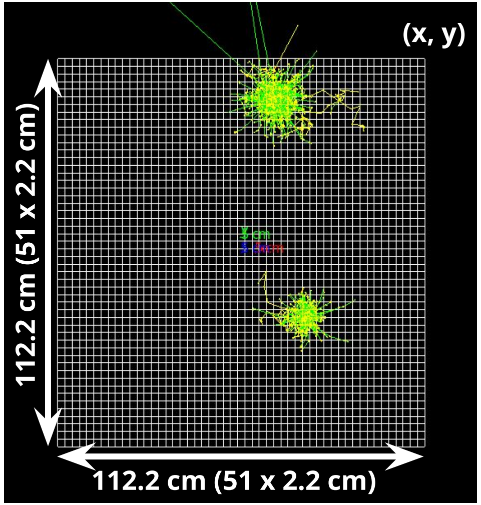
The simulation is compatible with the test beam results.



51 x 51 x 1 crystals

Size: 2.2 x 2.2 x 23 cm

Material: PbWO4



Energy smearing is done using Gaussian with μ = energy deposit in a crystal and σ taken from ECAL resolution formula.

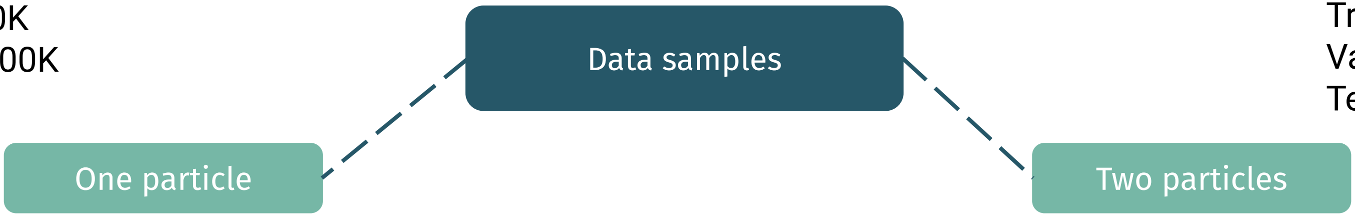
$$\left(\frac{\sigma}{E}\right)^2 = \left(\frac{a}{\sqrt{E}}\right)^2 + \left(\frac{\sigma_n}{E}\right)^2 + c^2$$

The parameters correspond to Run3 performance (a = 0.03 GeV$^{1/2}$, σ = 167 MeV, c = 0.0035).

Dataset creation

Photons with [1, 100] GeV energy (flat distribution) are used, directed perpendicularly to the calorimeter.

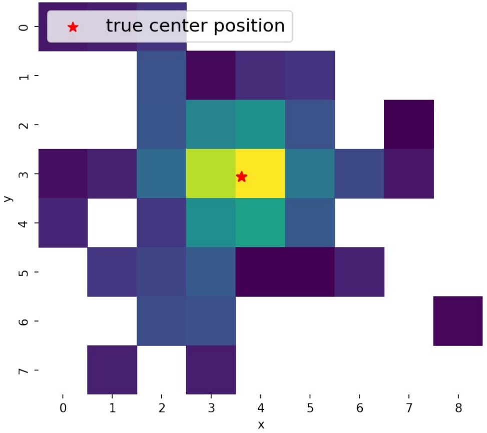
Training: 600K
Validation: 200K
Test: 100K

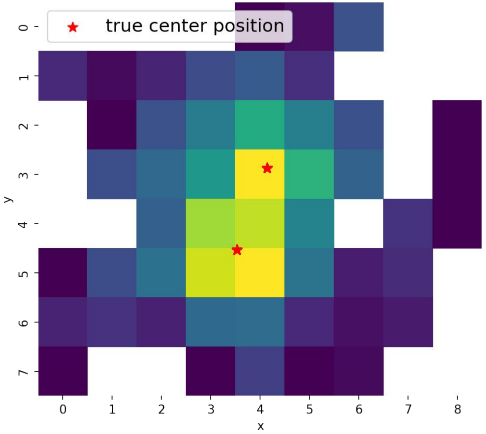

Data samples

One particle

Two particles

Training: 300K
Validation: 100K
Test: 50K





The dataset with two particles is created by superimposing one-particle events forcing the distance between two clusters to be < 3 crystals.

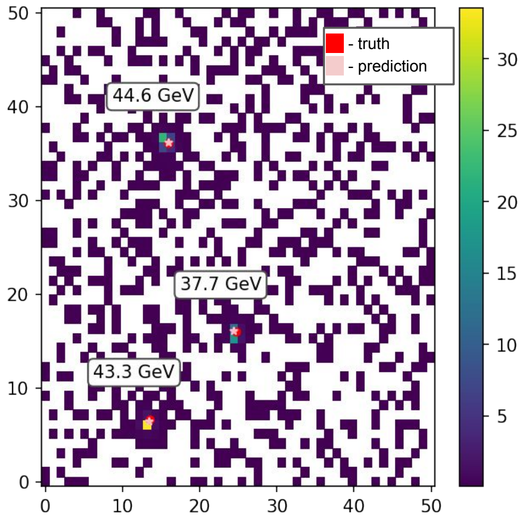
The model training is done using the mix of both data samples.

Traditional PFClustering algorithm is applied on the simulated samples, including a BDT energy regression.

Networks

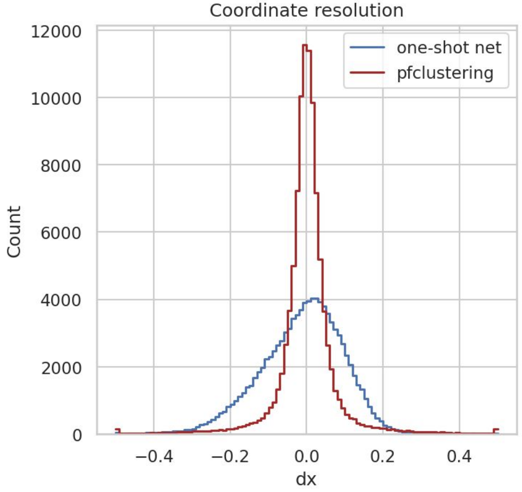
One-shot network

Energy deposits in crystals can be represented as **pixel intensities of an image** → allows to use **Convolutional Neural Network** (CNN).



First attempt:

- ➔ CNN applied on **full simulated detector** (dim: 51 x 51 crystals).
- ➔ Predict the **position (x, y)** of particles in the sample.
- ➔ Trained with the sample containing **up to 6 particles** per detector.

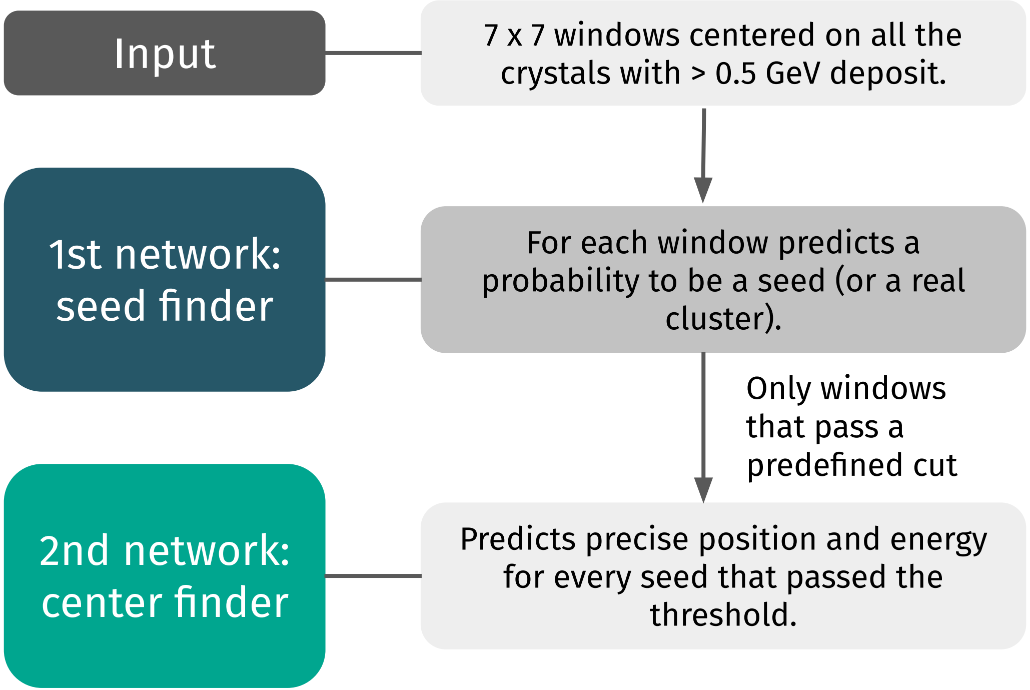


Results:

- ➔ One-shot network is **not able to predict precisely** number of particles and their position **simultaneously**, because of data sparsity and output ordering ambiguity.
- ➔ Not scalable to the complete ECAL detector (360 x 170 crystals in ECAL EB).

Two-step net – CNN: architecture

To solve these issues: separate the task into two CNN networks (with similar architecture).



Input

7 x 7 windows centered on all the crystals with > 0.5 GeV deposit.

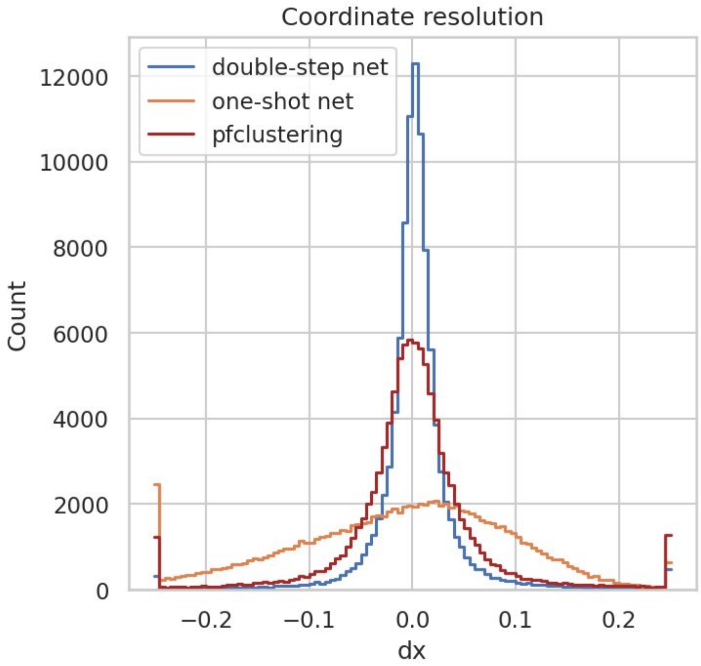
1st network: seed finder

For each window predicts a probability to be a seed (or a real cluster).

Only windows that pass a predefined cut

2nd network: center finder

Predicts precise position and energy for every seed that passed the threshold.



Improved resolution both for position and energy reconstruction. **This approach is also scalable.**

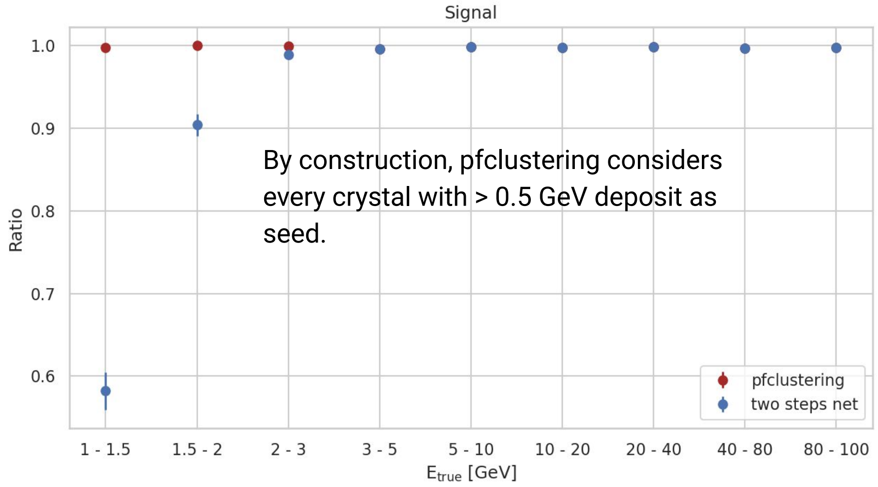
Two-step net – CNN: results

Each predicted cluster is matched to a true particle with the closest energy and position.

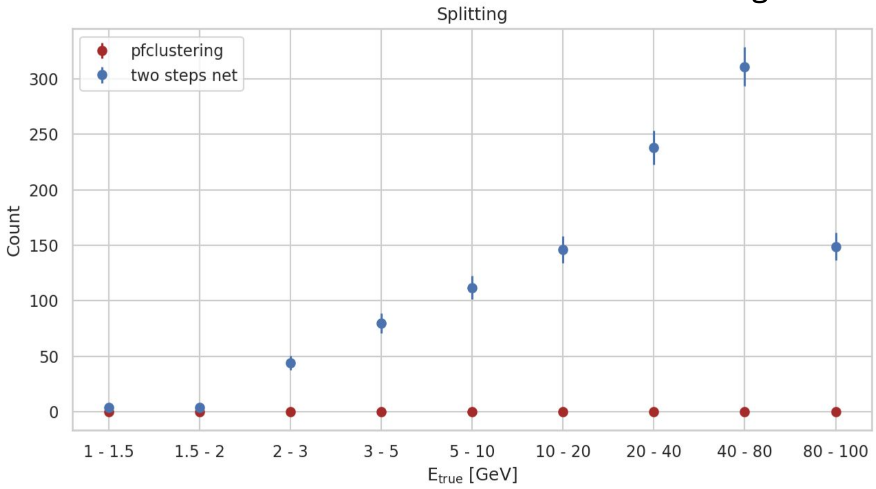
Signal efficiency – number of matched predicted clusters divided by the number of generated particles.
Splitting yield – number of events where one particle was reconstructed as two clusters.
Background yield – number of clusters reconstructed > 1.5 crystal away from true particle position.

One-particle sample

Results shown for
100K testing set



By construction, pfclustering considers every crystal with > 0.5 GeV deposit as seed.

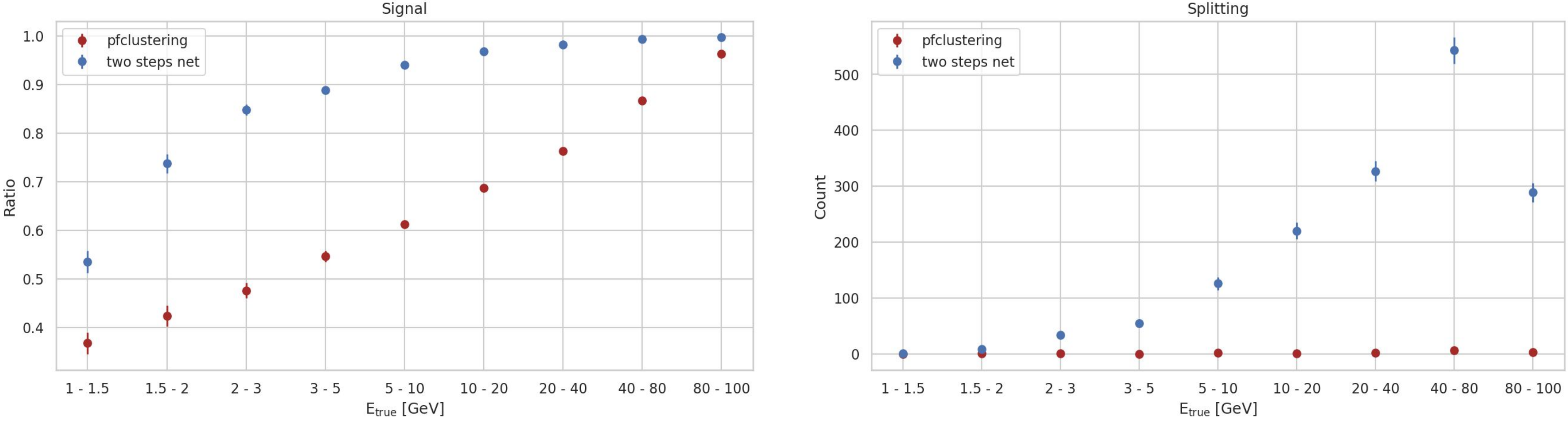


Splitting is created by what we call a "double-counting" problem - explained further.

Two-step net – CNN: results

Two-particle sample

Results shown for 50K testing set

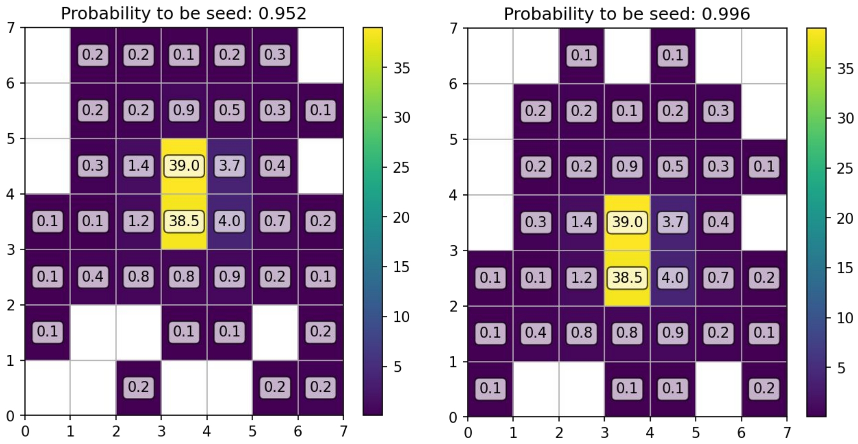


Efficiency for two-particle case is **much better for the network** compared to pfclustering, the performance is improving!

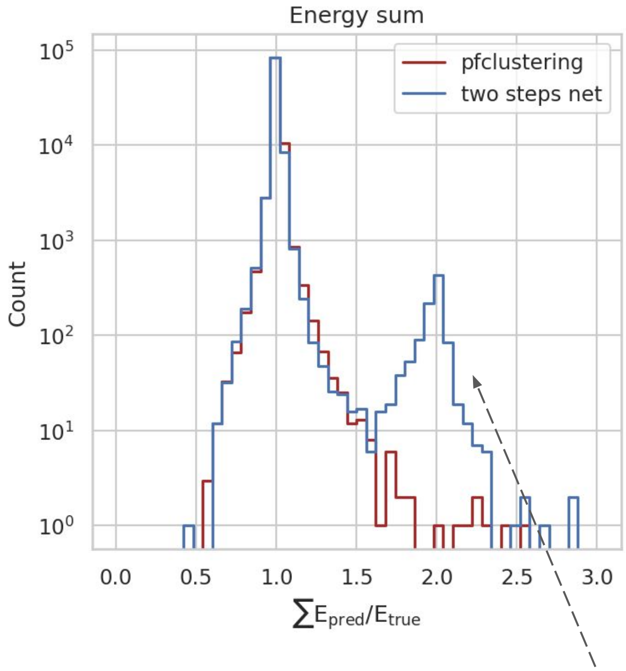
But same splitting problem as for one particle dataset.

“Double-counting” problem

Input windows do not communicate in the network → problem appears when a single-particle position is close to the crystal border:



A single particle creates two clusters with almost identical predicted positions and energies.



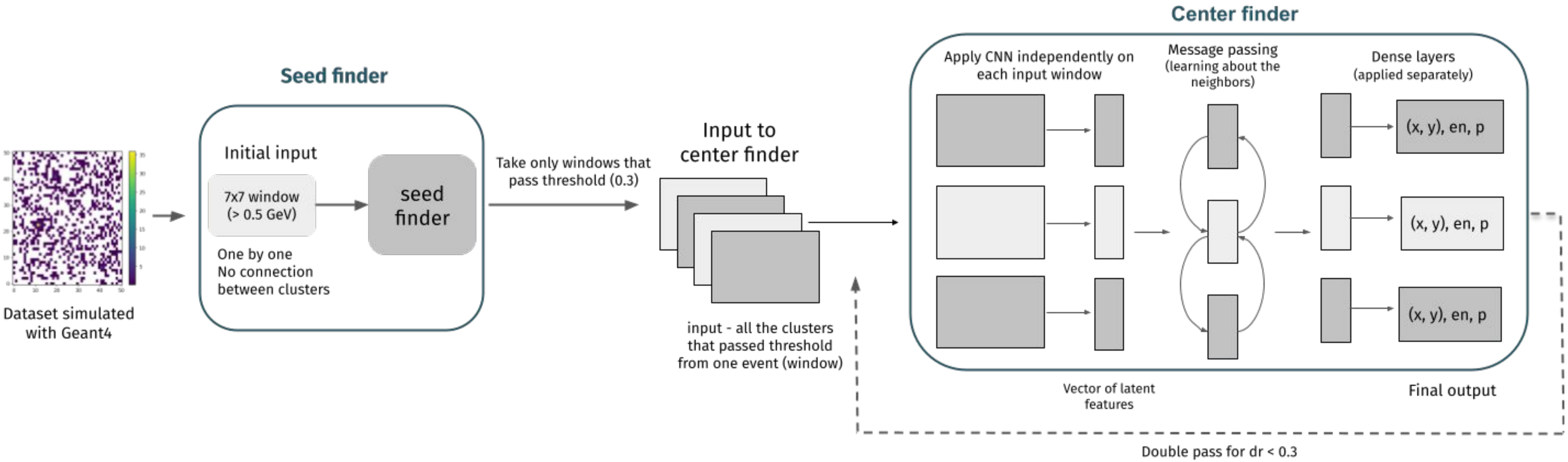
Creates a large energy overestimation.

The solution is to **use Graph Neural Network (GNN)**.

Two-step net – GNN: architecture

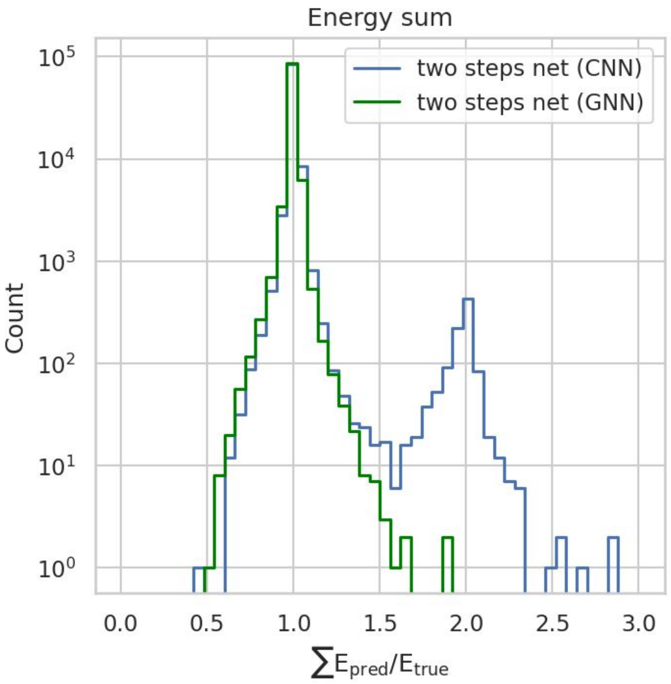
Solution: add communication between input windows.

Using **Graph Neural Networks** (GNN) and **Message-Passing** (MP) each window can **learn about its neighbors**.

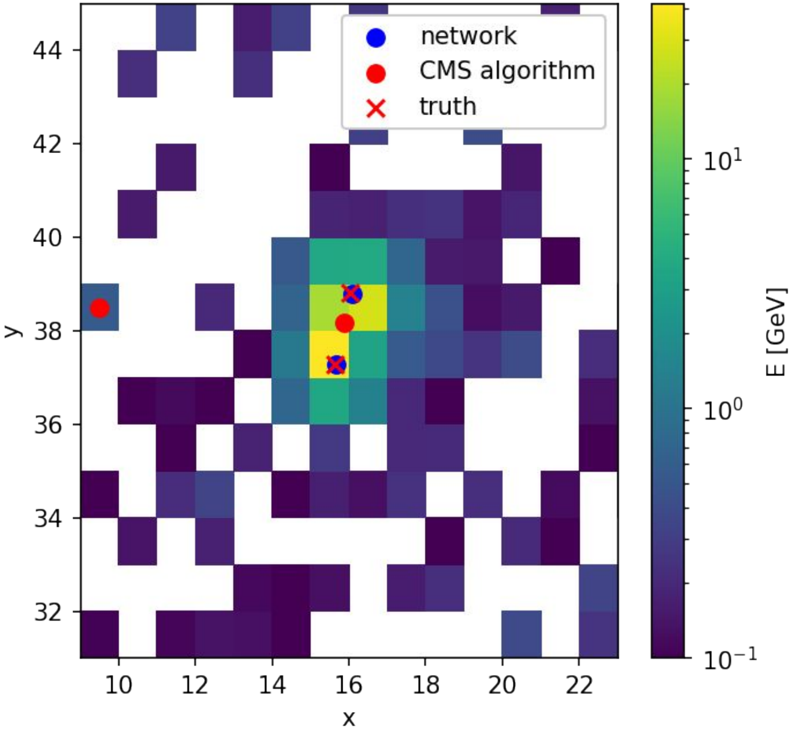

Dataset simulated with Geant4
Seed finder
Initial input
7x7 window (> 0.5 GeV)
seed finder
One by one
No connection between clusters
Take only windows that pass threshold (0.3)
Input to center finder
input - all the clusters that passed threshold from one event (window)
Center finder
Apply CNN independently on each input window
Message passing (learning about the neighbors)
Dense layers (applied separately)
(x, y), en, p
(x, y), en, p
(x, y), en, p
Vector of latent features
Final output
Double pass for dr < 0.3

Center-finder predictions are **precise coordinates, energy and corrected probability to be a real cluster** after adding MP.

Two-step net – GNN: results



With MP overestimation of energy is removed – “double-counting” solved.

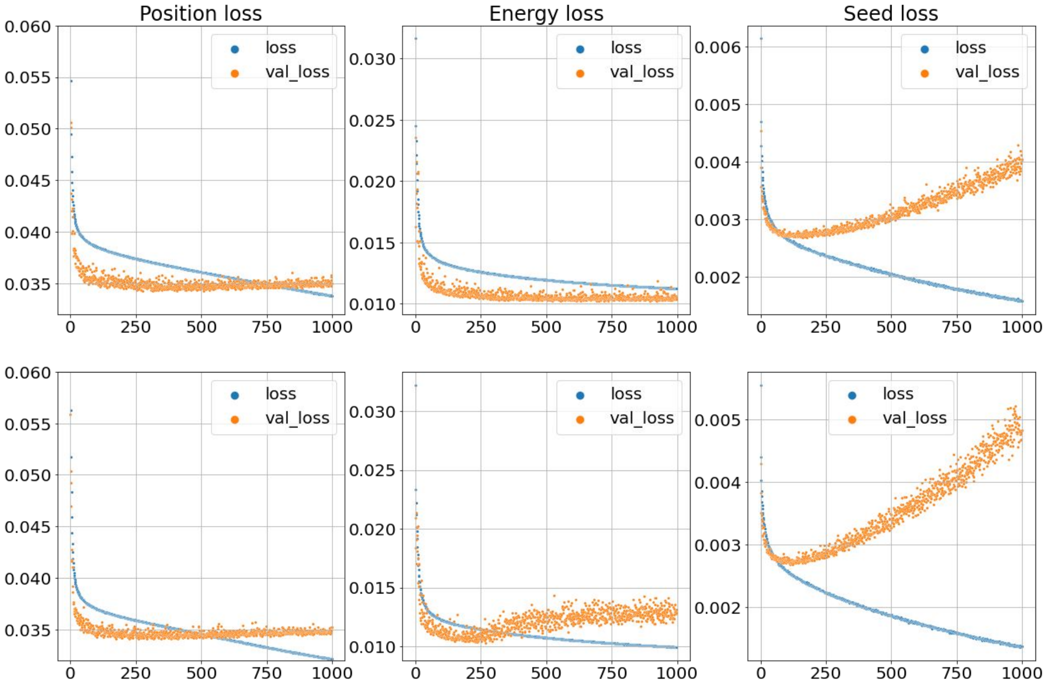


Event example where network correctly identifies two clusters while pfclustering predicts only one.

Two-step net – GNN: optimization

Choosing **weight** of the loss for the **seed probability** prediction (k).

$$\text{loss} = |x_{\text{pred}} - x_{\text{true}}| + |E_{\text{pred}} - E_{\text{true}}| + k \cdot \text{CrossEntropy}(p_{\text{pred}},\, p_{\text{true}})$$

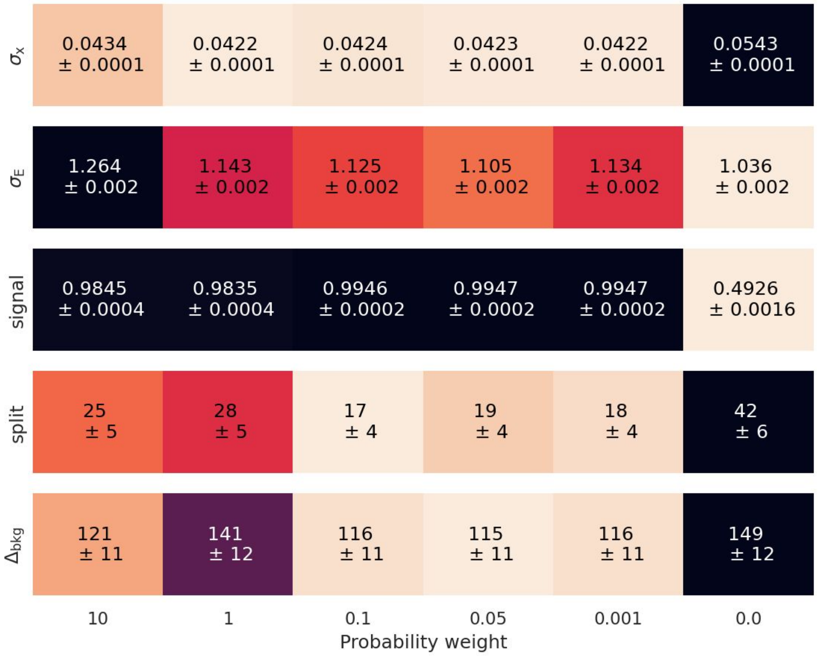


k = 0.05

k = 0.1

⇒ Need to find a “sweet spot” where all loss terms converge.

Network optimization



Probability weight	10	1	0.1	0.05	0.001	0.0
σ_x	0.0434 ± 0.0001	0.0422 ± 0.0001	0.0424 ± 0.0001	0.0423 ± 0.0001	0.0422 ± 0.0001	0.0543 ± 0.0001
σ_E	1.264 ± 0.002	1.143 ± 0.002	1.125 ± 0.002	1.105 ± 0.002	1.134 ± 0.002	1.036 ± 0.002
signal	0.9845 ± 0.0004	0.9835 ± 0.0004	0.9946 ± 0.0002	0.9947 ± 0.0002	0.9947 ± 0.0002	0.4926 ± 0.0016
split	25 ± 5	28 ± 5	17 ± 4	19 ± 4	18 ± 4	42 ± 6
Δ_{bkg}	121 ± 11	141 ± 12	116 ± 11	115 ± 11	116 ± 11	149 ± 12

All the results shown for the epoch with the smallest loss on the validation dataset.

The performance differs marginally for different weights.

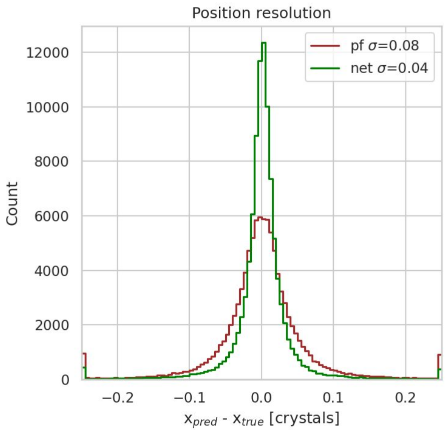
-> 0.05 chosen (epoch 300) for the best energy resolution performance.

+ Hyperparameter tuning is also performed (with the Keras Bayesian optimization).

Two-step net – GNN: results for *photon*

Performance for the final optimized network on 1 and 2 particle samples.

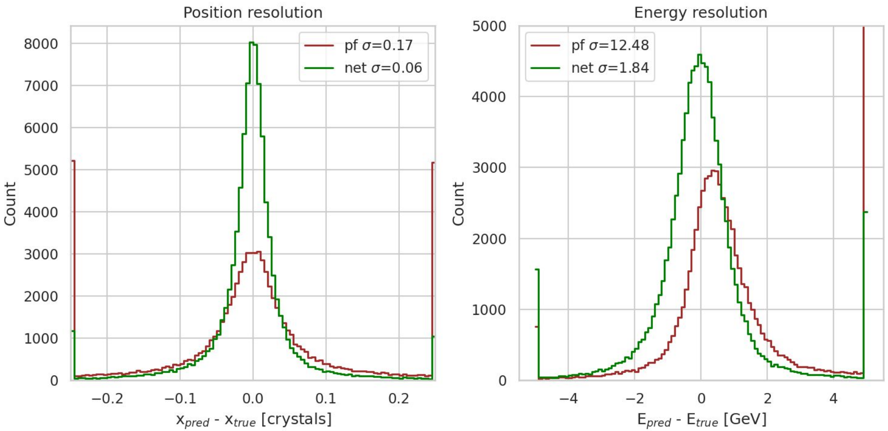
One-particle dataset



Position resolution

Energy resolution

Two-particle dataset



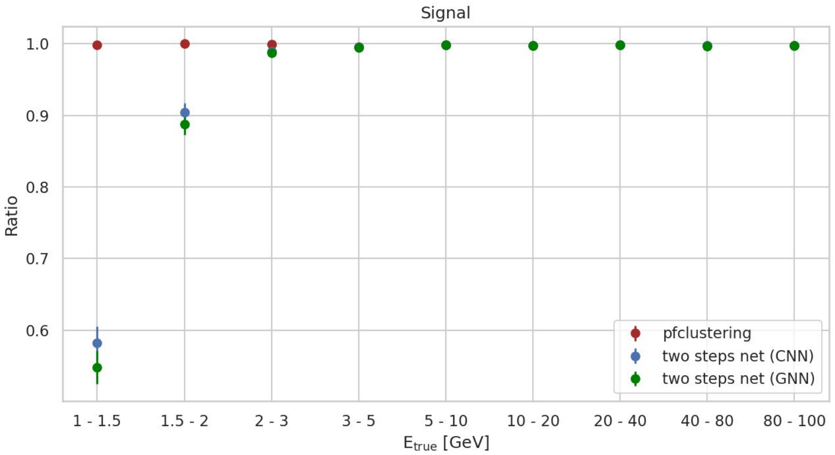
Position resolution

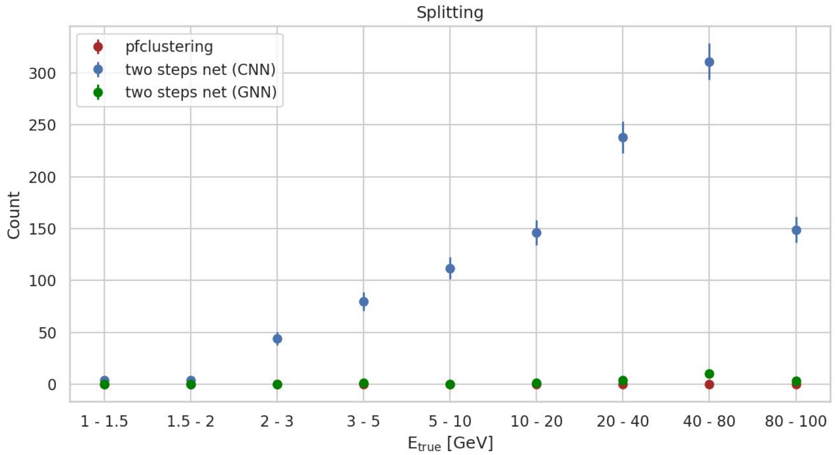
Energy resolution

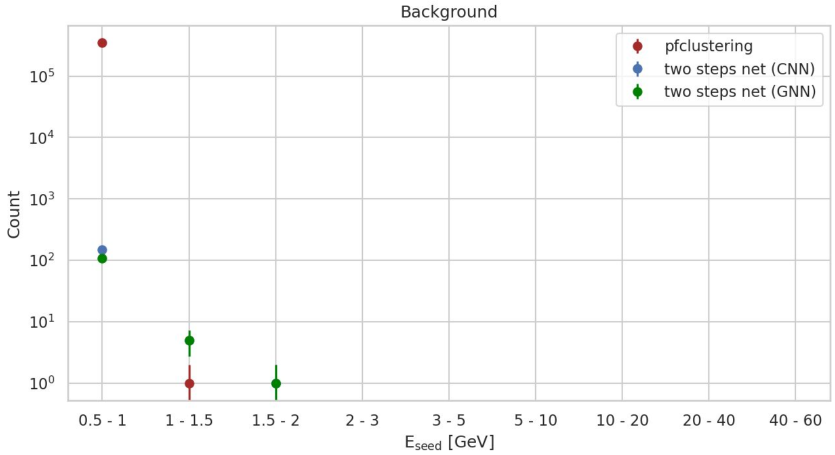
Improvement in resolution (for 1 particle):

- **Coordinate: 0.04 vs. 0.08** ECAL crystals
- **Energy: 1.10 vs. 1.22 GeV**

Two-step net – GNN: results for *one-photon dataset*

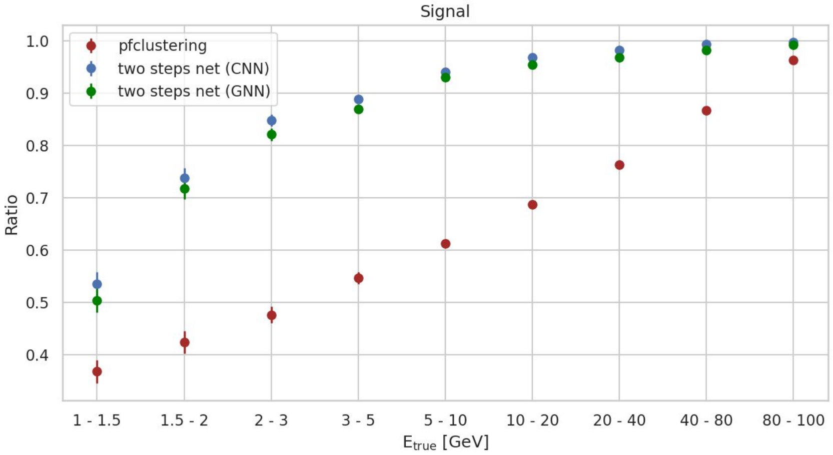


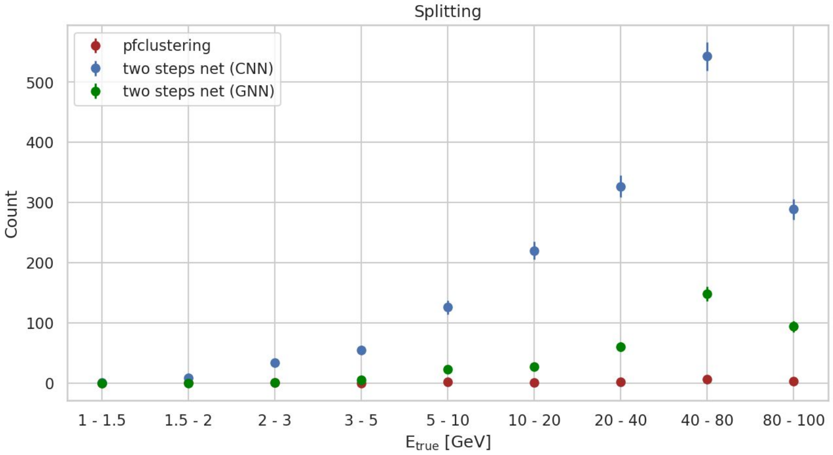


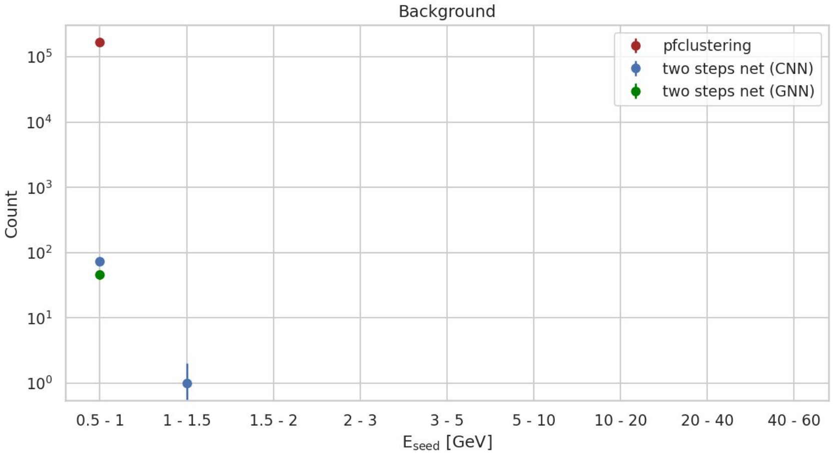


- Reaching **same efficiency** as pfclustering starting **from 3 GeV**.
- **Splitting** significantly **reduced** with GNN.
- **Much lower background** compared to pfclustering.

Two-step net – GNN: results for *two-photon dataset*



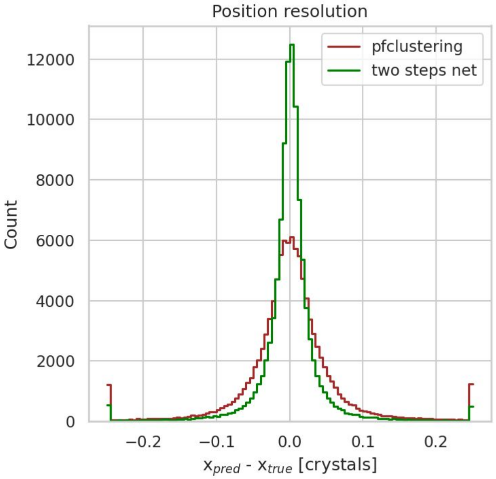


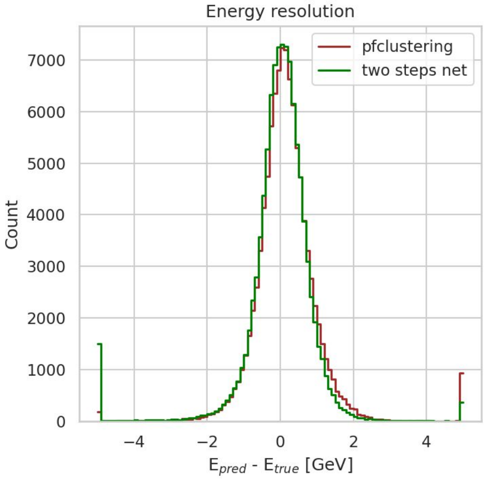


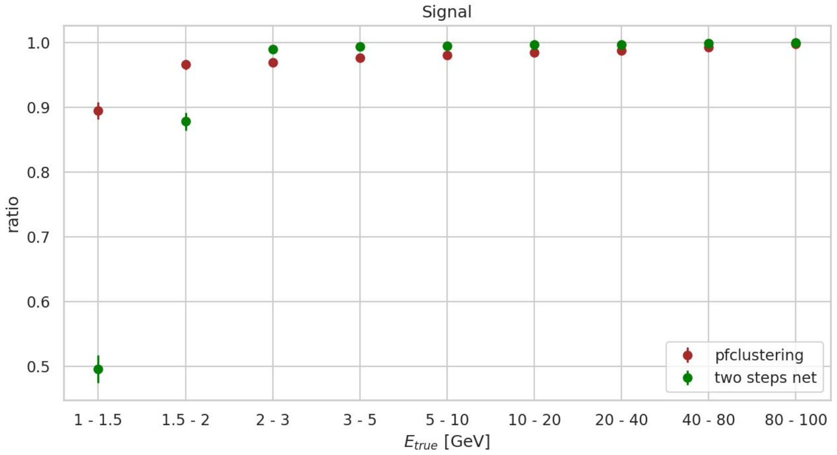
- Network **can identify** much better **two close-by particles** than pfclustering.
- **Splitting** significantly **reduced** with GNN.
- **Much lower background** compared to pfclustering.

Two-step net – GNN: results for *electron*

Electron sample with E = [1, 100] GeV (flat distribution) each event has up to six particles. No separate training done - only evaluation.





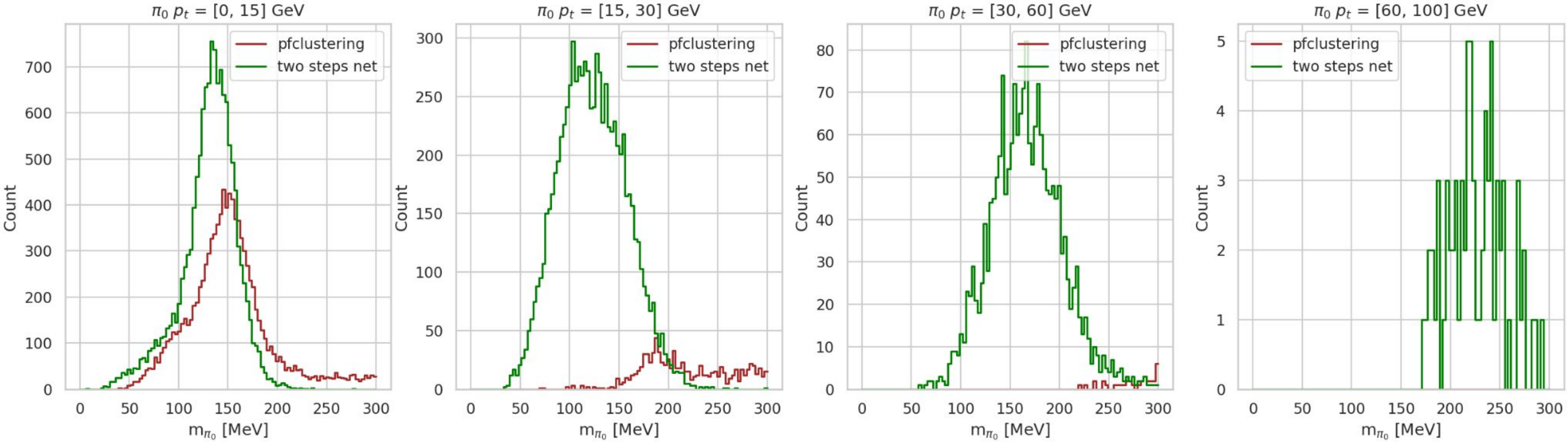


- Network shows high performance even though the samples were not part of the training (electrons and > 2 particles).
- Improved **position resolution** and **efficiency**!
- Similar energy resolution between pfclustering and the network.

Two-step net – GNN: results for *pion*

Pion sample with E = [1, 100] GeV (flat distribution). Only evaluation.

Two most energetic photons are chosen in the prediction to evaluate pion mass.



> 2 times more pions reconstructed with network

Better mass resolution is achieved

Summary and outlook

- A new ML-based ECAL-clustering reconstruction is presented with a simplified calorimeter simulation.
- Three different networks are developed, the GNN-based algorithm outperforms pfclustering in most aspects (tested on both photons and electrons).
- Significantly better reconstruction of two close-by photons (which is one of the main goals of the study).
  - Enables pion reconstruction.
  - Should significantly improve photon/pion discrimination.

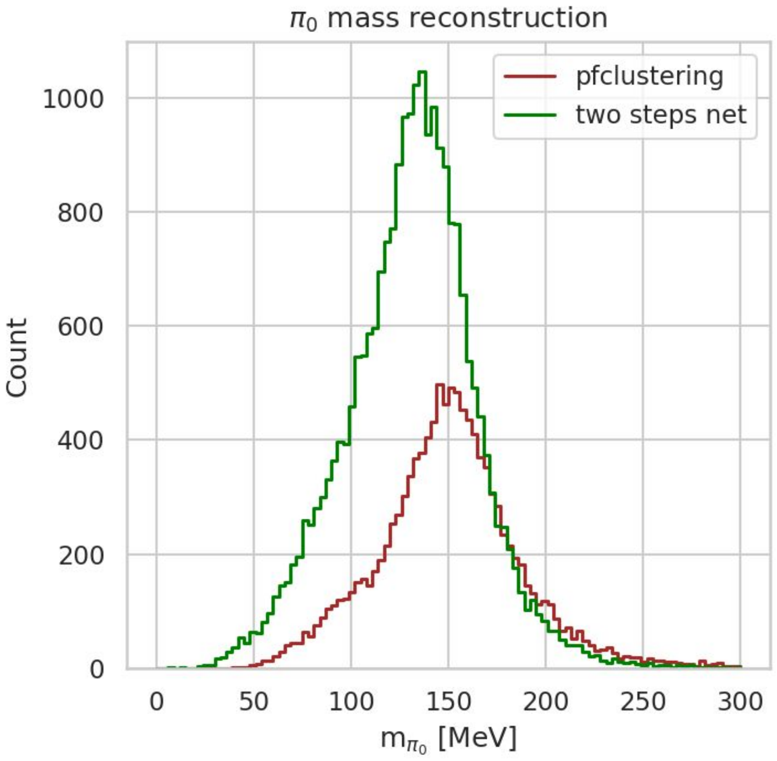
Outlook:

- Publication of a paper on the results is ongoing.
- Training on actual ECAL simulation is necessary before implementation in CMSSW.
- Implementation in CMSSW and performance estimation.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No *800945* — NUMERICS — H2020-MSCA-COFUND-2017

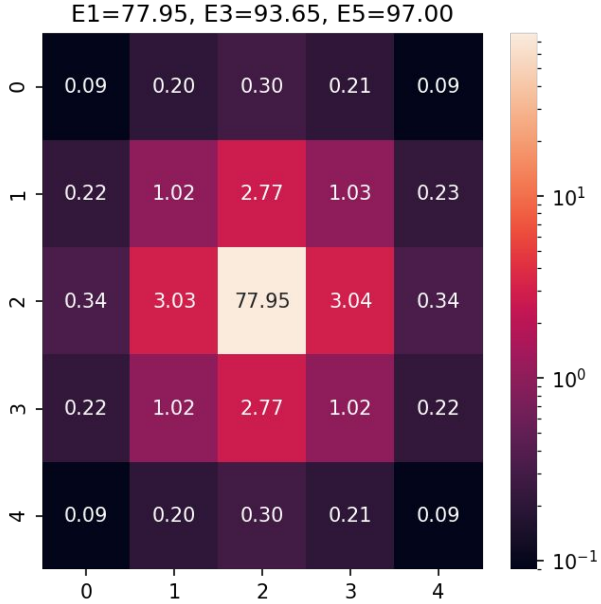
Backup

Pions linked to truth

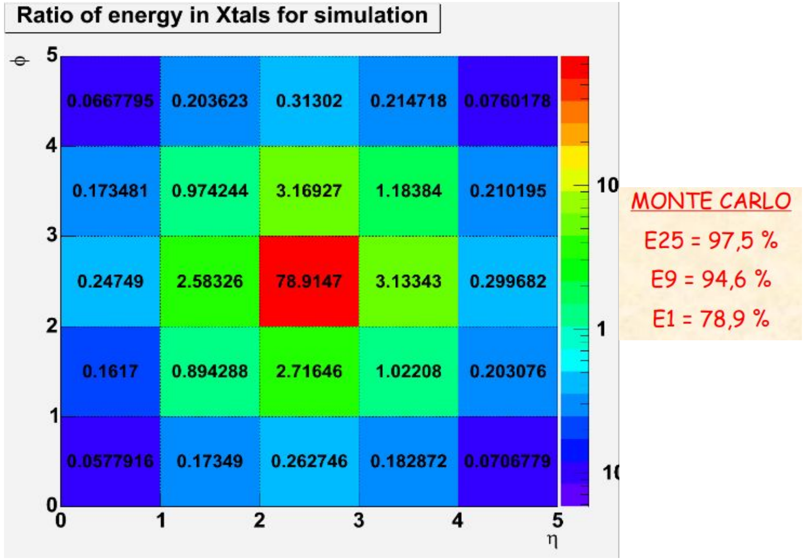


Energy deposit profile

- To **validate** the calorimeter simulation the **energy deposit profile** was plotted.
- Using the data from **1 000 electrons, all at 100 GeV** shooting at the **crystal center of the middle crystal.**



Energy deposits from the simplified detector.



MONTE CARLO

E25 = 97,5 %

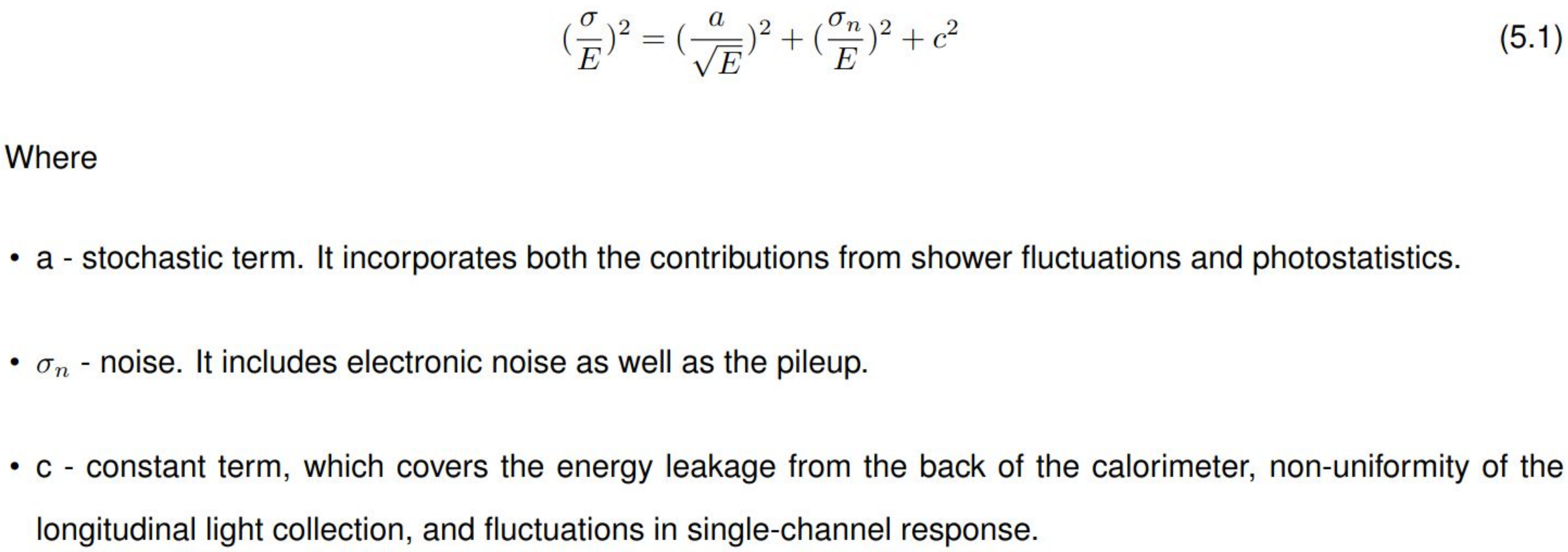
E9 = 94,6 %

E1 = 78,9 %

Energy deposits from Geant4 simulation of ECAL.
http://geant4.in2p3.fr/2005/Workshop/UserSession/P.Mine.pdf

The results are very similar => the simulation can be used as a **proxy for CMS ECAL**.

ECAL resolution

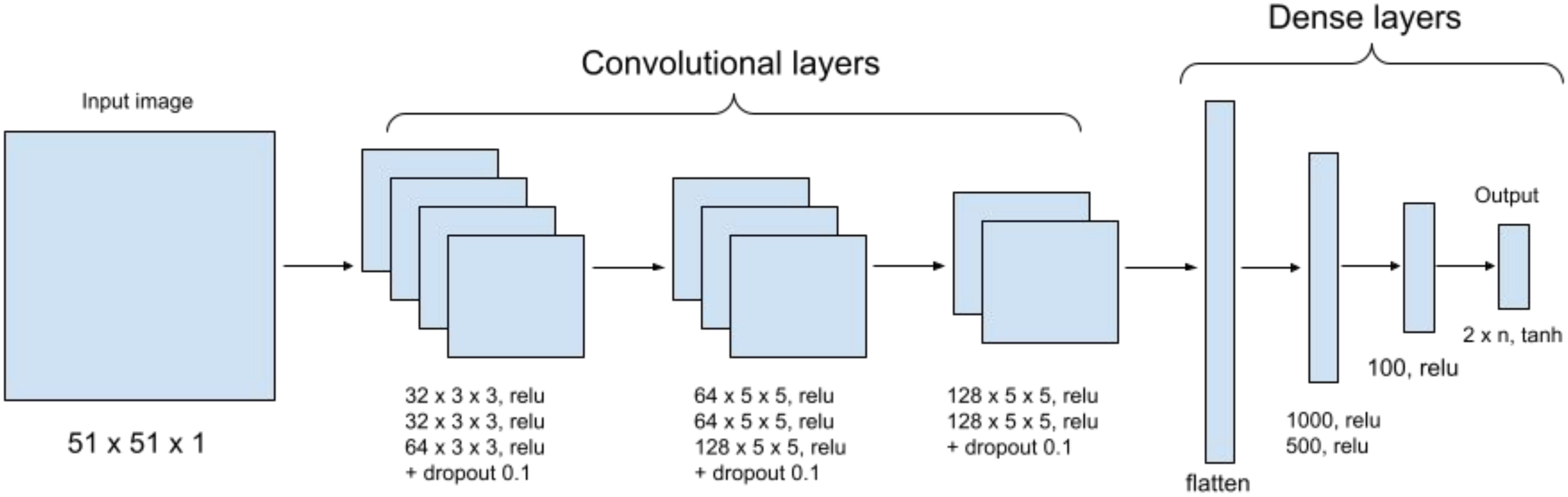


$$(\frac{\sigma}{E})^2 = (\frac{a}{\sqrt{E}})^2 + (\frac{\sigma_n}{E})^2 + c^2 \quad (5.1)$$

Where

- a - stochastic term. It incorporates both the contributions from shower fluctuations and photostatistics.
- σ_n - noise. It includes electronic noise as well as the pileup.
- c - constant term, which covers the energy leakage from the back of the calorimeter, non-uniformity of the longitudinal light collection, and fluctuations in single-channel response.

One-shot network architecture

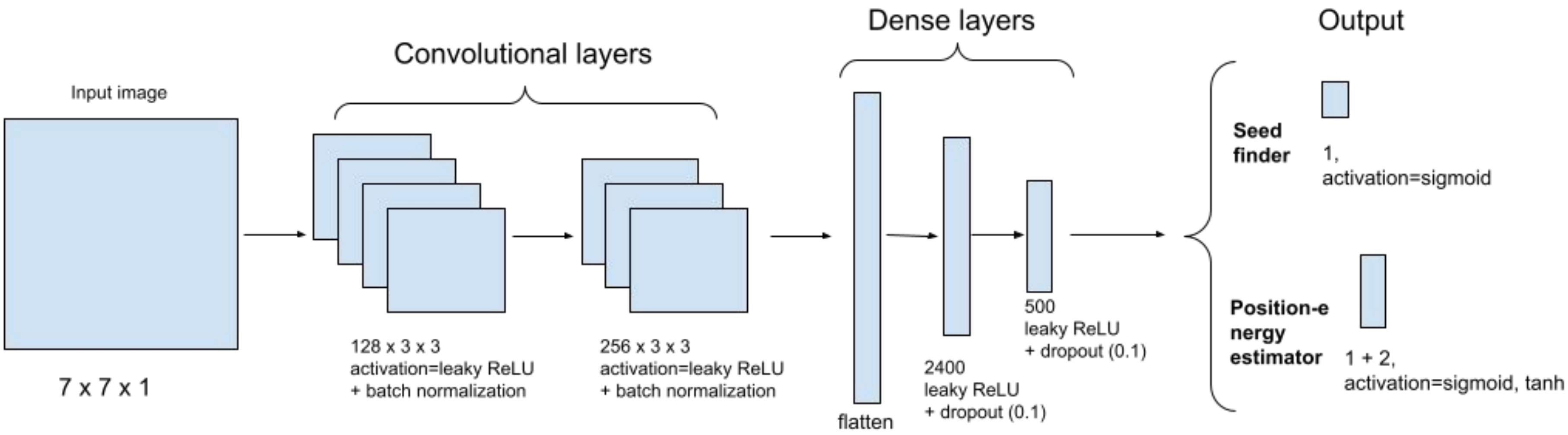


Hyperparameters:

- learning rate = 0.001
- batch size = 64
- epochs ~ 500

Loss function: Mean Absolute Error

Double-step network architecture



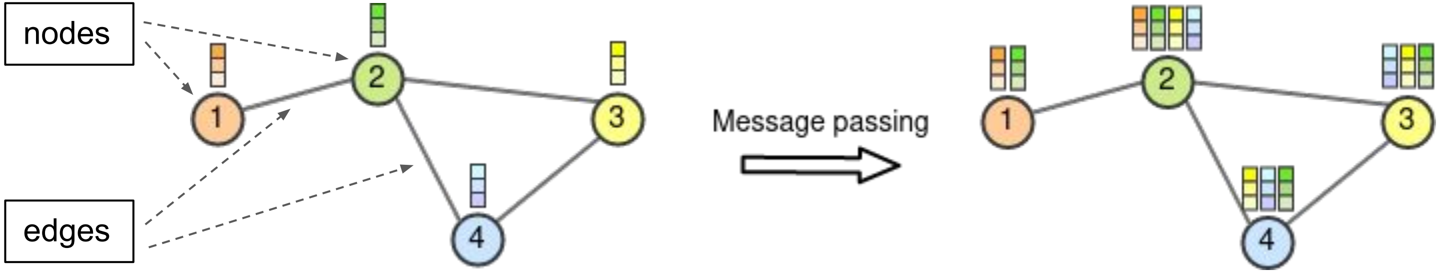
Hyperparameters:

- learning rate = 0.0001
- batch size = 64
- epochs ~ 500

Loss function: Binary Crossentropy (for seed finder) or Mean Absolute Error (for position-energy estimator).

Graph Neural Networks

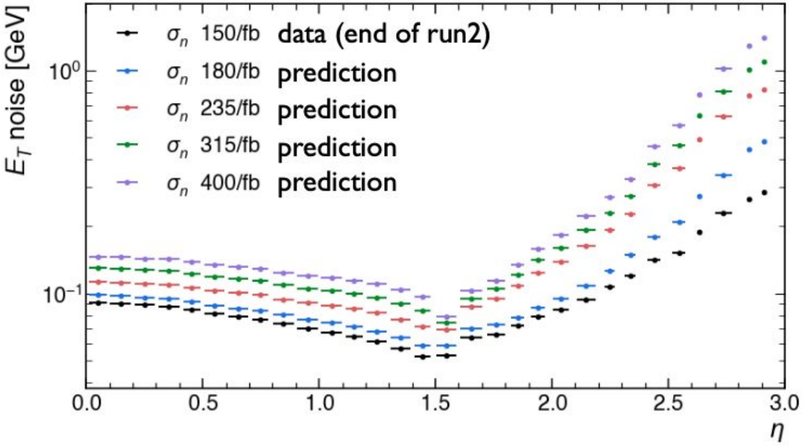
- Type of neural network that can operate on and analyze **graph structures**.
- Unlike other types of networks GNN can be easily applied on sparse data, doesn’t require padding.
- A graph consists of **nodes** (contain features of the object) and **edges** (reflect the relationship between the nodes).
- In GNNs the information can be shared between the neighbors:
  - The vector features of each node are transformed into “messages” (e.g. using dense layers) that are sent to the neighbors (message-passing).
  - In this way, **each node learns information about its neighbors and itself**. The process is carried out in parallel and repeated several times.



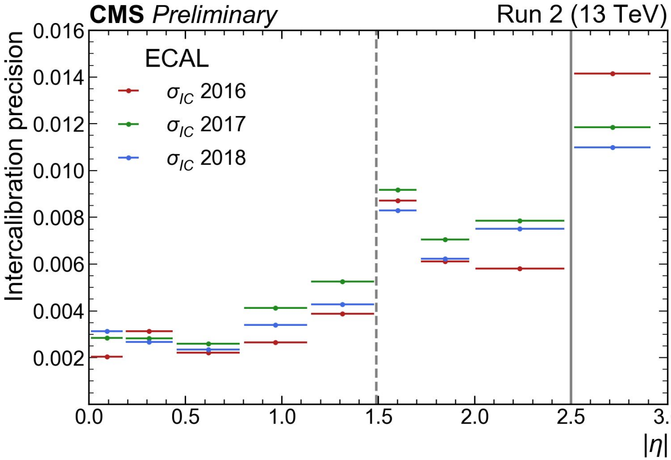
https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial7/GNN_overview.html

ECAL resolution parameters

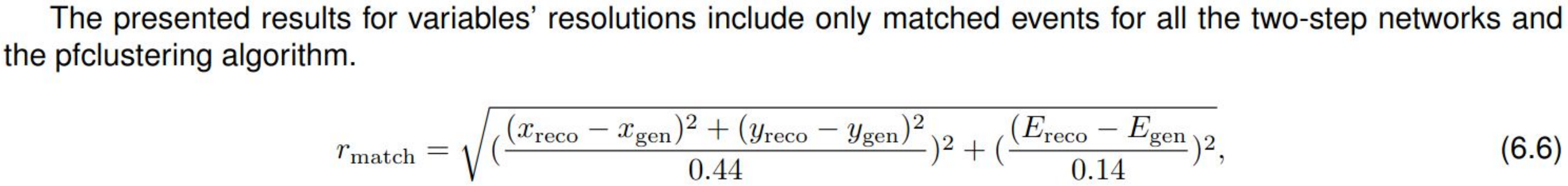
Noise term:



Constant term:



Matching conditions



The presented results for variables' resolutions include only matched events for all the two-step networks and the pfclustering algorithm.

$$r_{\text{match}} = \sqrt{(\frac{(x_{\text{reco}} - x_{\text{gen}})^2 + (y_{\text{reco}} - y_{\text{gen}})^2}{0.44})^2 + (\frac{(E_{\text{reco}} - E_{\text{gen}}}{0.14})^2}, \tag{6.6}$$