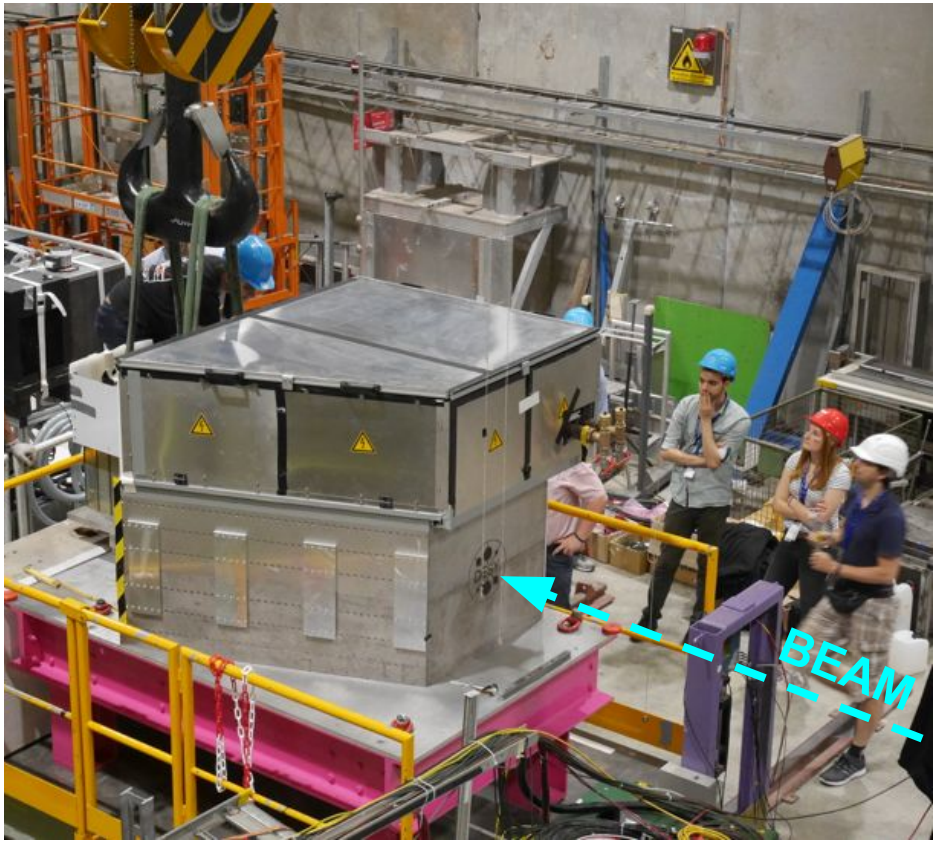
Deep Learning Studies with the CALICE AHCAL Technological Prototype

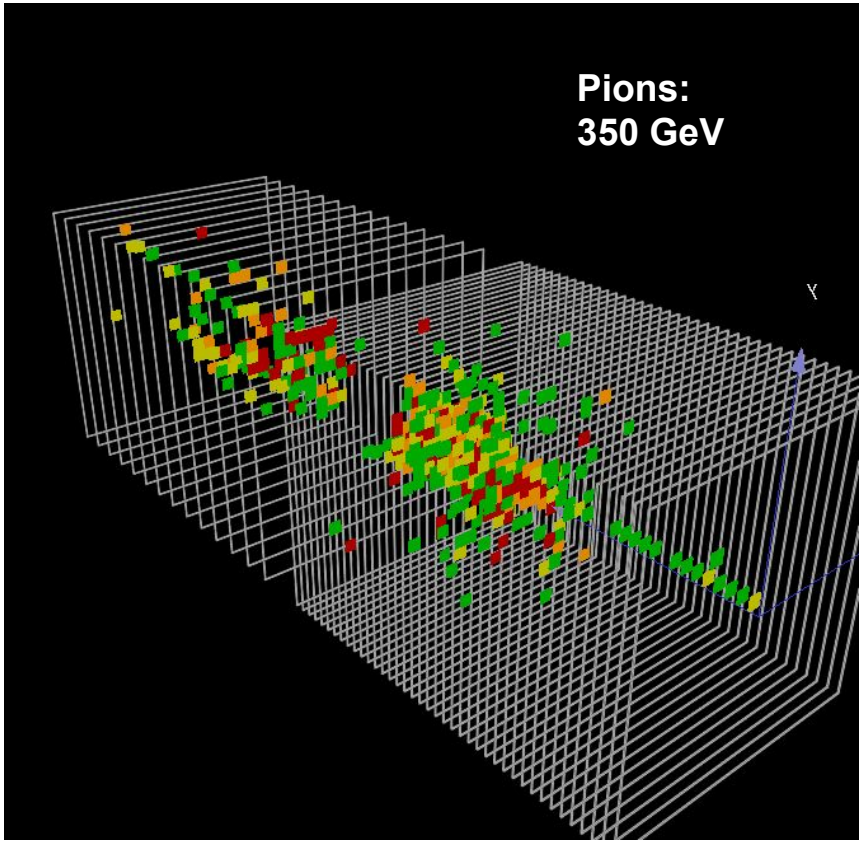
Erik Buhmann[1], Erika Garutti and Gregor Kasieczka

CALICE meeting, Utrecht
April 12, 2019

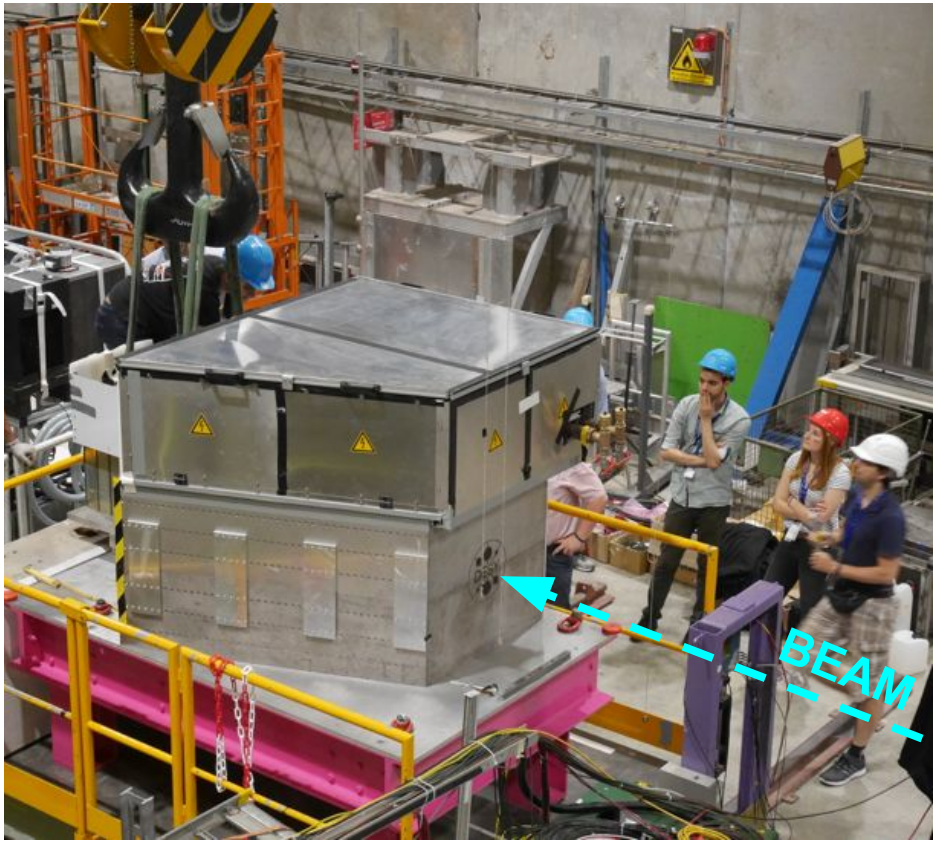
[1] eMail: erik.buhmann@desy.de

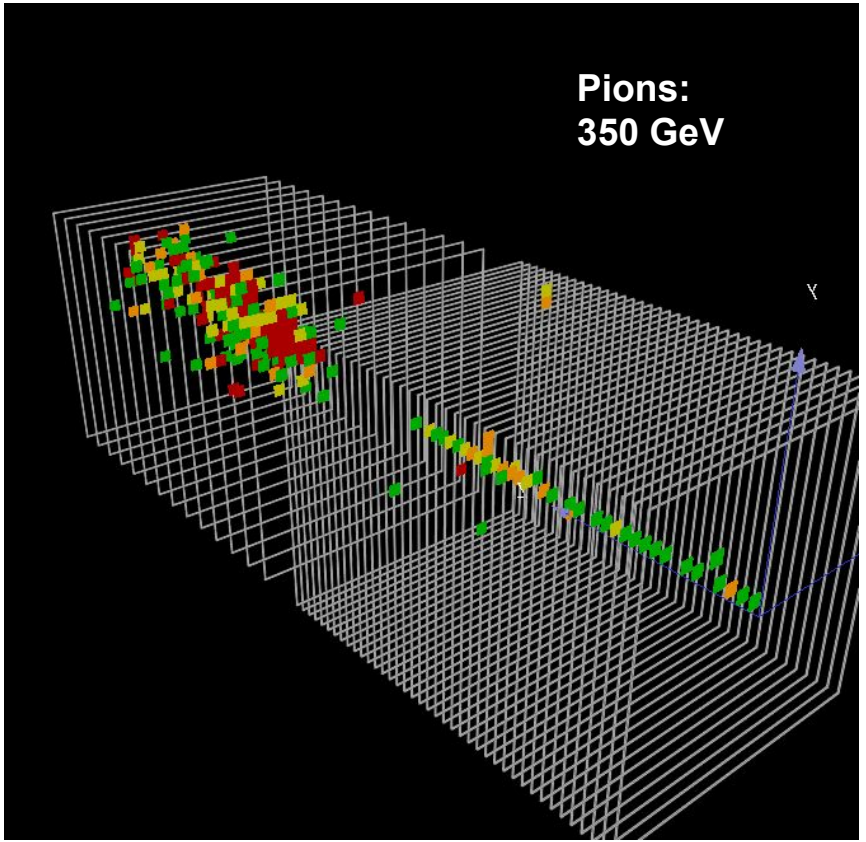
Test Beam Setup: Technological Prototype @ SPS



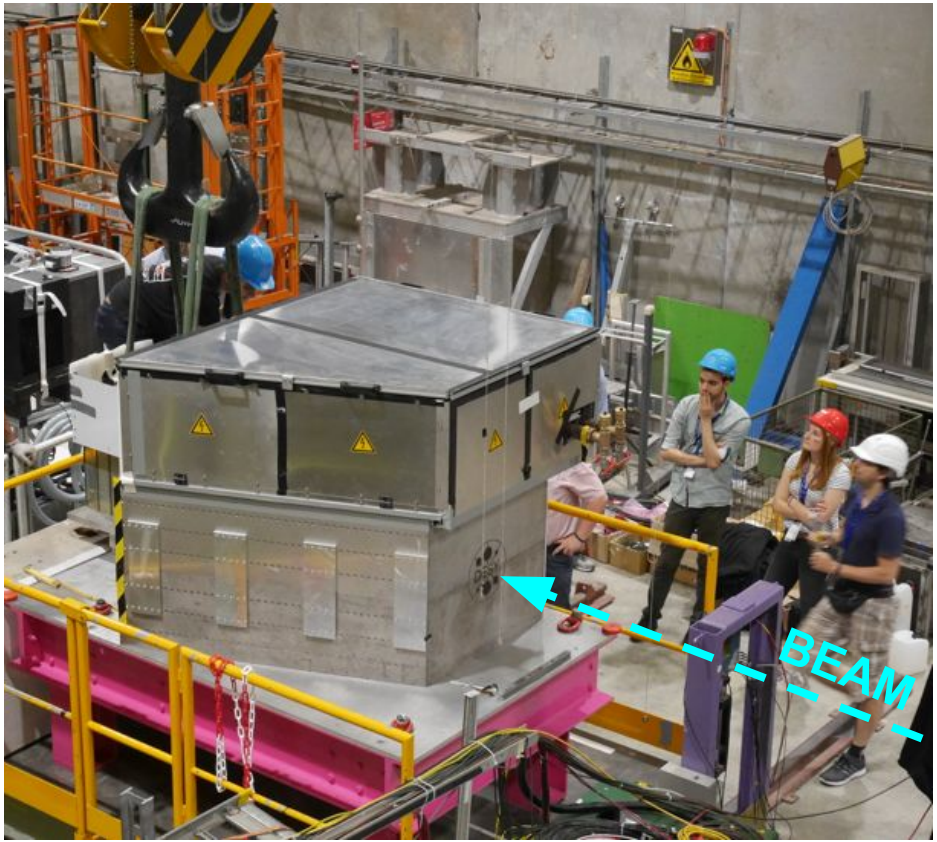


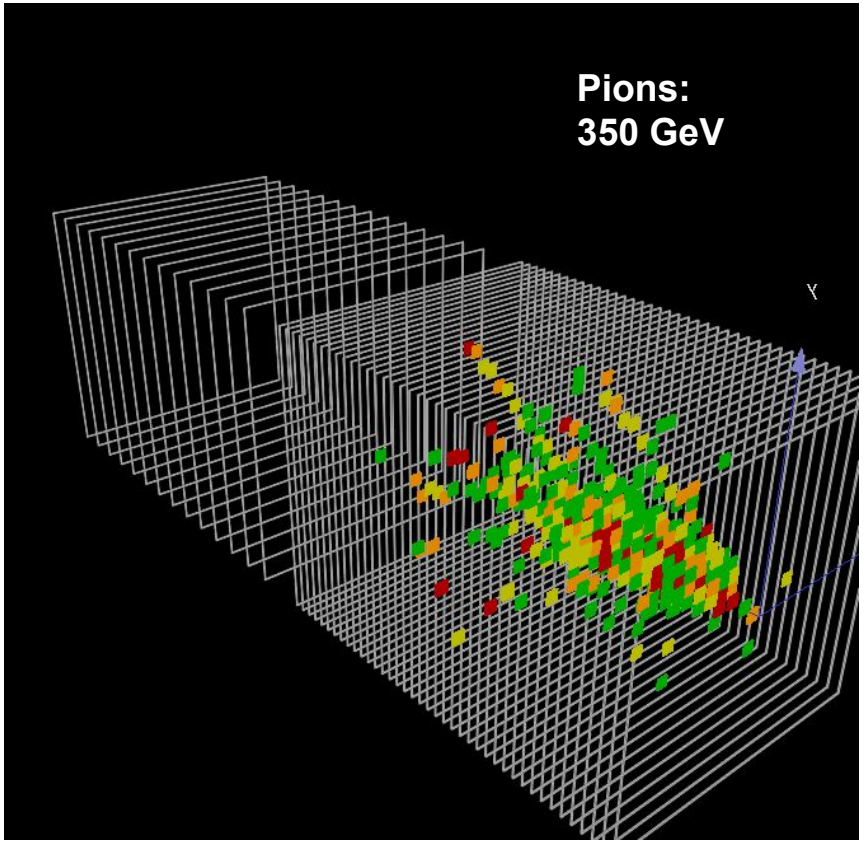
Test Beam Setup: Technological Prototype @ SPS





Test Beam Setup: Technological Prototype @ SPS





Deep Learned Calorimetry

Can we analyze the "shower images" using Deep Learning?

Test beam environment:

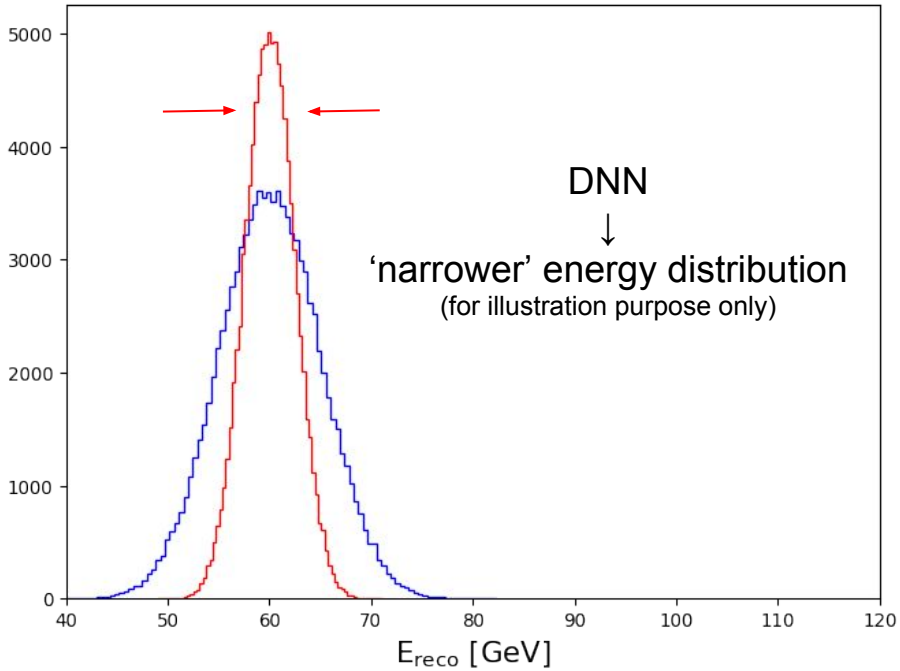
→ labeled experimental data

→ training on data possible
(instead of Monte Carlo)

Machine learning task:

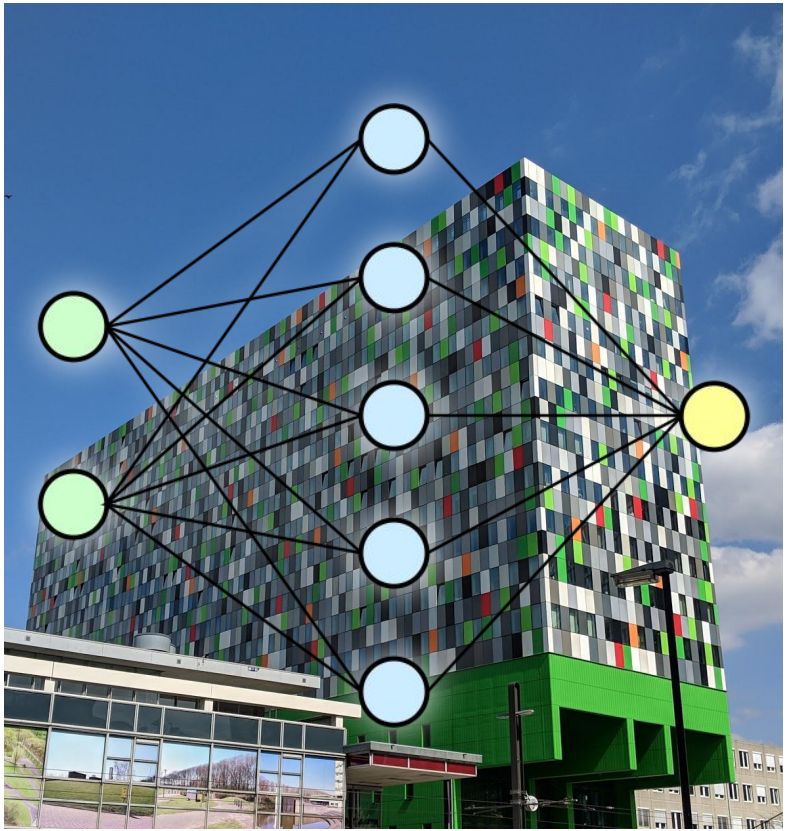
→ reconstruct particle energy

→ using low level information
of the highly granular calorimeter



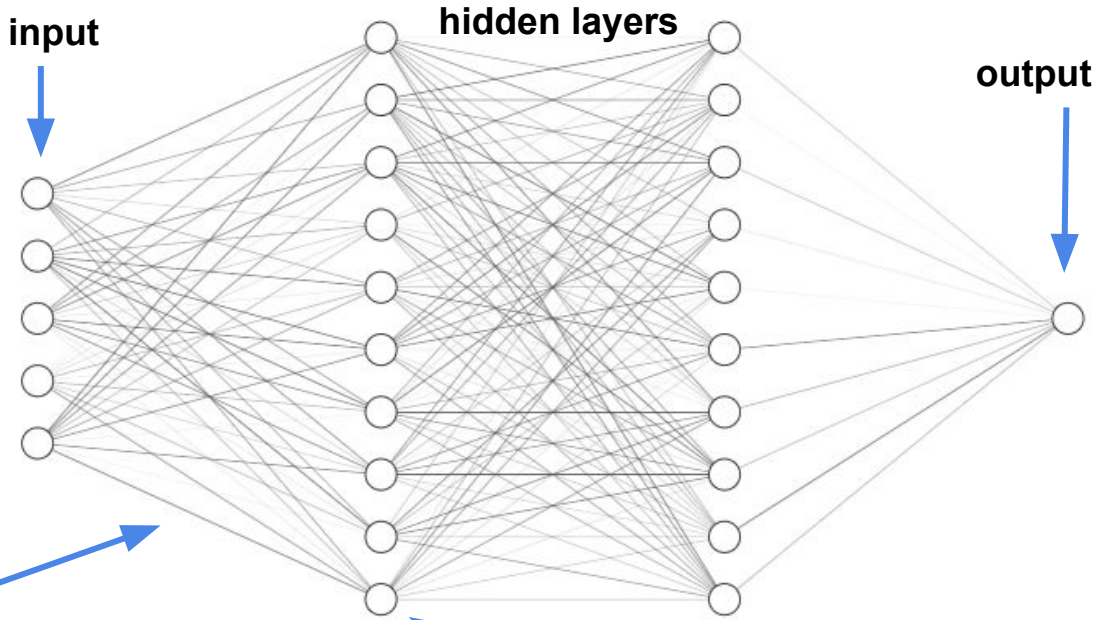
Deep Learning

- Image analysis with machine learning, i.e.
  - Classification → What particle?
  - Regression → What energy?
- **D**eep **N**eural **N**etwork (DNN) based on connection of different layers of “neurons”
- Deep Learning with data set of low-level variables
- Different types of layers in use → Network “architecture”
- Training of connections as iterative process on labeled data set



Fully Connected Layer

- “Classical” Artificial Neural Network (ANN)
- Row of neurons = 1 “layer”
- During training connections are strengthened or weakened

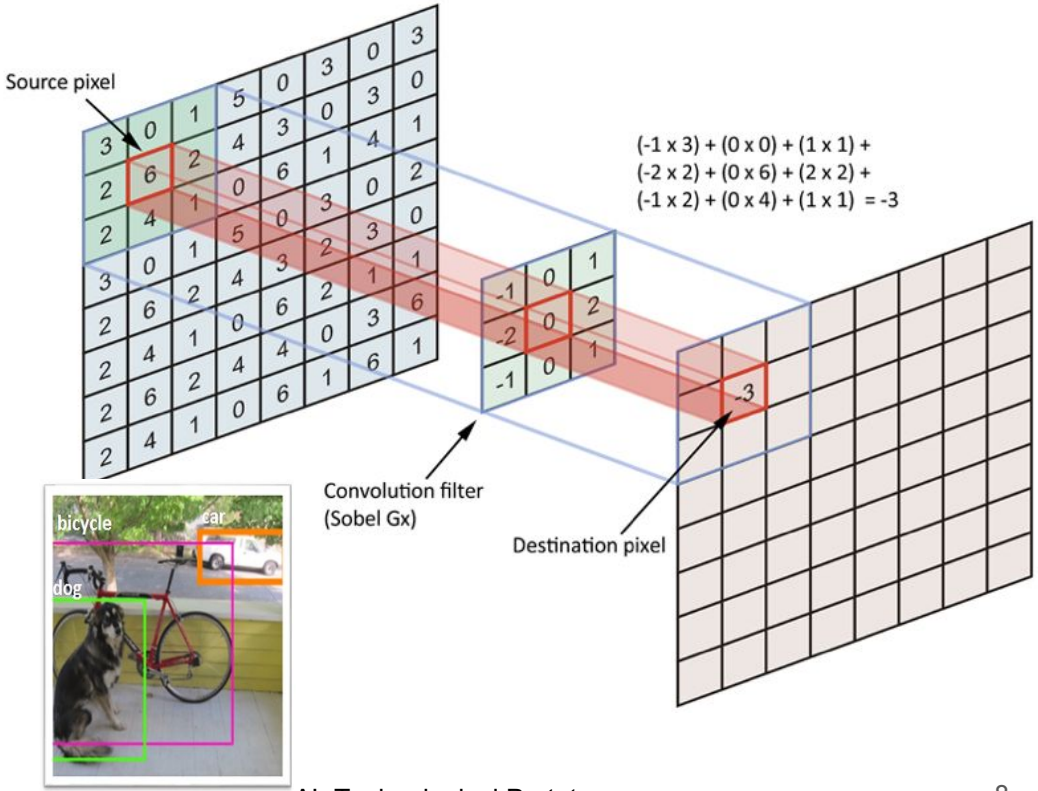


Convolutional Layer

- Output value of convolutional kernel:

$$O_{conv} = (w_1 E_1 + w_2 E_2 + w_3 E_3 + ...) + b$$

- **Shared** weights
  - Multiple filters used to learn different features of the image
  - Same filter (here 3 x 3) is used at every position
    → 9 trainable weights

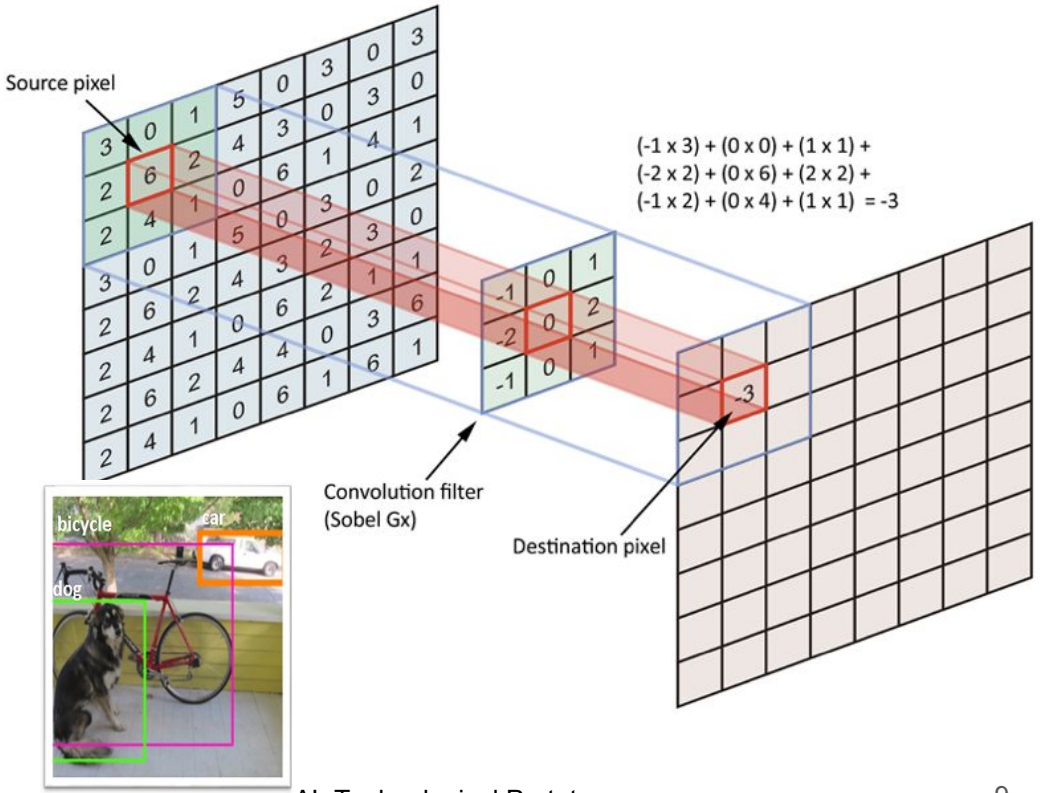


Locally Connected Layer

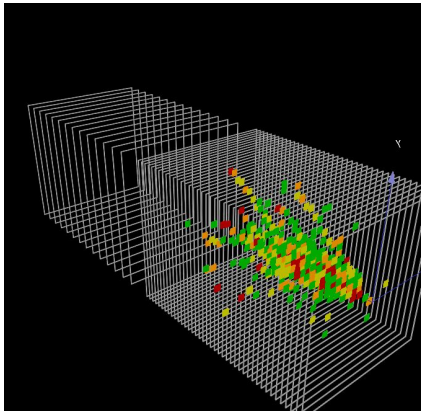
- Output value of convolutional kernel:

$$O_{conv} = (w_1 E_1 + w_2 E_2 + w_3 E_3 + ...) + b$$

- **Unshared** weights
  - Multiple filters used to learn different features of the image
  - New filter (here 3 x 3) is used at every position
    → Localized filters
    → 36 x 9 = 324 trainable weights



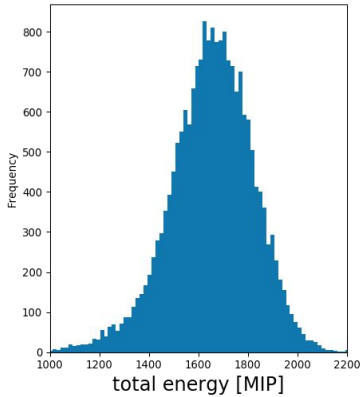
Input & Preprocessing



- Pion data from May 2018 testbeam @ SPS
  - Reconstruction performed with CALICE software
- "Shower images" as (24,24,38) arrays
  - With hit energies at (I,J,K) coordinates
- Ground truth values: known beam energies
- N_{hits} cuts for "Gaussian" distribution in event energy
  - nHits in first & last 2 layers < 10 & energy dependent total nHits cut
- Applied rough MIP to TeV conversion factor = 2.9E-5
- Sample of 50,000 events per energy
  - 60 % for training, 20 % for validation & testing
  - Beam energies: **10**, **15**, **20**, **30**, **40**, **50**, **60**, **80**, **100**, **120** & **160** GeV
  - **Training sample with only every second beam energy (!)**
- Two test samples ("*trained on*" & "*NOT trained on*")
  - **Can network reconstruct energies it was NOT trained on?**

Input & Preprocessing

- Pion data from May 2018 testbeam @ SPS
  - Reconstruction performed with CALICE software
- “Shower images” as (24,24,38) arrays
  - With hit energies at (I,J,K) coordinates
- Ground truth values: known beam energies
- N_{hits} cuts for “Gaussian” distribution in event energy
  - nHits in first & last 2 layers < 10 & energy dependent total nHits cut
- Applied rough MIP to TeV conversion factor = 2.9E-5
- Sample of 50,000 events per energy
  - 60 % for training, 20 % for validation & testing
  - Beam energies: **10**, **15**, **20**, **30**, **40**, **50**, **60**, **80**, **100**, **120** & **160** GeV
  - **Training sample with only every second beam energy (!)**
- Two test samples (“*trained on*” & “*NOT trained on*”)
  - **Can network reconstruct energies it was NOT trained on?**

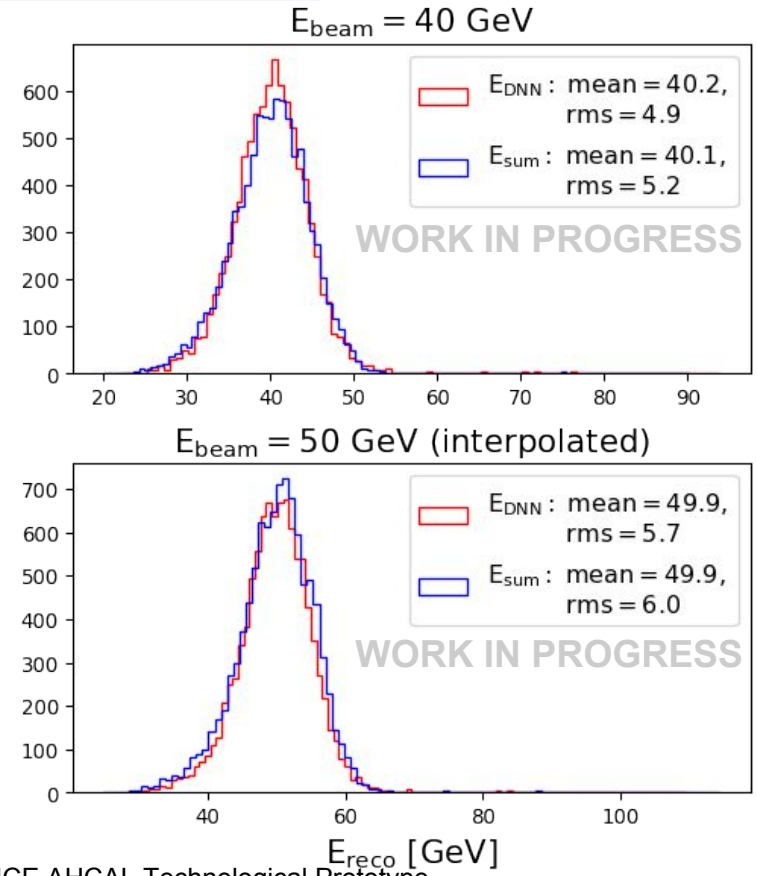


3D Conv. Filter Size = (24,24,38)

- One trainable Convolutional Layer with size of detector
- Output: Weighted sum of all channels
- Better performance than energy sum
- No systematic difference between “trained on” and “NOT trained on” energies

$$E_{DNN} = \sum_i E_i w_i \cdot f_{MIP2GeV}$$

~ 22,000 weights

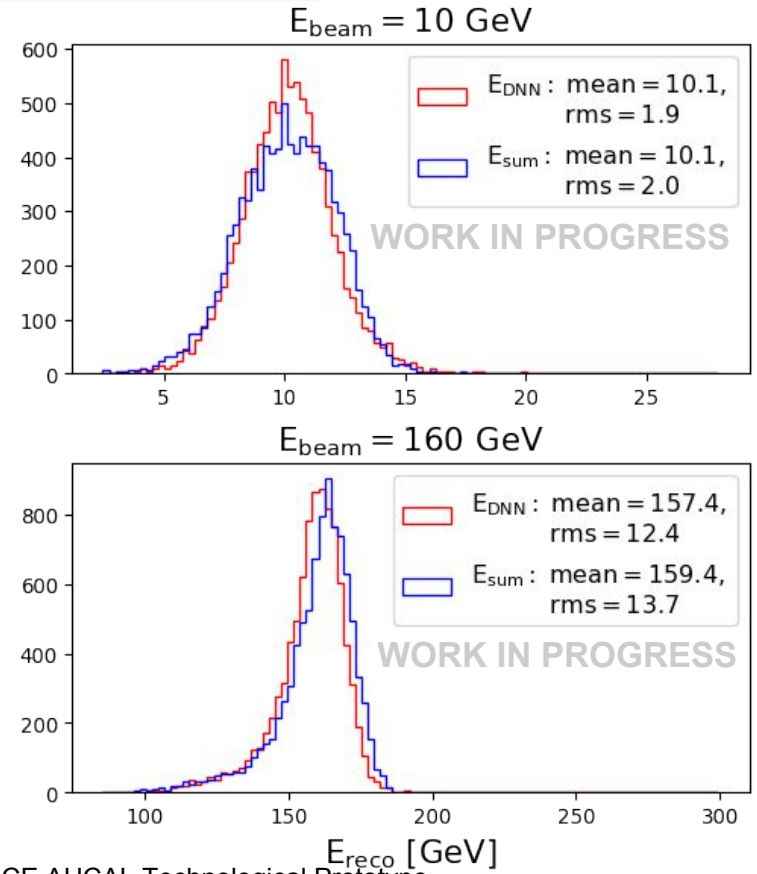


3D Conv. Filter Size = (24,24,38)

- One trainable Convolutional Layer with size of detector
- Output: Weighted sum of all channels
- Better performance than energy sum
- No systematic difference between “trained on” and “NOT trained on” energies

$$E_{DNN} = \sum_i E_i w_i \cdot f_{MIP2GeV}$$

~ 22,000 weights

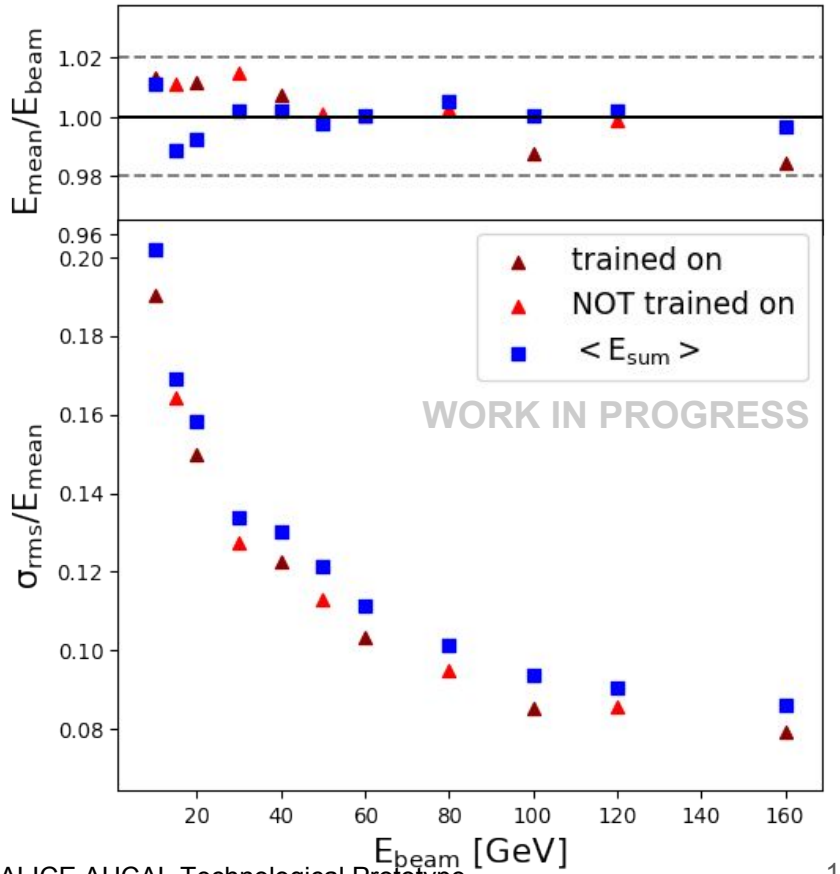


3D Conv. Filter Size = (24,24,38)

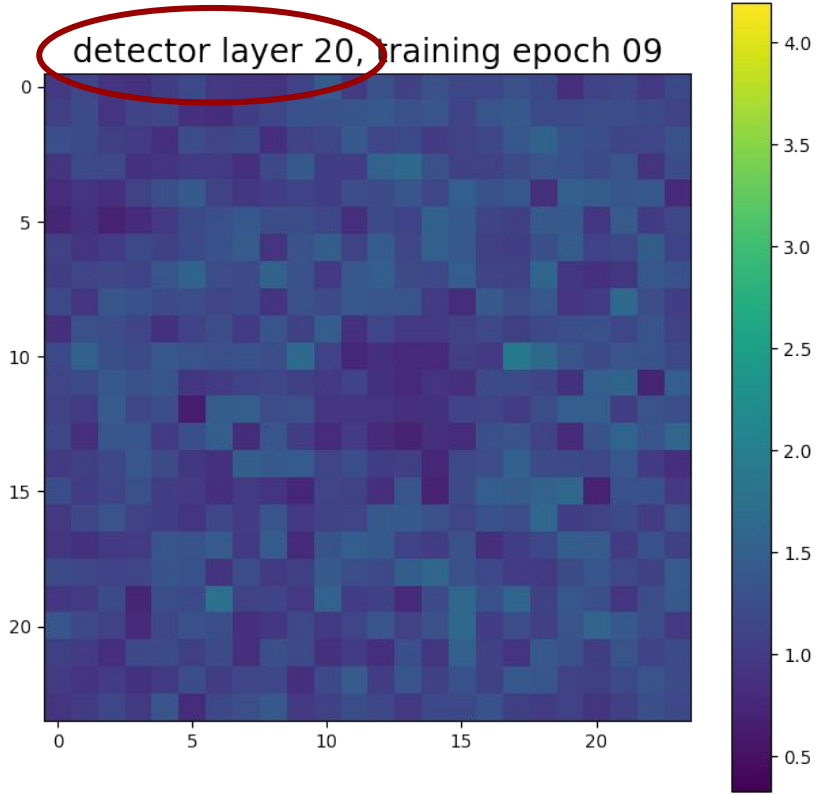
- One trainable Convolutional Layer with size of detector
- Output: Weighted sum of all channels
- Better performance than energy sum
- No systematic difference between “trained on” and “NOT trained on” energies

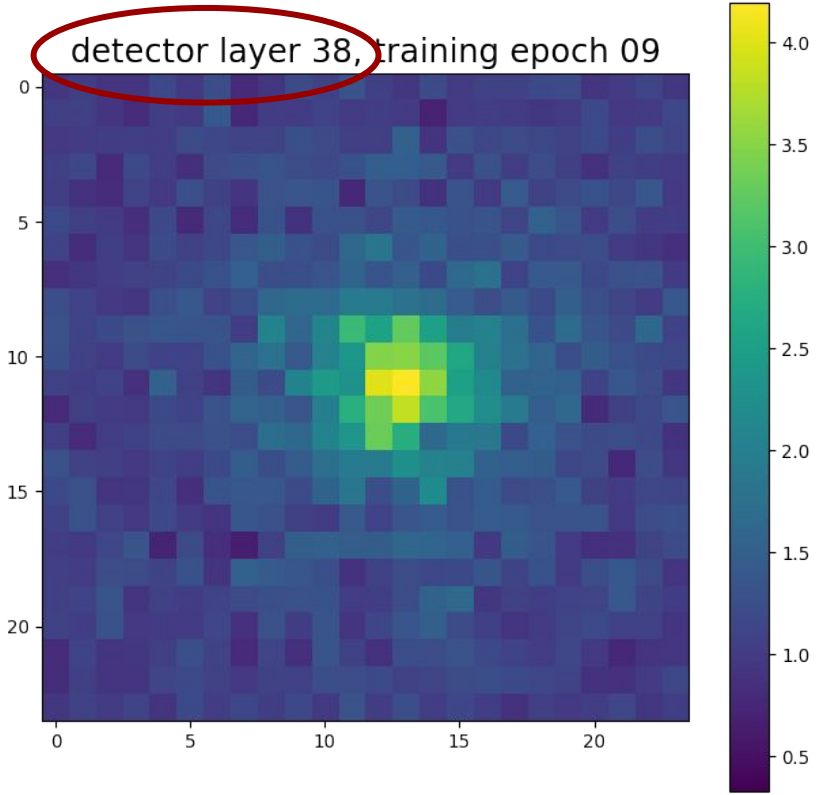
$$E_{DNN} = \sum_i E_i w_i \cdot f_{MIP2GeV}$$

~ 22,000 weights



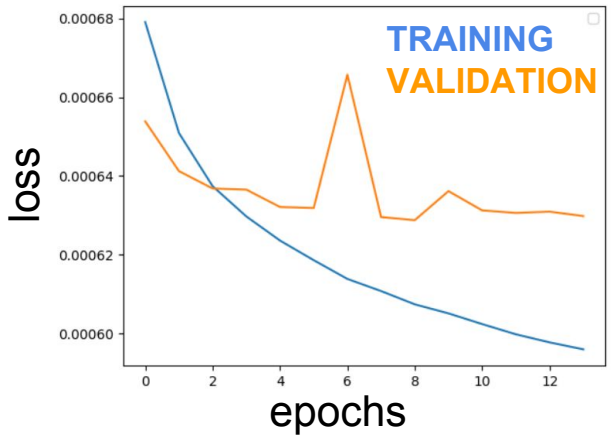
Learned Weights





Comparison of Loss Functions

→ The loss function is **minimized during training** and used to adjust the weights after each training interval.



Options compared:

1) mean squared error (MSE):

$$L(E_{i,true}, E_{i,pred}) = \frac{1}{N} \sum_i \left(E_{i,pred} - E_{i,true}\right)^2$$

2) mean squared relative error (MSRE):

$$L(E_{i,true}, E_{i,pred}) = \frac{1}{N} \sum_i \left(\frac{E_{i,pred} - E_{i,true}}{E_{i,true}}\right)^2$$

3) mean relative squared error (MRSE):

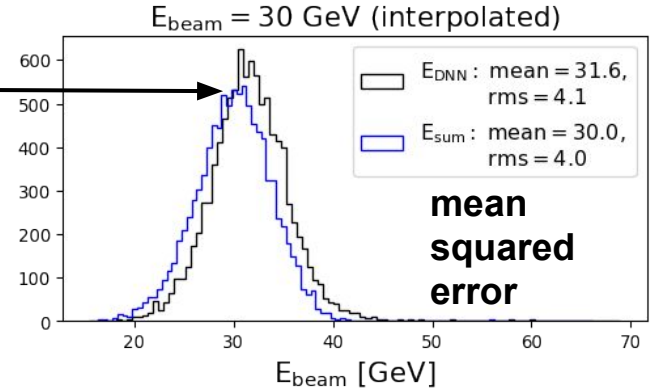
$$L(E_{i,true}, E_{i,pred}) = \frac{1}{N} \sum_i \left(\frac{E_{i,true} - E_{i,pred}}{\sqrt{E_{i,true}}}\right)^2$$

Comparison of Loss Functions

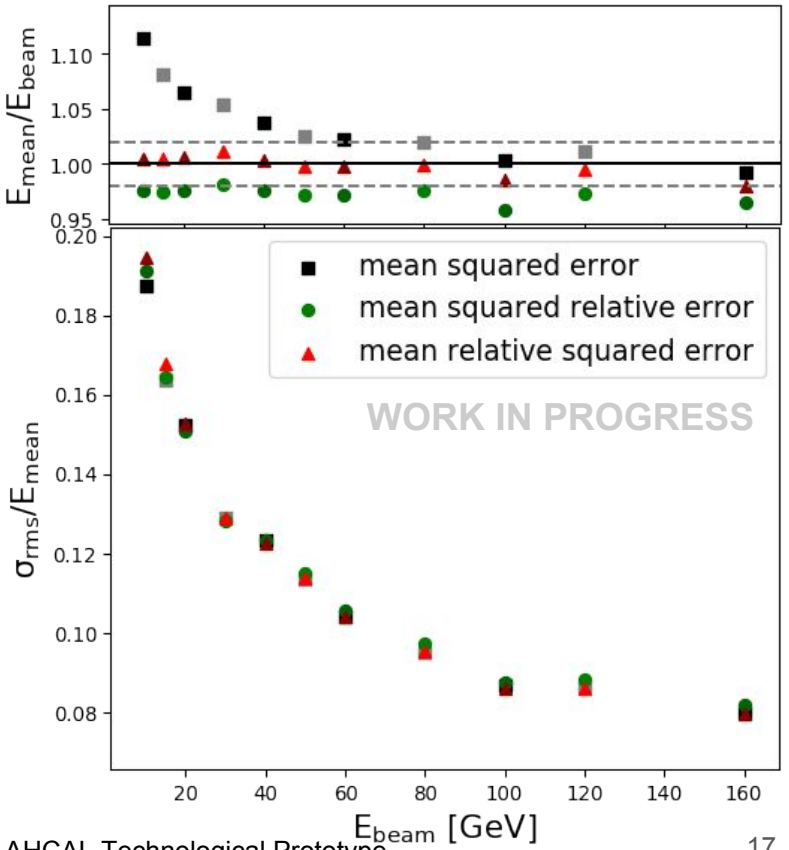
Mean energy is best reconstructed with **mean relative squared error** as loss:

$$L(E_{i,true}, E_{i,pred}) = \frac{1}{N} \sum_i \left(\frac{E_{i,true} - E_{i,pred}}{\sqrt{E_{i,true}}} \right)^2$$

mean energy shift



➔ Explainable with the 1/√E dependence of the calorimeter energy resolution.



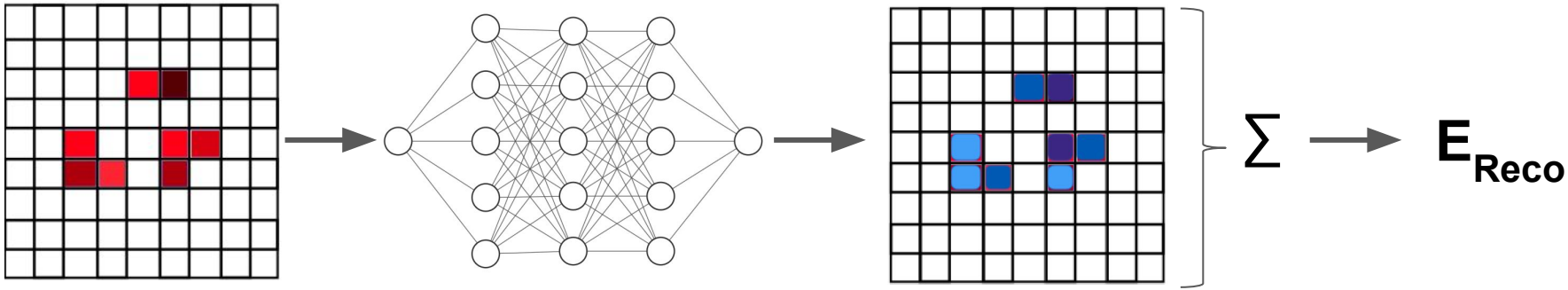
Small Locally Connected Layers

- Fully Connected Network
  for each input channel
  - Learns per channel MIP to TeV conversion
- Implemented with Locally Connected Layer
  - Unshared weights
  - Kernel size = (1)

$$I'_{i,b} = \sum_{a} \left(I_{i,a} \circ W^{b}_{i,a} \right) + B^{b}_{i}$$

I': Output matrix
I: Input matrix
W: Weights
B: Bias

i: Input length (for 1D)
a: Input channel dimensions
b: Output channel dimensions



Input calorimeter image [MIP]

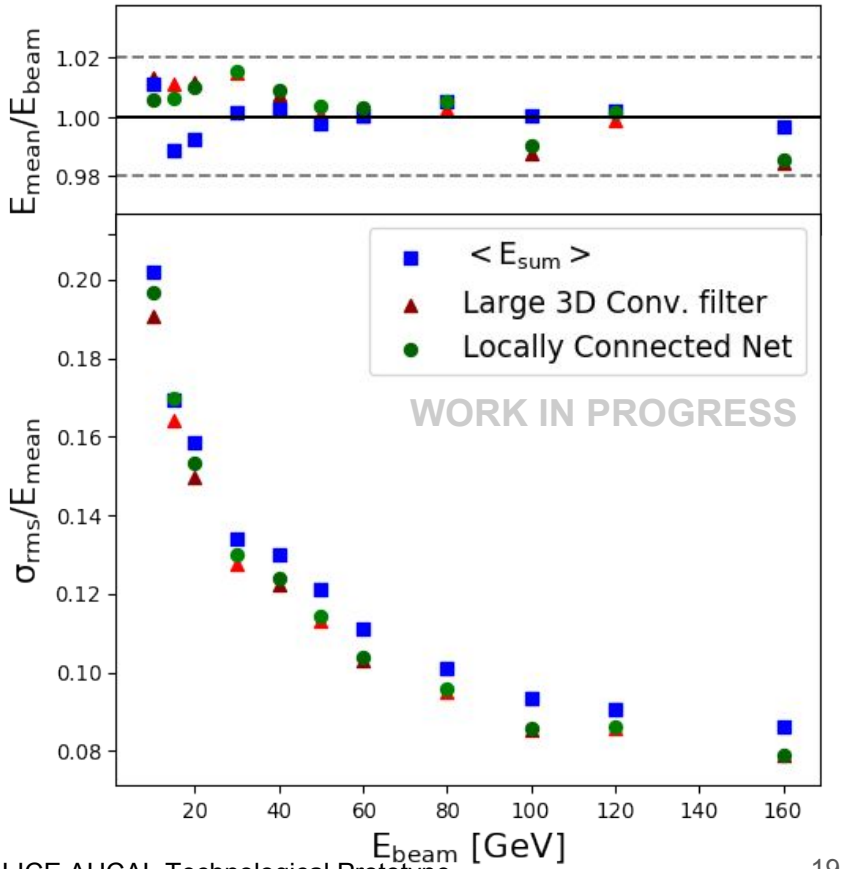
Fully Connected network for each channel

Converted calorimeter image [TeV]

Small Locally Connected Layers

- Network architecture:
  - 1 layer with 32 channels
  - Linear activation function & no bias
  - Training procedure:
    i. Train with shared weights (Conv. Layer)
    ii. Transfer weights to Locally Connected Layer
    iii. Train transferred unshared weights

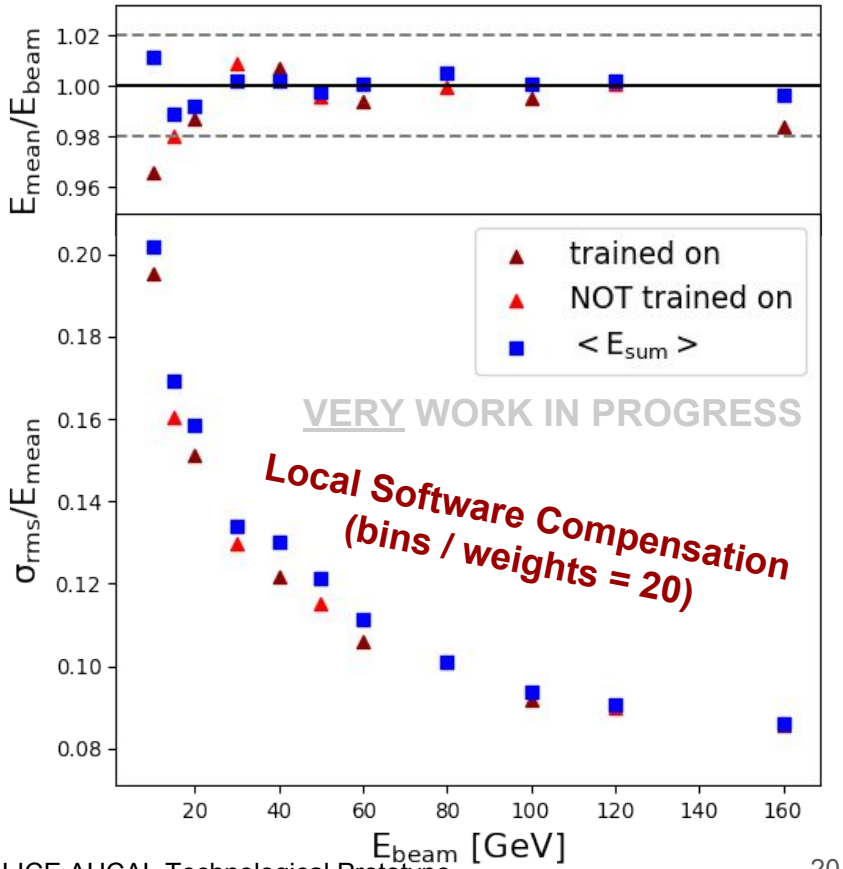
➔ Network performs worse for low energies & similar for high energies in comparison to implementation with large Convolutional Layer



Current Studies

- More layers & training on simulation samples with 1 GeV beam energy steps
  → Currently working on event selection
- Next: Locally Connected architecture in front of Convolutional Neural Network

- For comparison:
  Implementation of Local Software Compensation as Keras / Tensorflow
  → Non-linearity at low energies



Summary & Outlook

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CALICE

Summary:

- Highly granular calorimeters offer interesting applications for deep learning studies in particle physics
- Test beam data offers possibility for training on labeled experimental data
- Simple network architecture can interpolate for energies not trained on

Outlook:

- Deeper architectures for energy reconstruction; comparison to offline compensation algorithms
- Adding timing information to input
- DNN for shower separation

S Outlook

Thank you!

Summary:

- Highly granular calorimeters offer interesting applications for deep learning studies in particle physics
- Test beam data offers possibility for training on labeled experimental data
- Simple network architecture can interpolate for energies not trained on

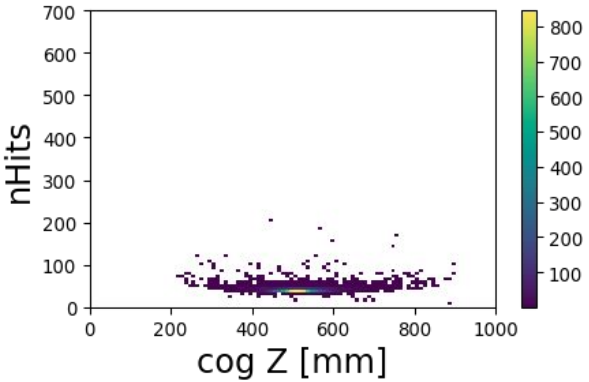
Outlook:

- Deeper architectures for energy reconstruction; comparison to offline compensation algorithms
- Adding timing information to input
- DNN for shower separation

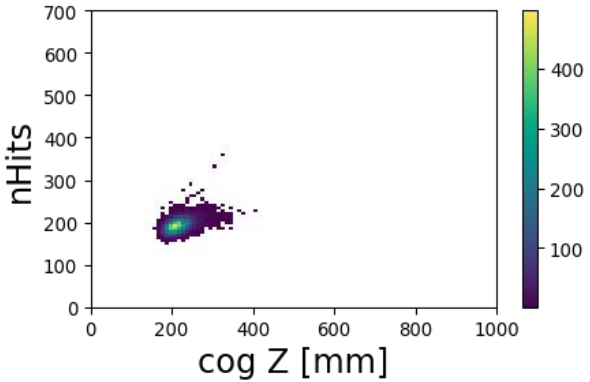
Bonus Slides

2.) Classification: Particle ID

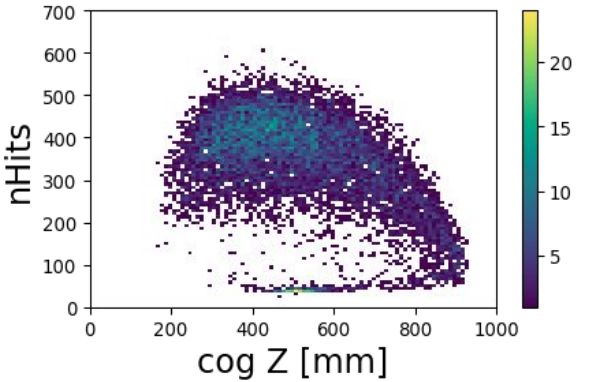
Task: Classification of 50 GeV events into muons, electrons or pions



Muons



Electrons



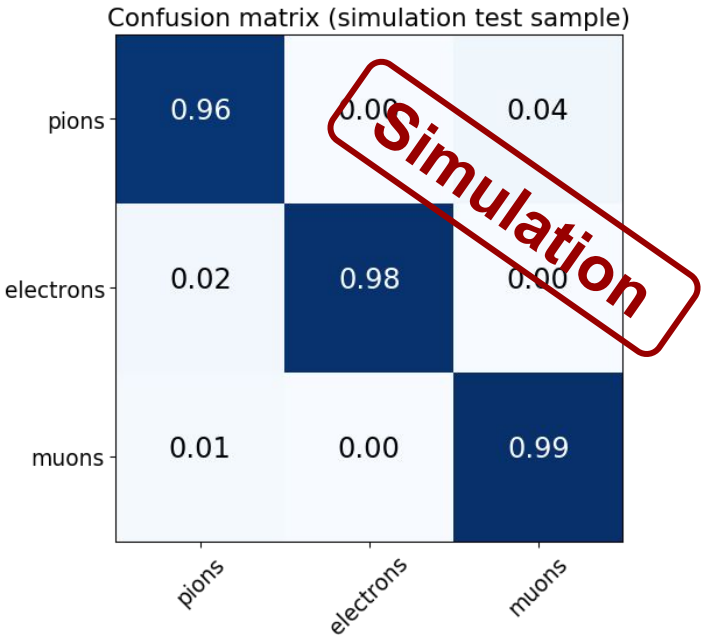
Pions

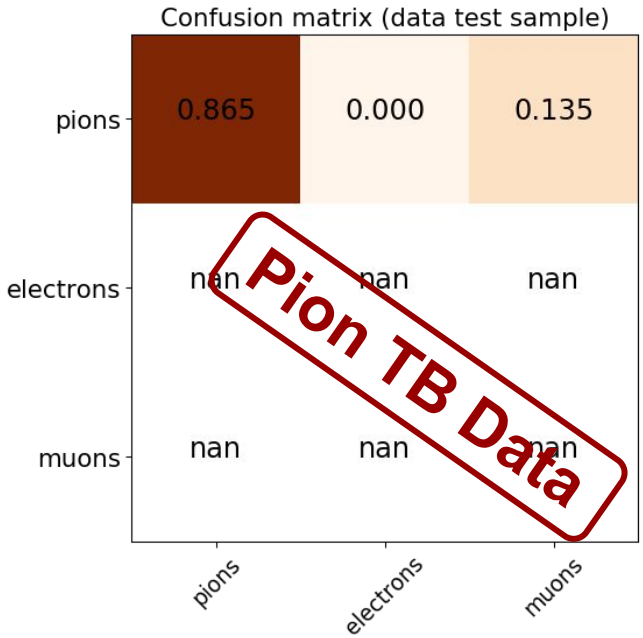
2.) Classification: Particle ID

Classification: Identify particle as Pion, Electron or Muon

VGG-16 inspired network was used (CNN, filter size = (3,3,3,X))

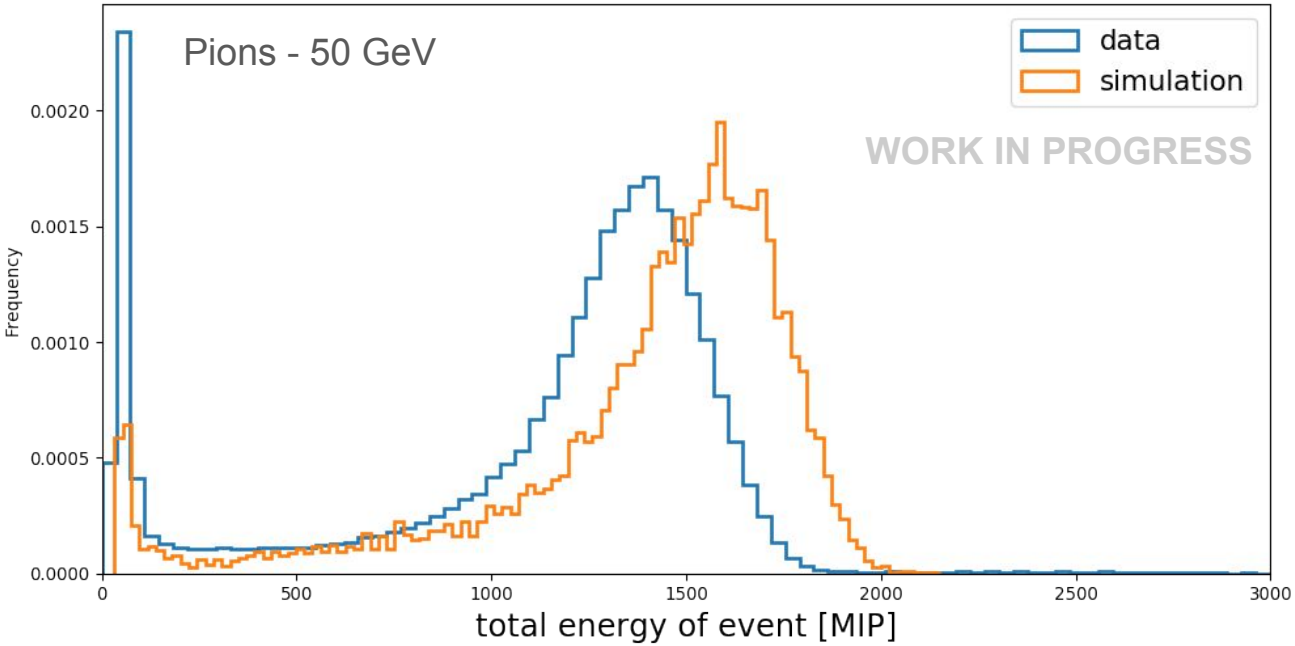
Trained on simulation → Tested on simulation & data





- ➔ CNN classifies simulated particles well
- ➔ Simulation needs to improve to apply CNN classifier to data

2.) Classification: Particle ID

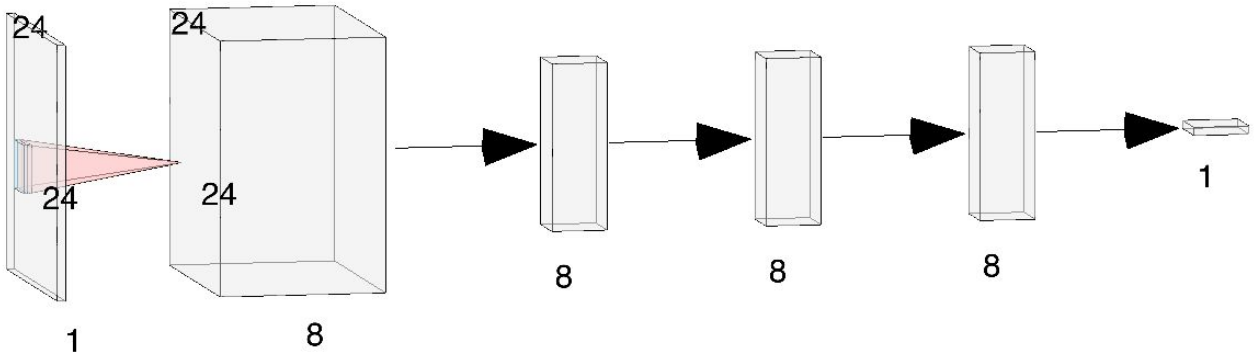


➔ Difference in total energy between simulation and data

➔ Need to improve simulation for trustworthy training

Studies with Deep Neural Networks

- Deep Neural Networks (DNN) can learn complex functions
- Convolutional layer + fully connected network
- Many trainable parameters (here: 1.4 M)
- Learns “trained on” energies, does not work well for “NOT trained on” ones

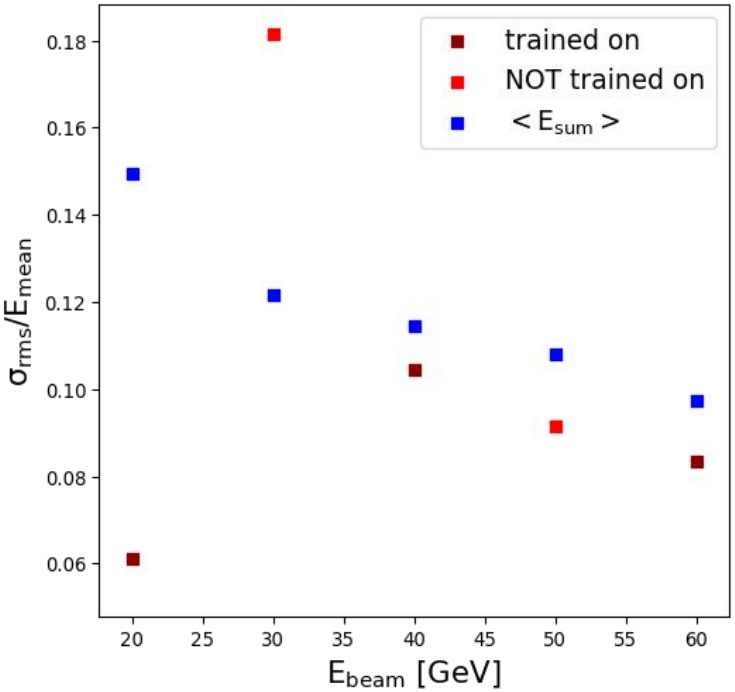


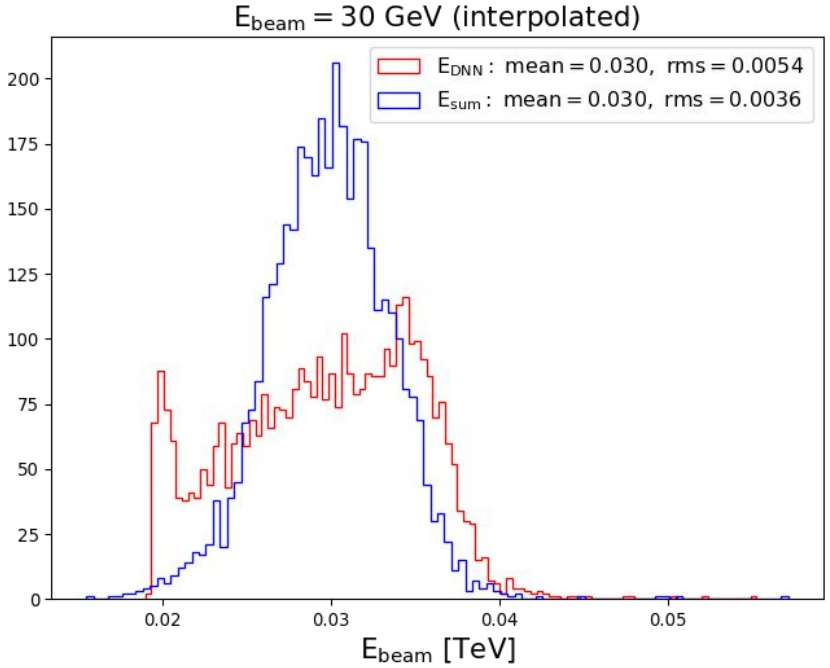
omitting 3rd dimension for visibility - 1 out of 38 layers

(3D conv. Kernel size = (8, 6, 6, 12))

Deep Network Performance

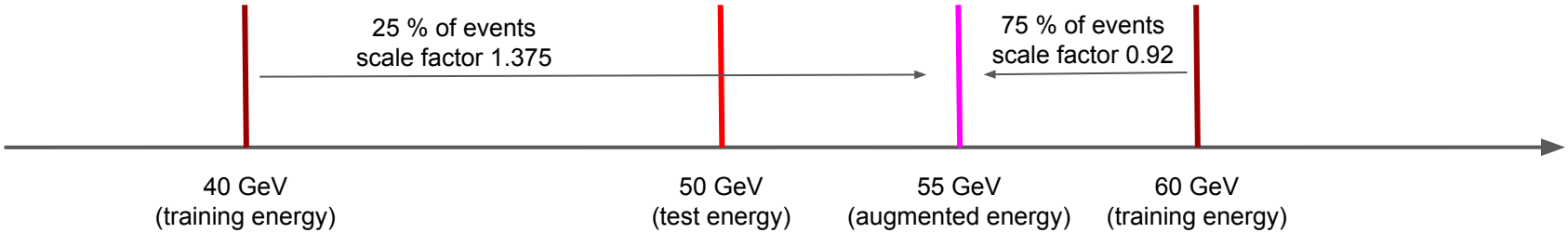
For data set: 20 - 60 GeV → Bad performance for a “NOT trained on” energy





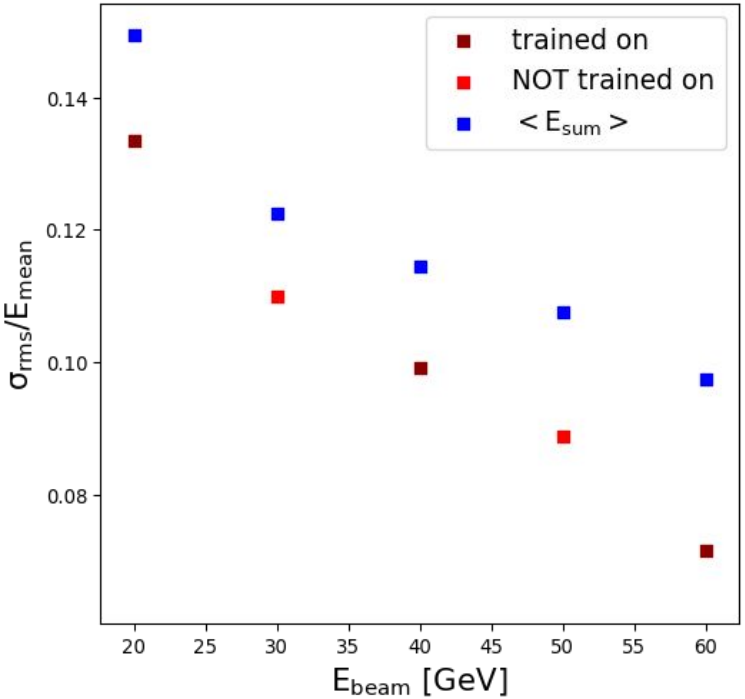
Hit Energy Augmentation

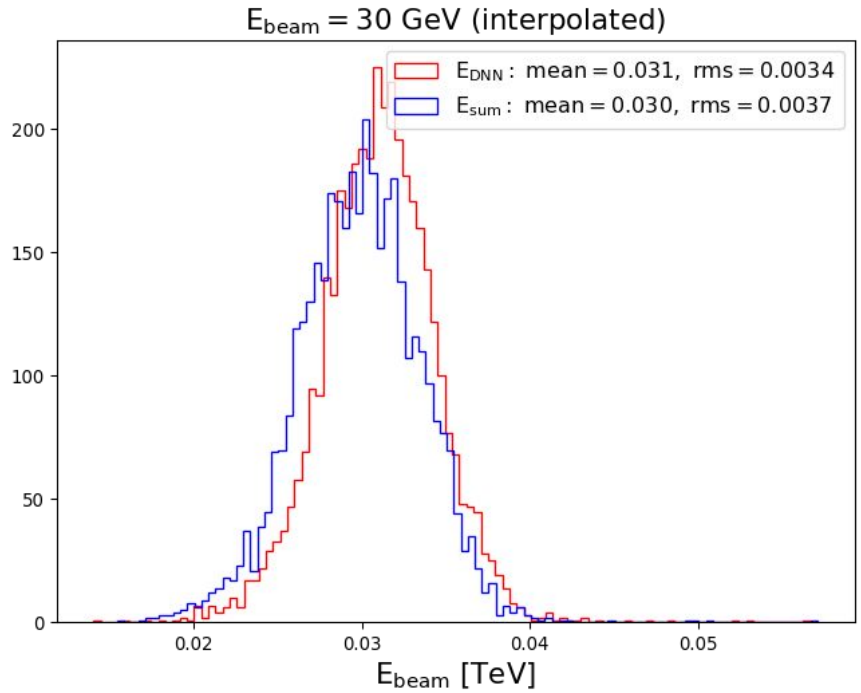
- Assuming linear scaling of hit energies with beam energy
- Filling energy regions without data by linear scaling of data
  - From 1 - 169 GeV in 1 GeV steps
  - Choosing energy for scaling weighted by energy gap
  - Augmented training set = 3x regular training set



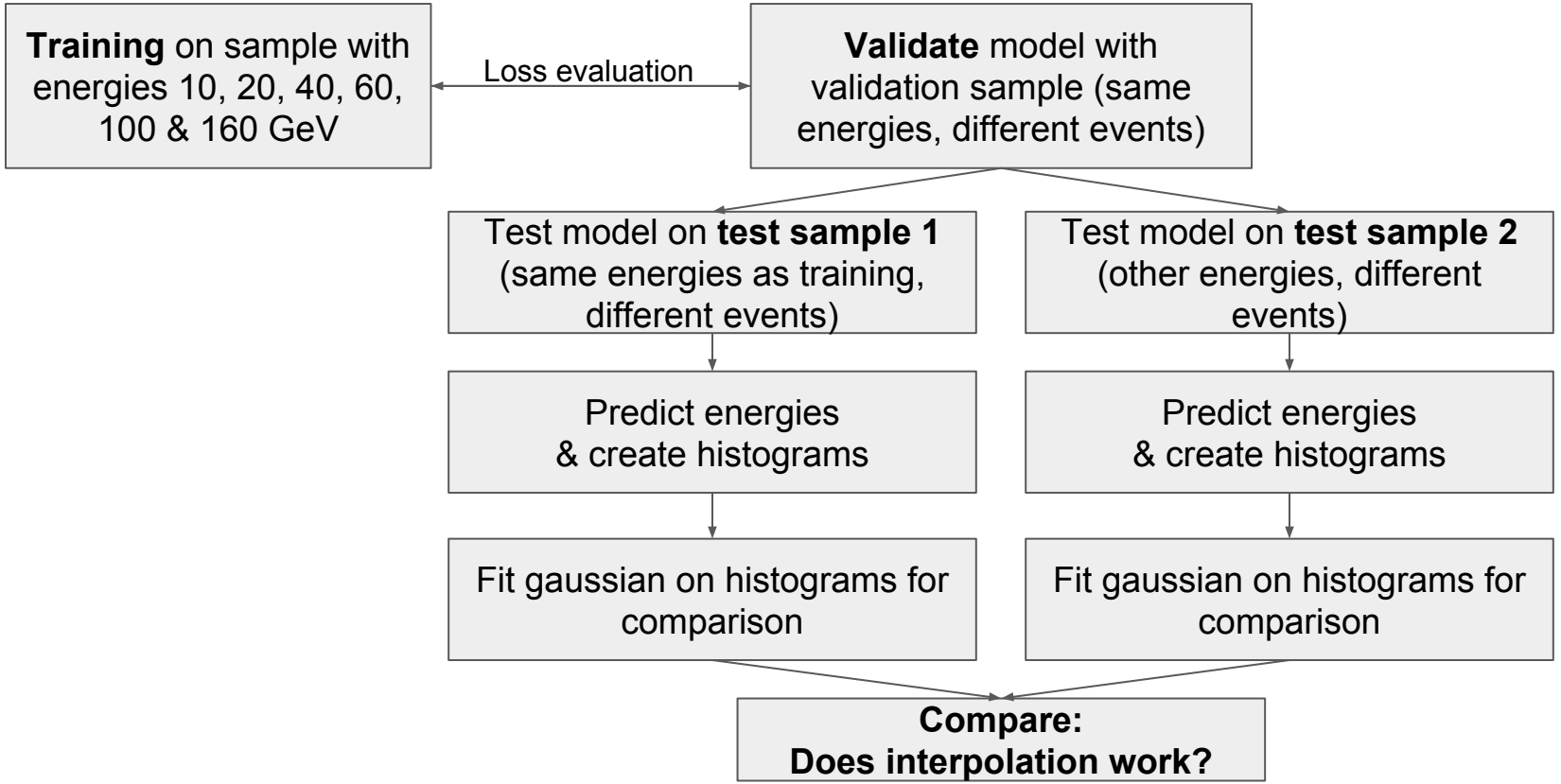
Hit Energy Augmentation

For data set: 20 - 60 GeV → Better performance for “NOT trained on” energies





Training & Testing Process



Loss Function (performance evaluation)

Example 1) Mean Squared Error (MSE):

$$L(E_{i,true}, E_{i,pred}) = \frac{1}{N} \sum_i \left(E_{i,pred} - E_{i,true}\right)^2$$

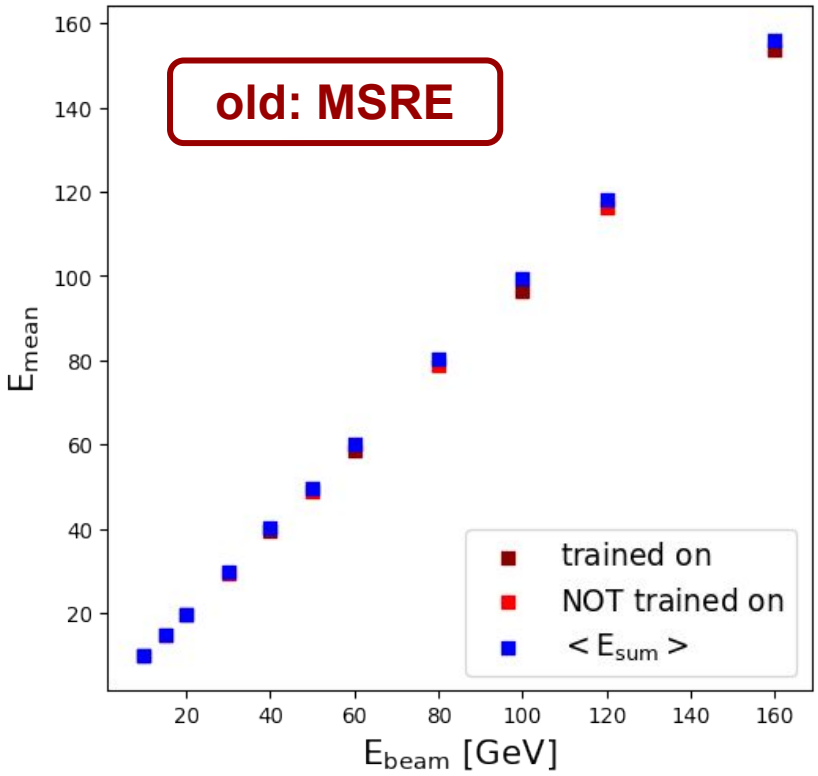
Example 2) Mean Squared Relative Error (MSRE):

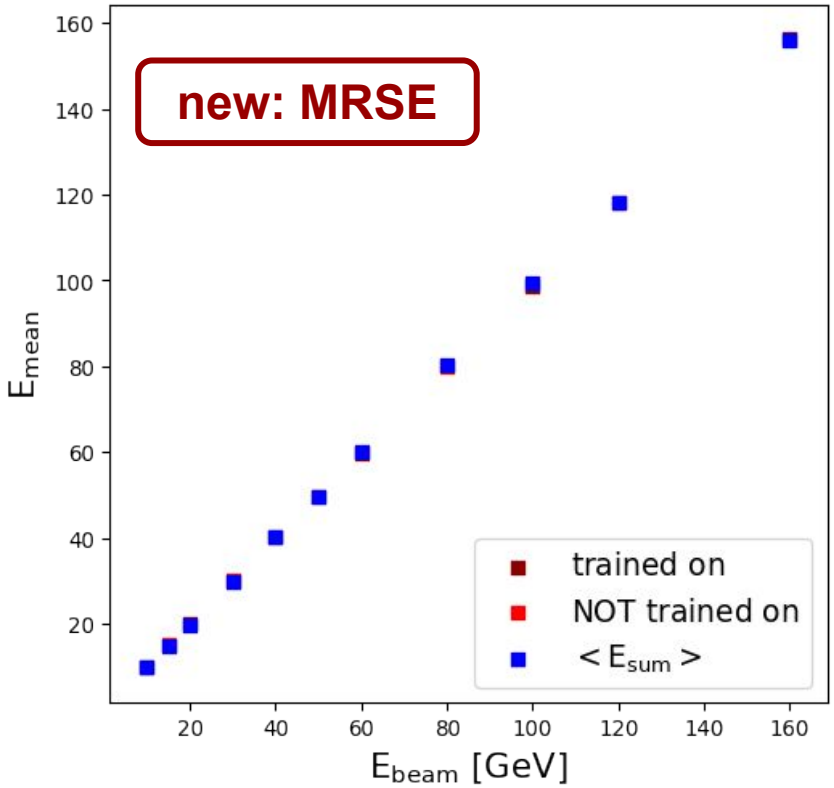
$$L(E_{i,true}, E_{i,pred}) = \frac{1}{N} \sum_i \left(\frac{E_{i,pred} - E_{i,true}}{E_{i,true}}\right)^2$$

Example 3) Mean Relative Squared Error (MRSE):

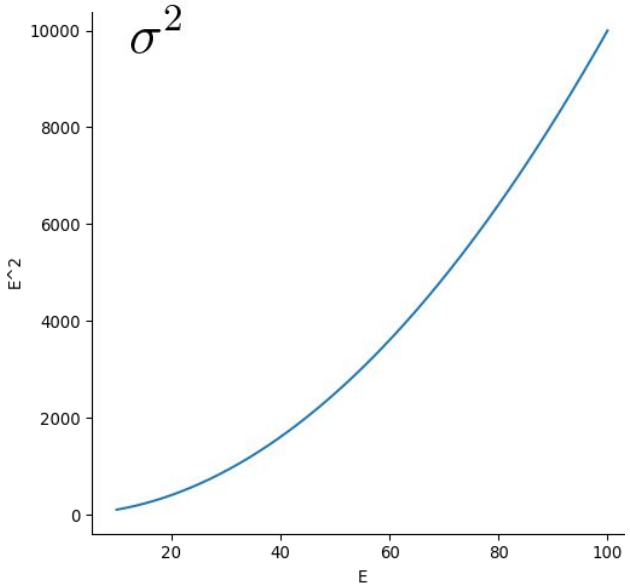
$$L(E_{i,true}, E_{i,pred}) = \frac{1}{N} \sum_i \left(\frac{E_{i,true} - E_{i,pred}}{\sqrt{E_{i,true}}}\right)^2$$

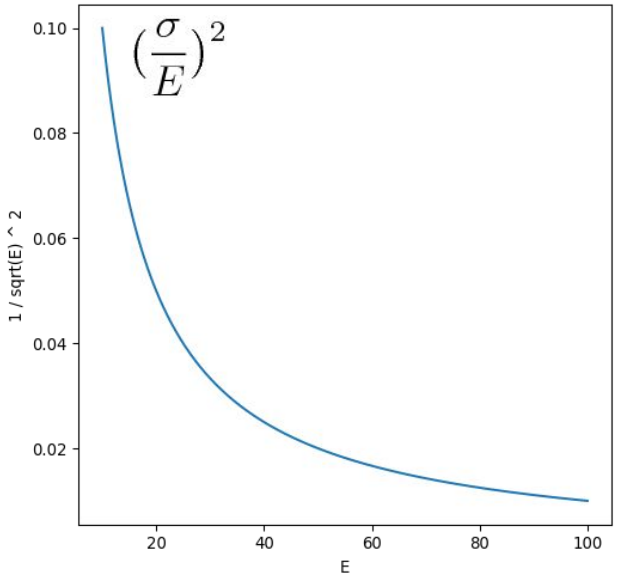
MSRE vs MRSE

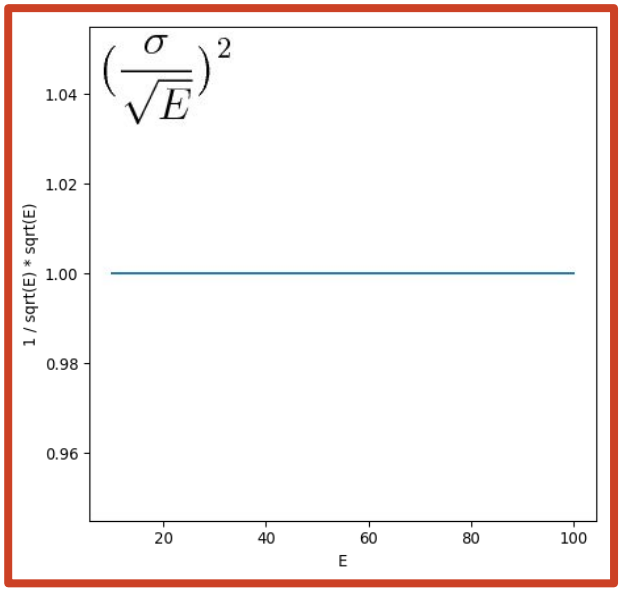

old: MSRE
trained on
NOT trained on
< E_sum >
E_mean
E_beam [GeV]


new: MRSE
trained on
NOT trained on
< E_sum >
E_mean
E_beam [GeV]

Function Comparison

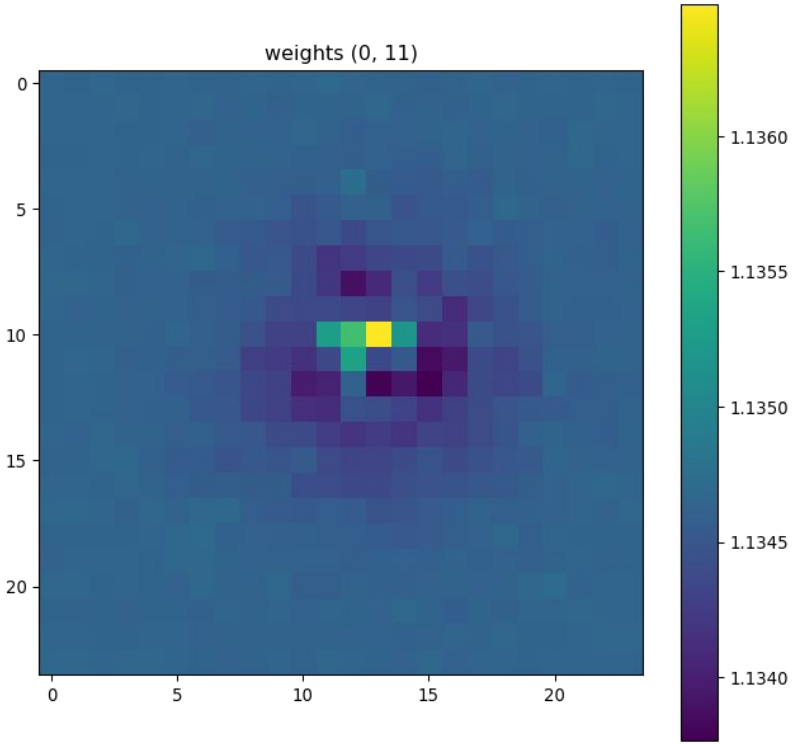




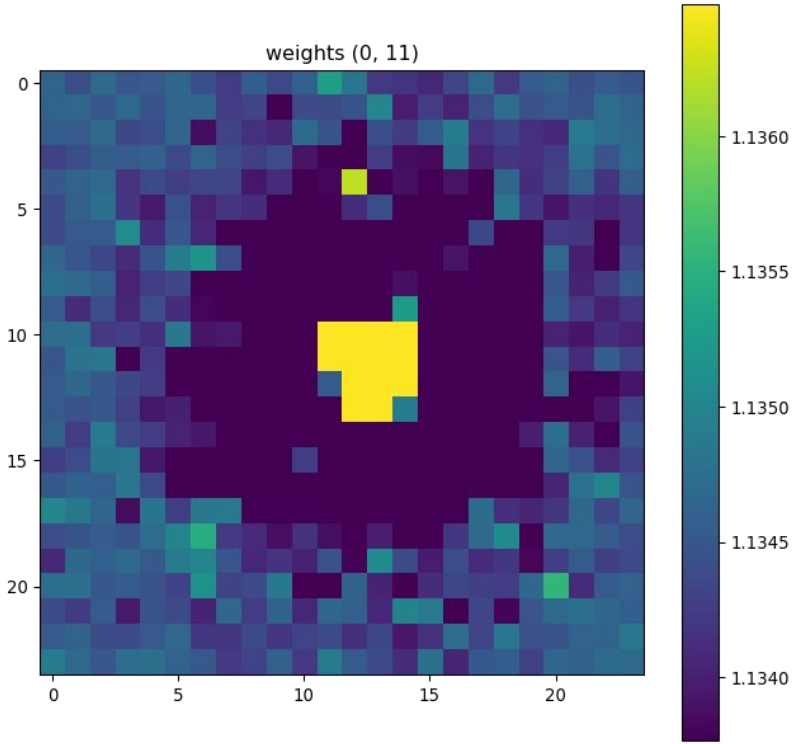


LC weight heatmaps (after transfer)

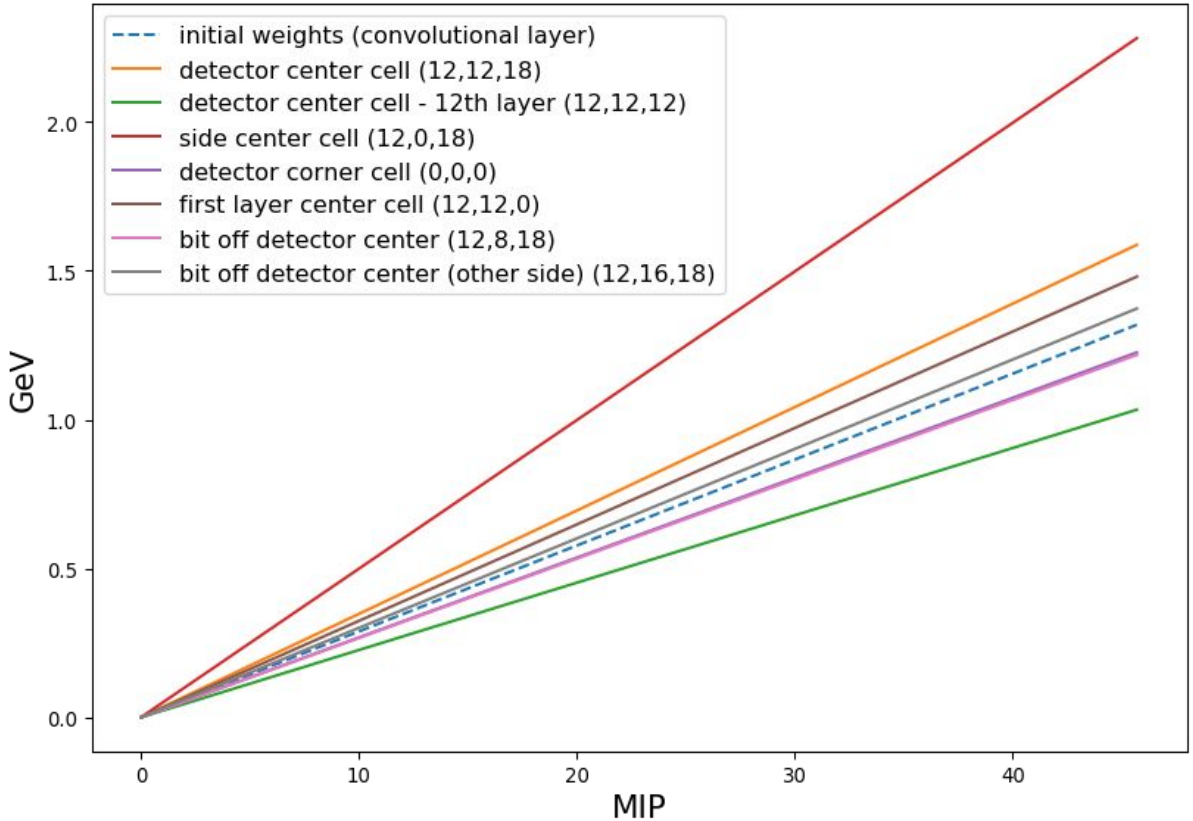
detector_layer 18, network_layer 1, epoch 001



detector_layer 18, network_layer 1, epoch 009



LC calibration functions



Test Beam Setup: Technological Prototype @ SPS