Deep Learning based Energy Reconstruction for the CALICE AHCAL

Erik Buhmann[1], Erika Garutti, Gregor Kasieczka

ML4Jets 2020 @ NYU
January 17, 2020

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

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

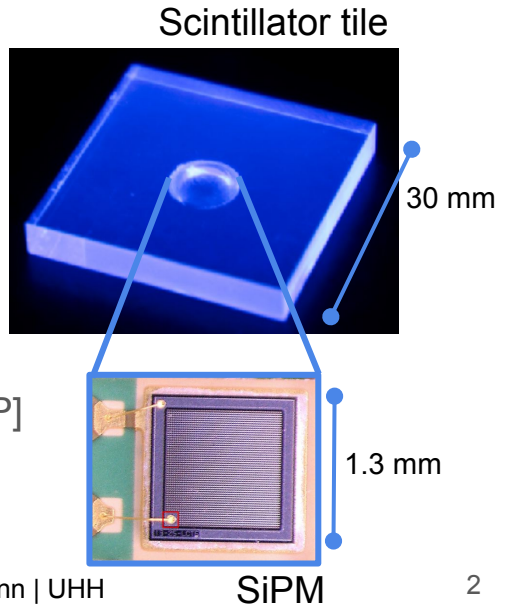
CALICE AHCAL Technology

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

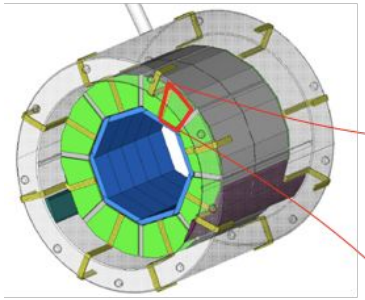
"**CA**lorimeter for **LI**near **C**ollider **E**xperiment" collaboration:
Development of calorimeter technology for future e^+e^--collider with high granularity optimized for the Particle Flow Approach

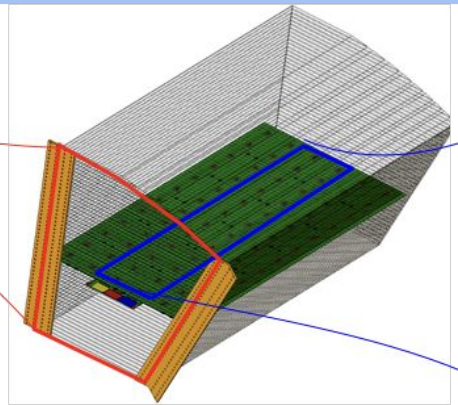
Analog **H**adron **CAL**orimeter technology:

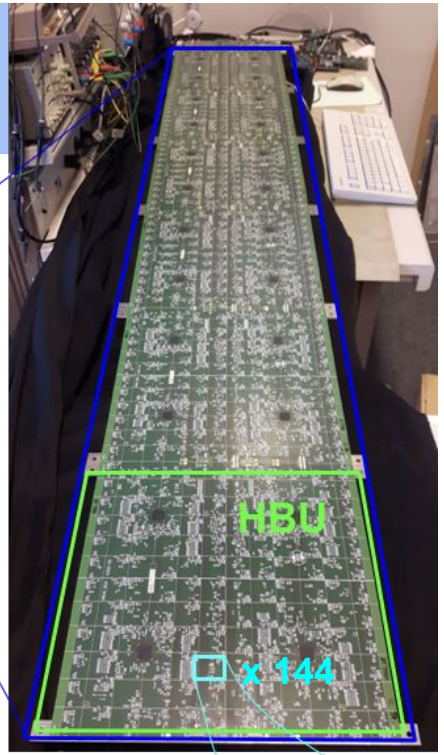
- Signal generation in scintillator + **Si**licon-**P**hoto**M**ultiplier
  - *"SiPM-on-tile"* technology (also applied to CMS HGCAL)
- **Readout per tile: Energy & Time**
  - **A**nalog to **D**igital **C**onverter (ADC) → **M**inimum **I**onising **P**article [MIP]
  - **T**ime to **D**igital **C**onverter (TDC) → nanoseconds [ns]

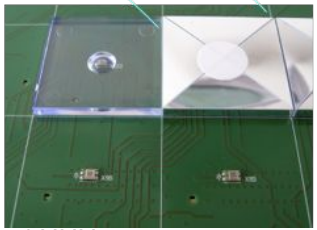


CALICE AHCAL Design









ILD HCAL Design:

→ ~ 8 mio. channels of 3 x 3 x 0.3 cm^3

2018 Engineering Prototype:

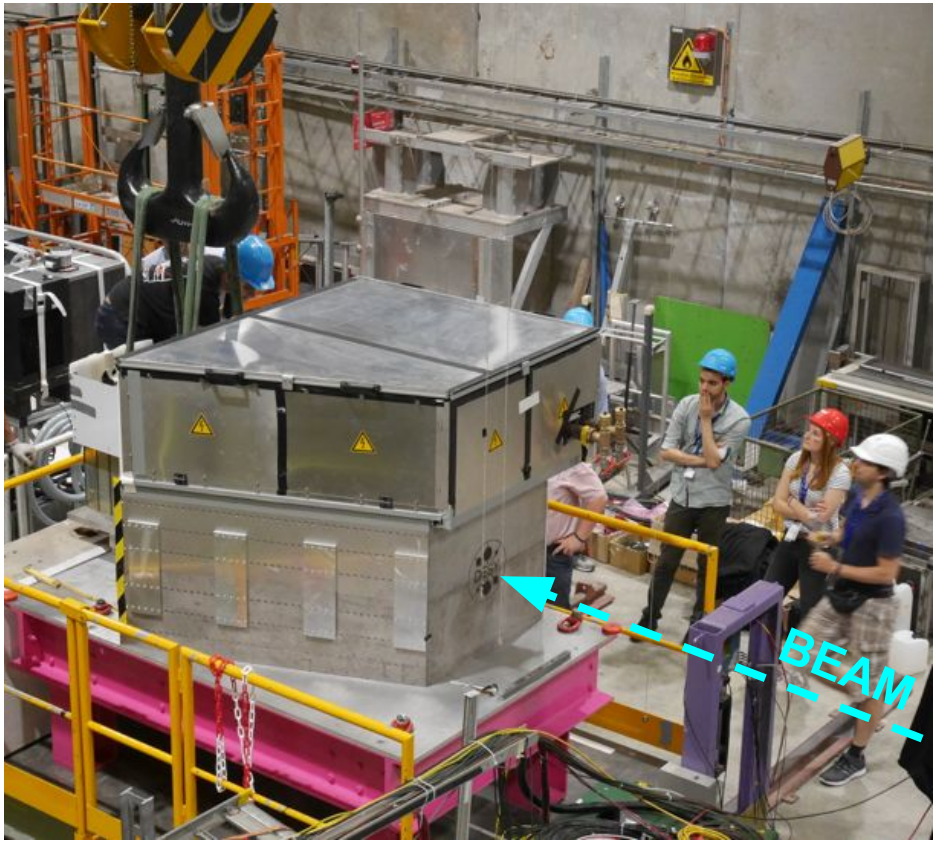
→ 22,000 tiles of 3 x 3 x 0.3 cm^3

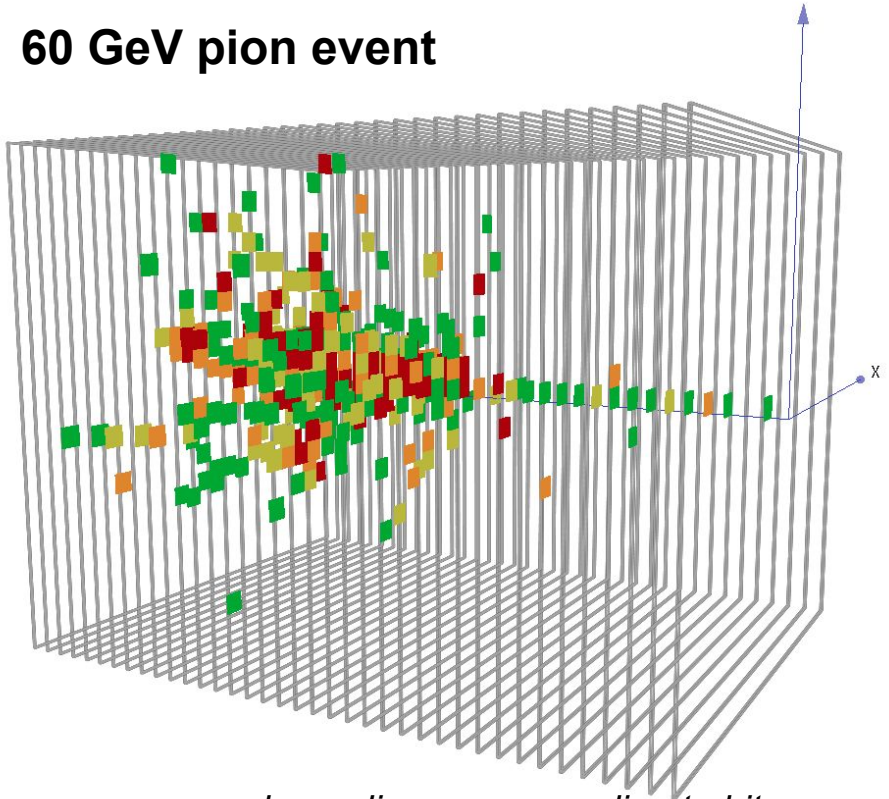
→ ≈ 24 x 24 x 38 scintillator tiles

→ Stainless steel as absorber (~ 4 λ_{int})

→ non-compensating (e/h > 1)

Test Beam Setup in May 2018: Engineering Prototype @ CERN SPS





color coding corresponding to hit energy

Energy Reconstruction

Non-compensating calorimeter: e/h > 1

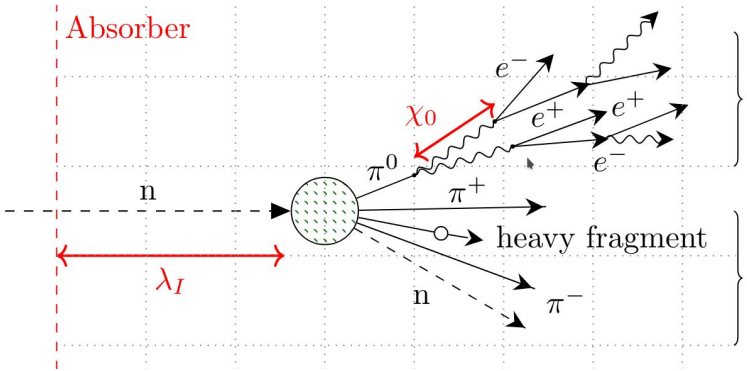
→ hit energie needs reweighting

→ hit timing helps: EM hits fast → down-weighting fast hits

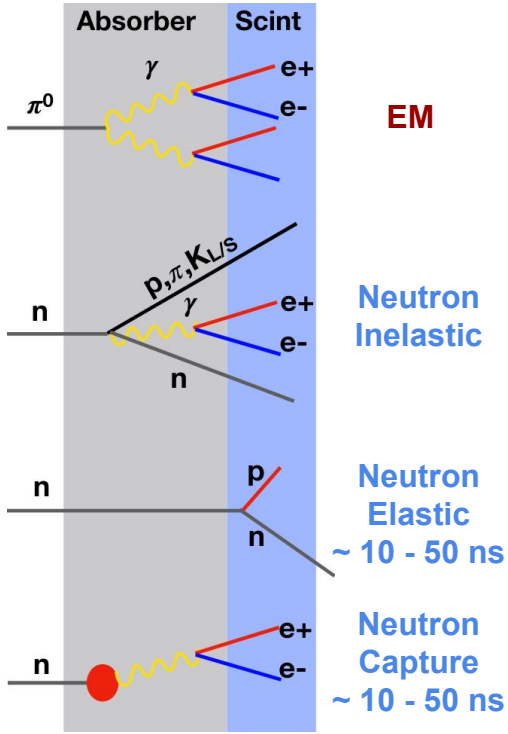
hadr. hits late → up-weighting late hits

(procedure called *offline* or *software compensation*)

Hadr. shower:



Timing of EM and hadr. processes:



Energy Reconstruction

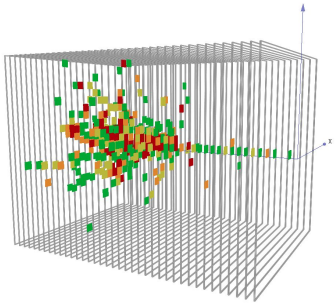
Can we reconstruct the energy of these "5D shower images" using Deep Learning?

Deep Learning regression task:

→ supervised learning

→ using low-level detector information

→ training on labeled test beam data possible



E_{reco}

Compare to standard reconstruction & local software compensation (CALICE)

$$E_{sum} = \sum_i E_i \qquad E_{reco} = \sum_i E_i w_i(E_i, E_{sum})$$

E_i: hit energy in channel i

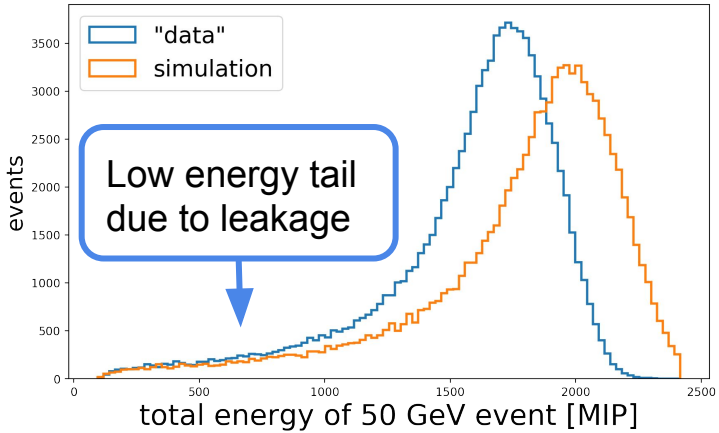
→ implementation with 24 weights

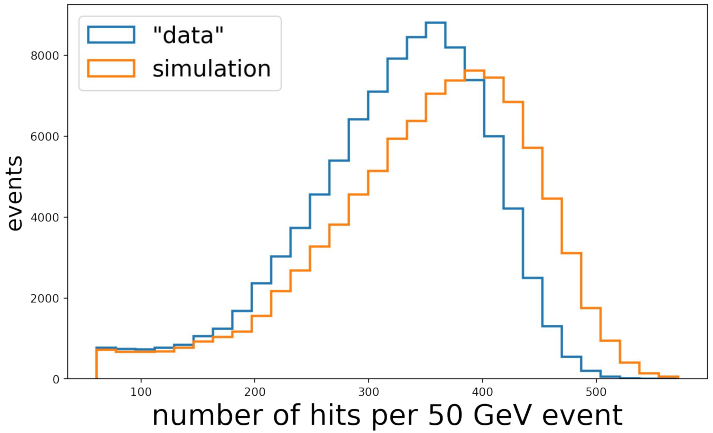
→ hit energies position independent

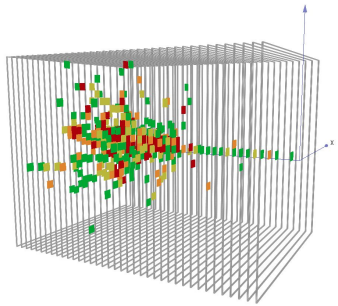
Data / Simulation Processing

Pion "Data"
- Test beam data in sparse 10 GeV steps
- Taken May 2018 at CERN's SPS
- Pending collaboration approval!
- cannot show final energy & time resolution
→ "Data" shown here is altered simulation to reproduce observed data-MC differences

Simulation
- Pion beam energies 10-80 GeV in 1 GeV steps
- Geant4 simulation + digitization
- Event reconstruction performed with same software and calibrations as for data
- Hit timing: 1 ns Gaussian smearing







Data / Simulation Processing

Cuts

- Shower start in first 5 layers
- Number of hits > 60 (rejecting 'muon-like' pions)
- 16,000 samples per beam energy in training set
- 2 test samples ('trained on" & "NOT trained on")

→ **Can the network interpolate for energies it was NOT trained on?**

Implementation

- Shower images as (24,24,38,2) arrays
- With hit energy & time at (X, Y, Z) coordinate
- Ground truth values: known beam energies
- Loss function: **M**ean **A**bsolute **R**elative **E**rror

$$L_{MARE} = \frac{1}{N} \sum_i \left| \frac{E_{reco,i} - E_{beam,i}}{\sqrt{E_{beam,i}}} \right|$$

E_i: energy of event i

Network Architectures

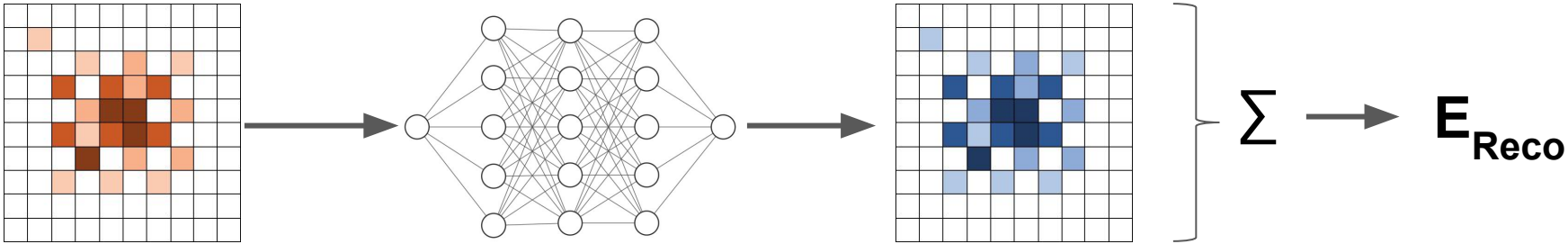
1. **L**ocally **C**onnected **N**etwork (LCN)
2. **C**onvolutional **N**eural **N**etwork (CNN)
+ versions using both energy & timing information

Locally Connected Network (LCN)

- Implemented with **1** Locally Connected Layer of filter size (1) → "**LC1**"
  - *~ 22,000 weights (no bias)*
  - Channel-wise energy independent weights w_i : (channel-wise linear calibration)

$$E_{reco} = \sum_i E_i w_i$$

- And with **4** Locally Connected Layer of filter size (1) → "**LC4**"
  - layers 16 → 16 → 16 → 1, LeakyReLU, *12M weights*
    - ➢ Channel-wise non-linear calibration



Input calorimeter image [MIP]

Fully Connected network for each channel

Calibrated calorimeter image [GeV]

Locally Connected Network (LCN)

- Implemented with **1** Locally Connected Layer of filter size (1) → "**LC1**"
  - *~ 22,000 weights (no bias)*
  - Channel-wise energy independent weights w_i : (channel-wise linear calibration)

$$E_{reco} = \sum_i E_i w_i$$

- And with **4** Locally Connected Layer of filter size (1) → "**LC4**"
  - layers 16 → 16 → 16 → 1, LeakyReLU, *12M weights*
    - ➢ Channel-wise non-linear calibration
- **Add timing information:**

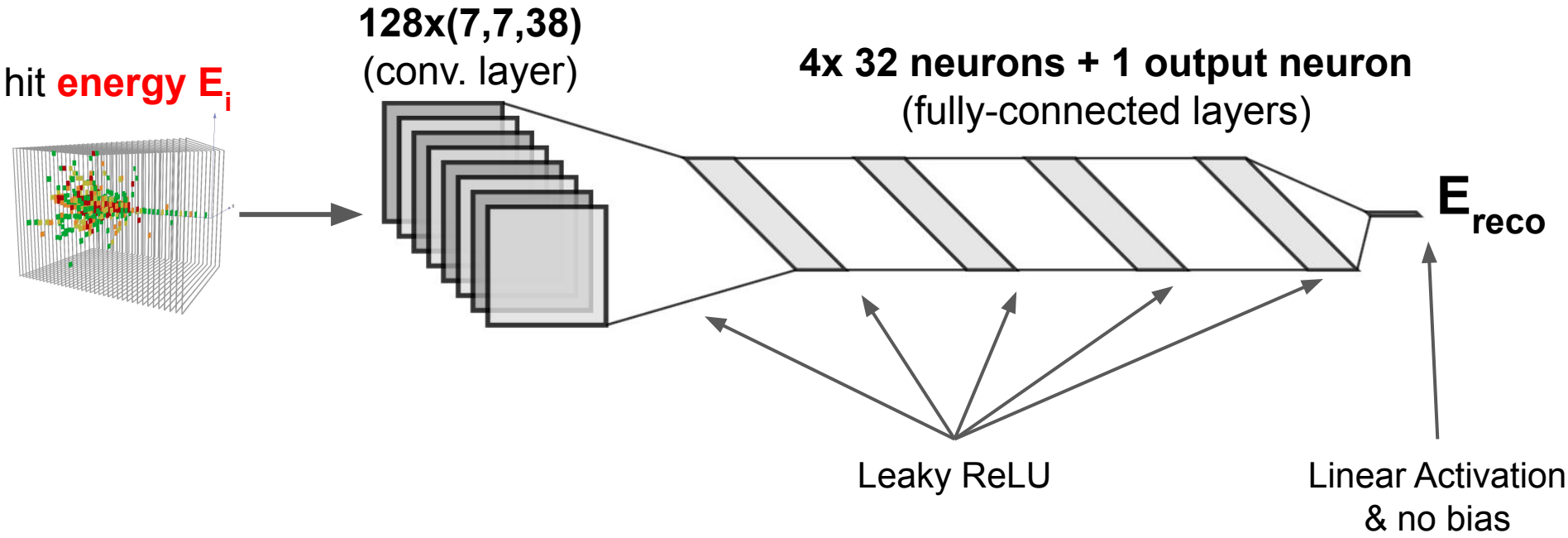
  Introduce timing calibration factor t_i with a channel-wise CNN of filter size (1) (layers 128 → 64 → 1, LeakyReLu, *10,000 weights*) → "**LC1+T**"

$$E_{reco} = \sum_i E_i w_i t(T_i)$$

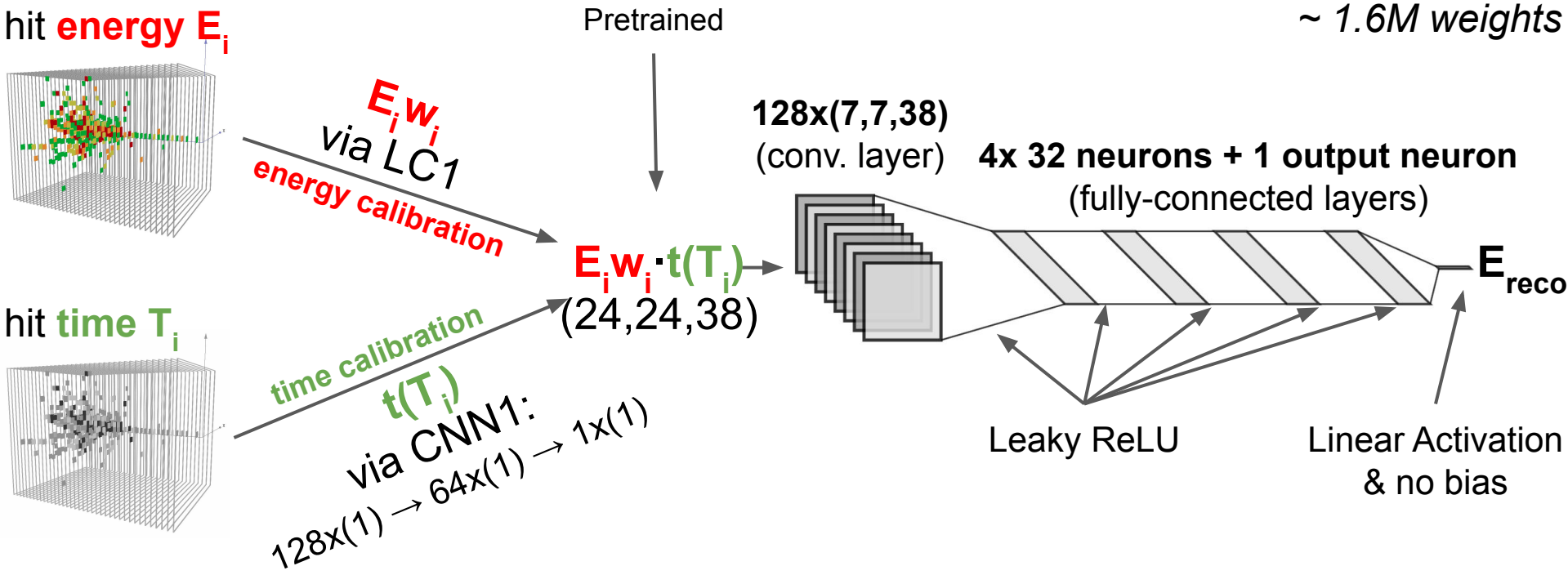
T_i: hit time in channel *i*

CNN Architecture

- 1x 3D convolutional layer + 5x fully-connected layers → "**CNN**" *~ 1.6M weights*



Merged Architecture (LC1+T+CNN)



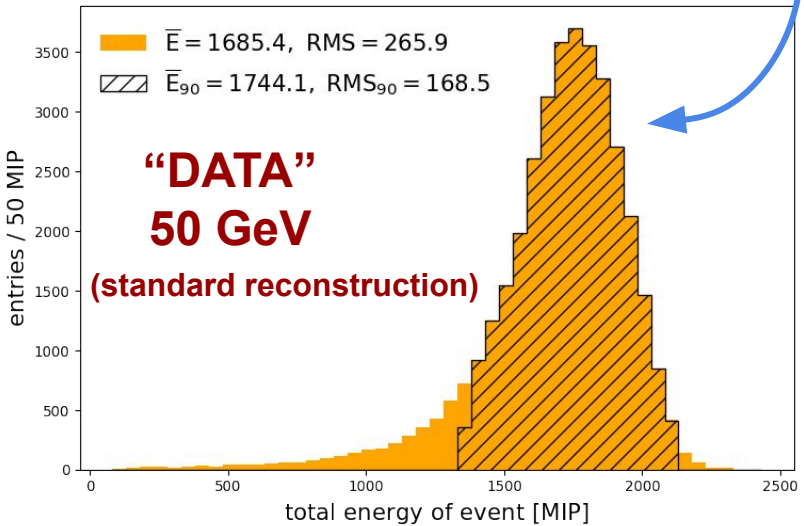
Results: Energy Resolution Improvements

1. **L**ocally **C**onnected **N**etwork (LCN)
2. **C**onvolutional **N**eural **N**etwork (CNN)

\+ trained on either simulation or “data”

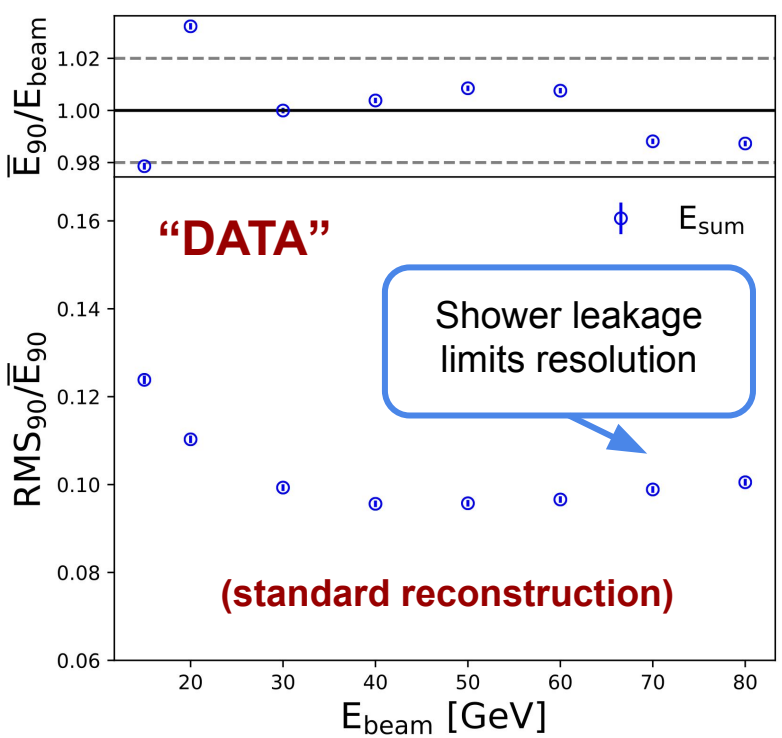
RMS90 & energy resolution plots

RMS90 / MEAN90:
RMS / MEAN of the smallest histogram window in which 90 % of all events are contained.

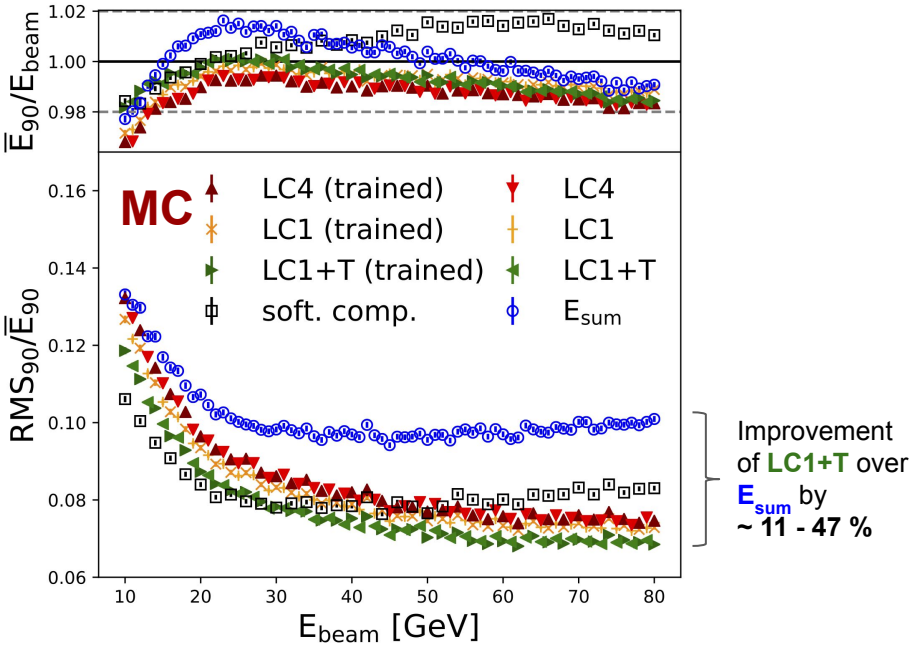


linearity

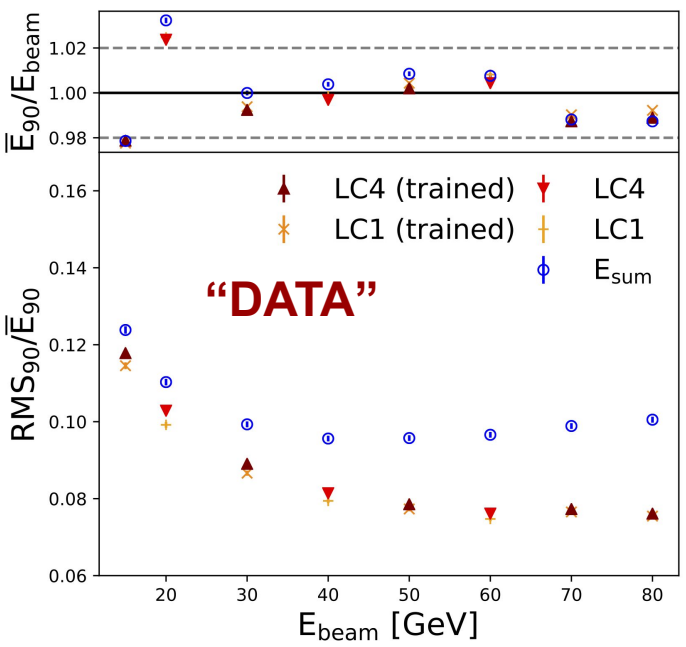
resolution



Locally Connected Networks (LCN)



Improvement of **LC1+T** over E_{sum} by **~ 11 - 47 %**



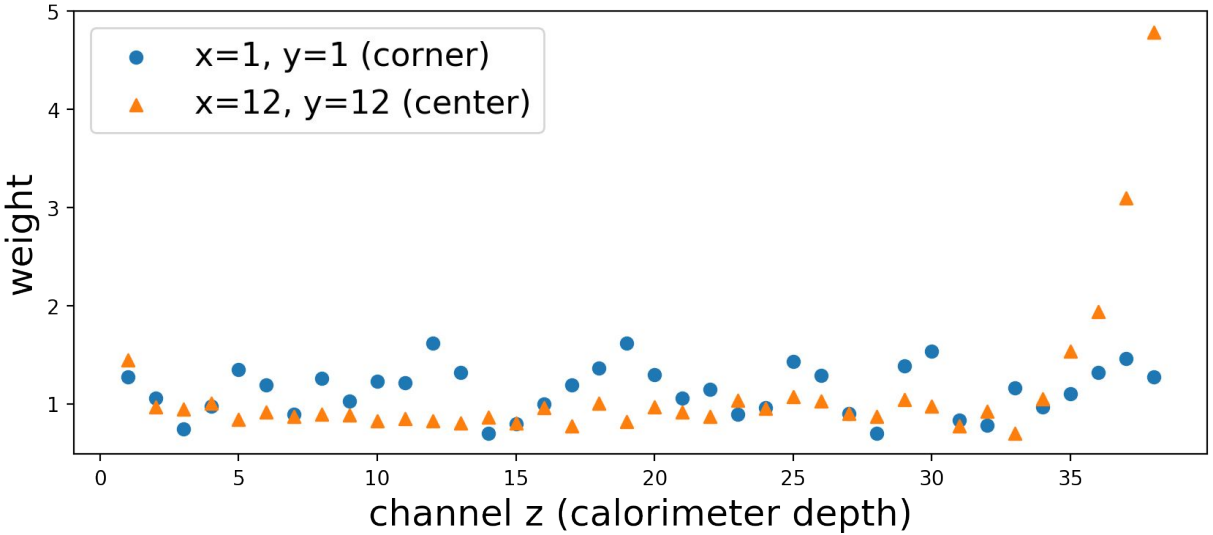
Improvement of **LC1** over E_{sum} by **~ 8 - 33 %**

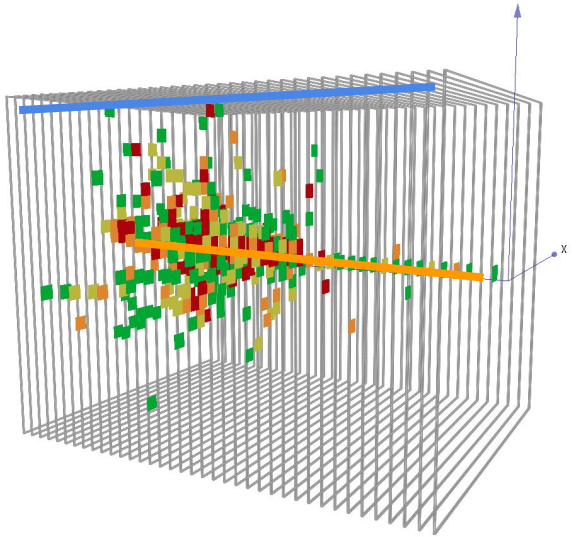
- No difference between test sets observable
- **T**iming improves resolution over **LC1** by 8%

- Trainable for sparse datasets
- **LC1** better than **LC4**

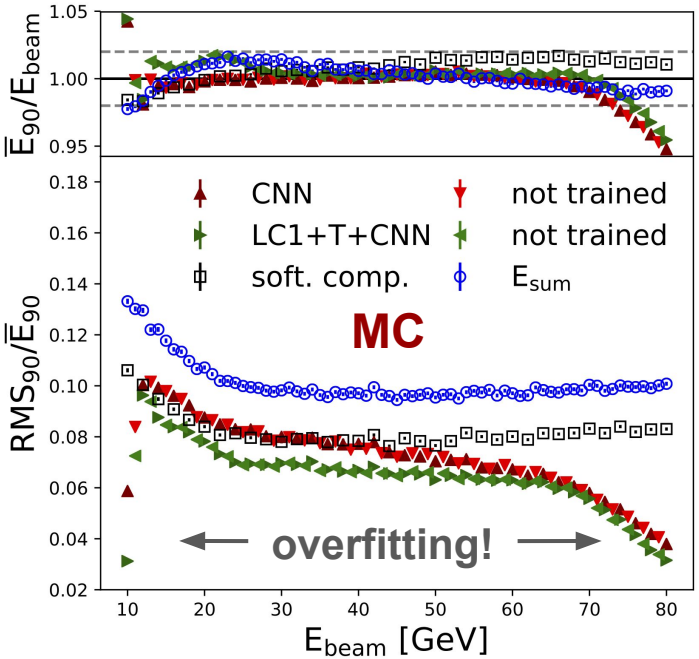
LCN: Leakage Compensation

- Hit energies in last layers are weighted higher



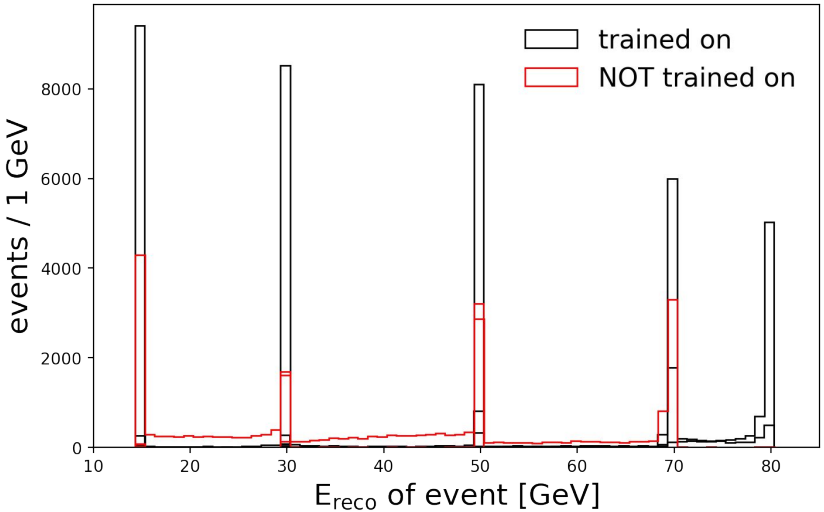


CNN Architectures



- Trainable with 2 GeV steps,
  fails for 10 GeV steps (like “data”)
- Overfits on the edges of trained energy space

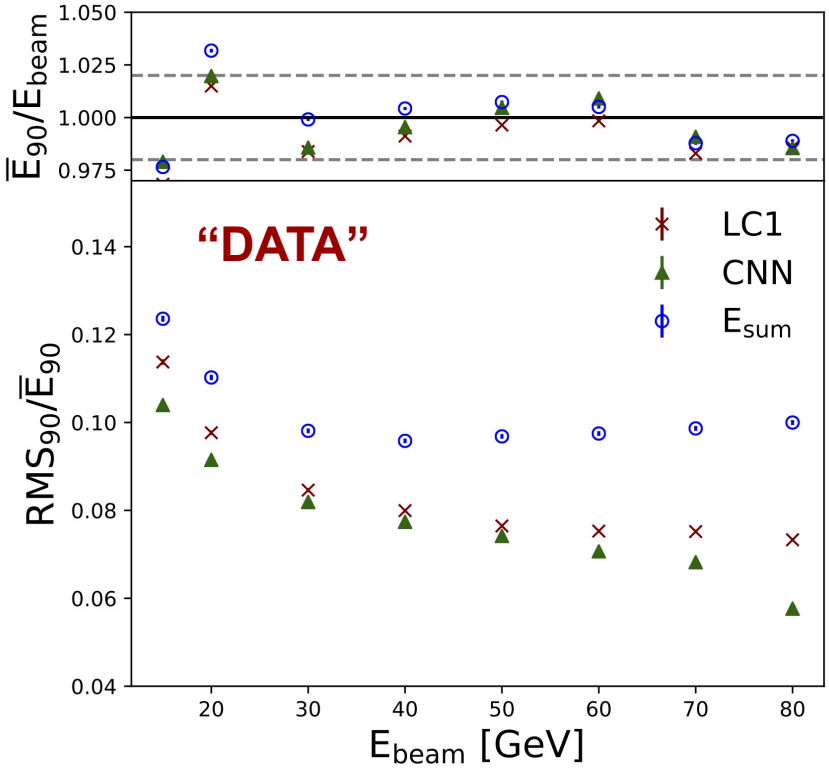
Fails for “DATA”:



- Interpolation fails for sparse dataset

➢ Best performance by **LC1+T+CNN** for MC

Trained on MC, Test on “Data”



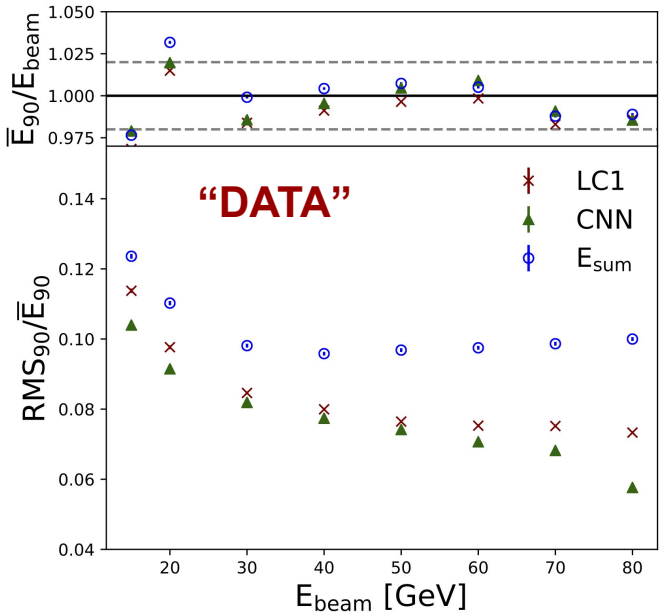
- Networks trained on MC with 2 GeV steps
- Tested on sparse “data” with 10 GeV steps

➢ Energy reconstruction improves
  - for LC1 by ~ 10 - 35 %
  - for CNN by ~ 20 - 75 % (for 10 - 80 GeV)

➢ “Data” reconstructable with MC trained networks

Conclusions

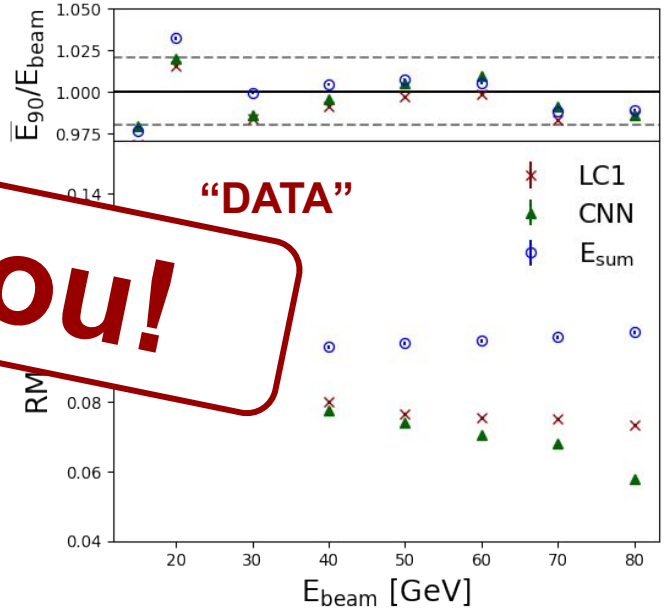
- ➔ Simple model (LCN) trainable on sparse dataset (20 GeV steps)
- ➔ Complex model (CNN) trainable only on dense dataset (2 GeV steps)
- ➔ Networks trained on dense MC dataset successfully transfered to improve sparse “data” reconstruction
- ➔ DNN architectures can improve achievable energy resolution significantly
- ➔ Including timing information improves reconstruction



Conclusions

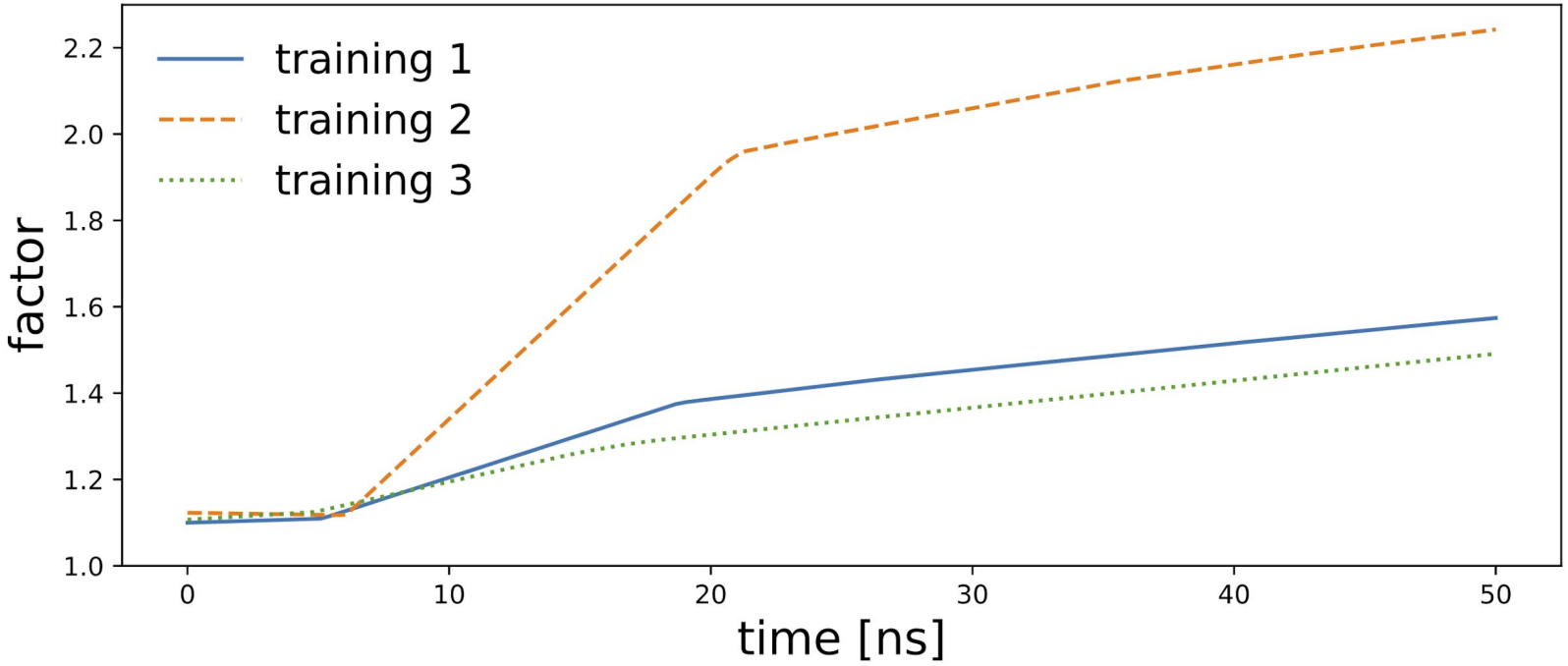
- ➔ Simple model (LCN) trainable on sparse dataset (20 GeV steps)
- ➔ Complex model (CN… only on dense dataset (2 G…
- ➔ Networks trained on… successfully transfered to improve … “data” reconstruction
- ➔ DNN architectures can improve achievable energy resolution significantly
- ➔ Including timing information improves reconstruction

Thank you!

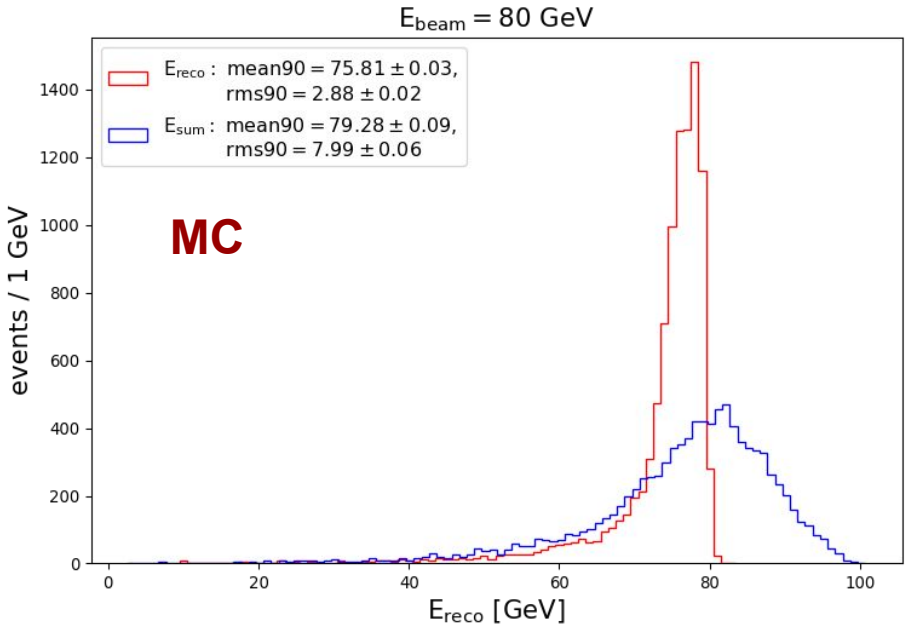


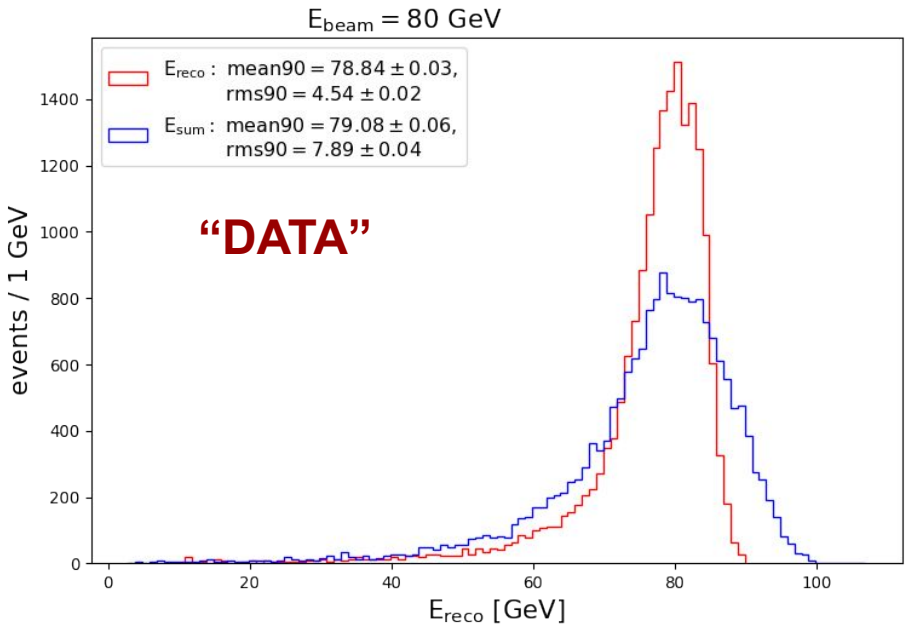
Bonus Slides

LC1+time: Learned timing function

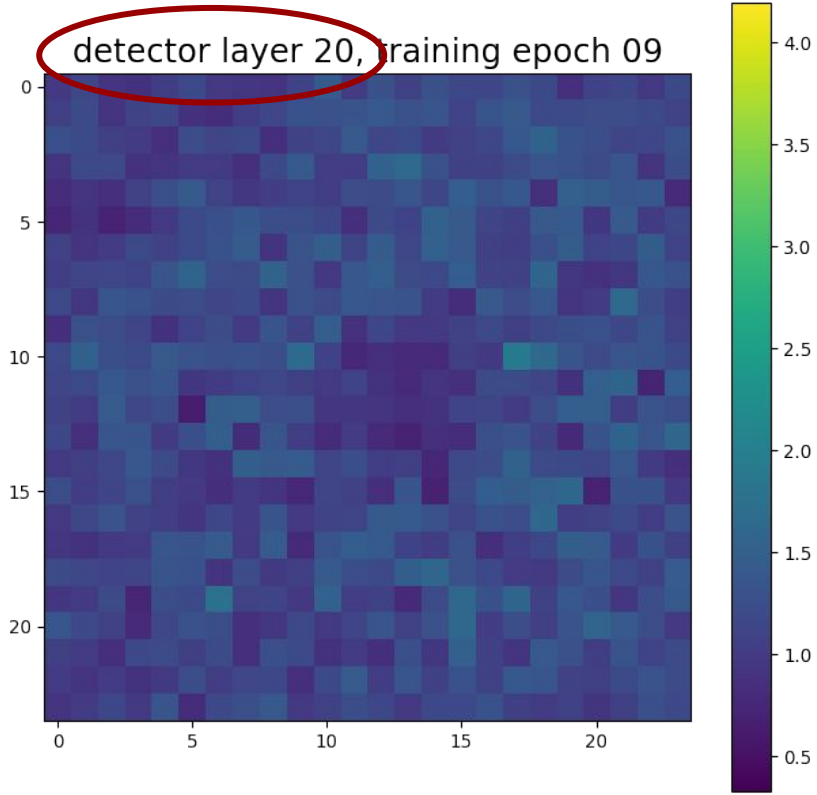


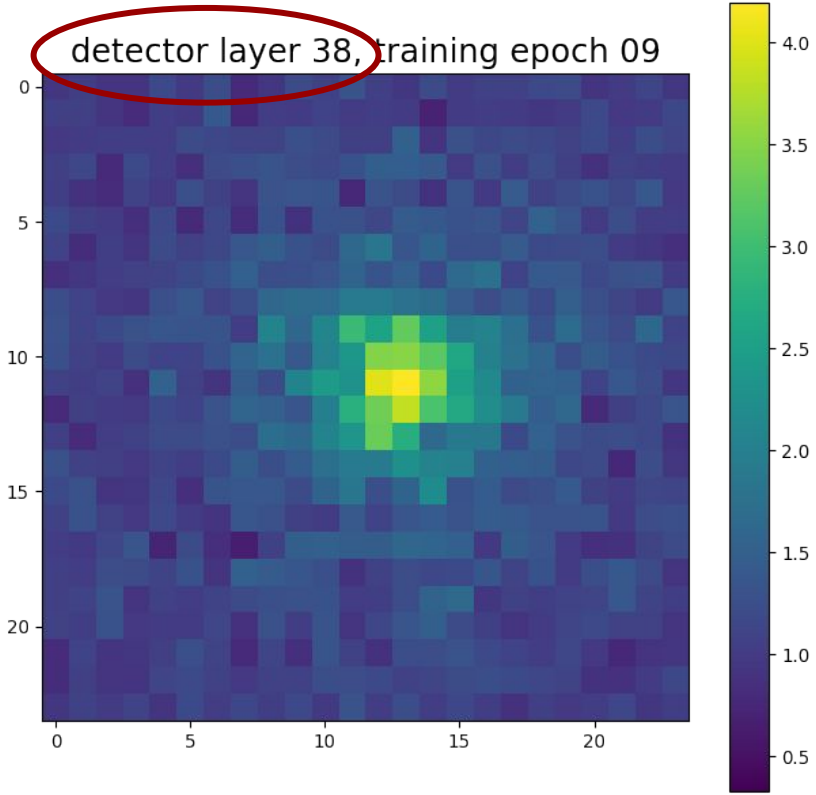
80 GeV: Trained on MC, Test on “Data”





Learned Weights





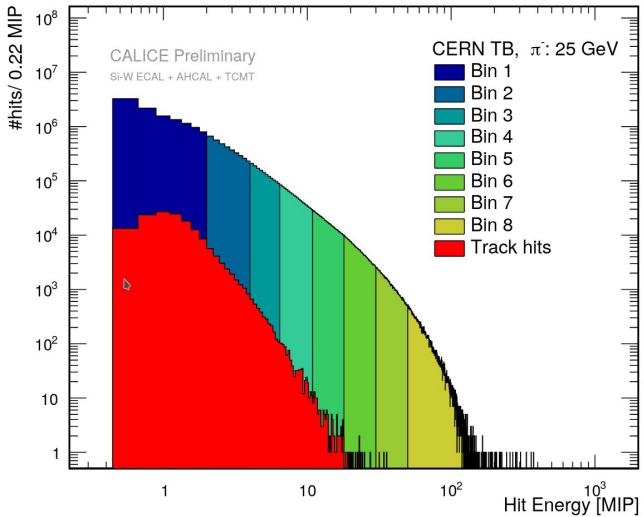
Local software compensation (CALICE)

Weight hits by local energy density

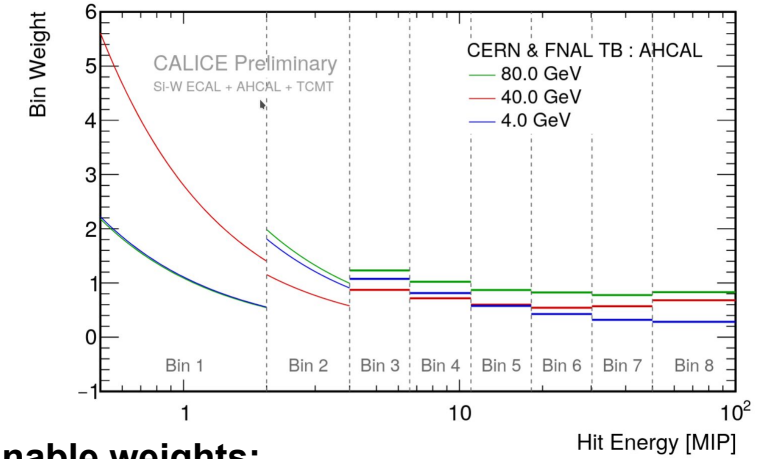
$$E_{reco} = \sum_i E_i w_i (E_i, E_{sum})$$

Only based on hit energy information, no usage of spatial and timing information

Shower hits in 8 bins:



Weights applied to each bin:



→ **24 trainable weights:**
8 bins & 2nd order polynomial parametrization per bin

arXiv:1803.05232

AHCAL Energy Calibration

$$E_i = f_{sat,i}^{-1}\left(\frac{(a_i - P_i - O_{i,m}) \cdot IC_i^{HGLG}}{\frac{G_i}{IC_1^{phy}}}\right) \cdot \frac{G_i}{C_{i,p}^{MIP}}$$

a_i : raw signal in [ADC]

P_i : pedestal subtraction

$O_{i,m}$: pedestal memory cell offset

$C_{i,p}^{MIP}$: MIP constant

G_i : SiPM gain

IC_i^{HGLG} : high-gain / low-gain intercalibration

IC_i^{phy} : physics / calibration mode intercalibration

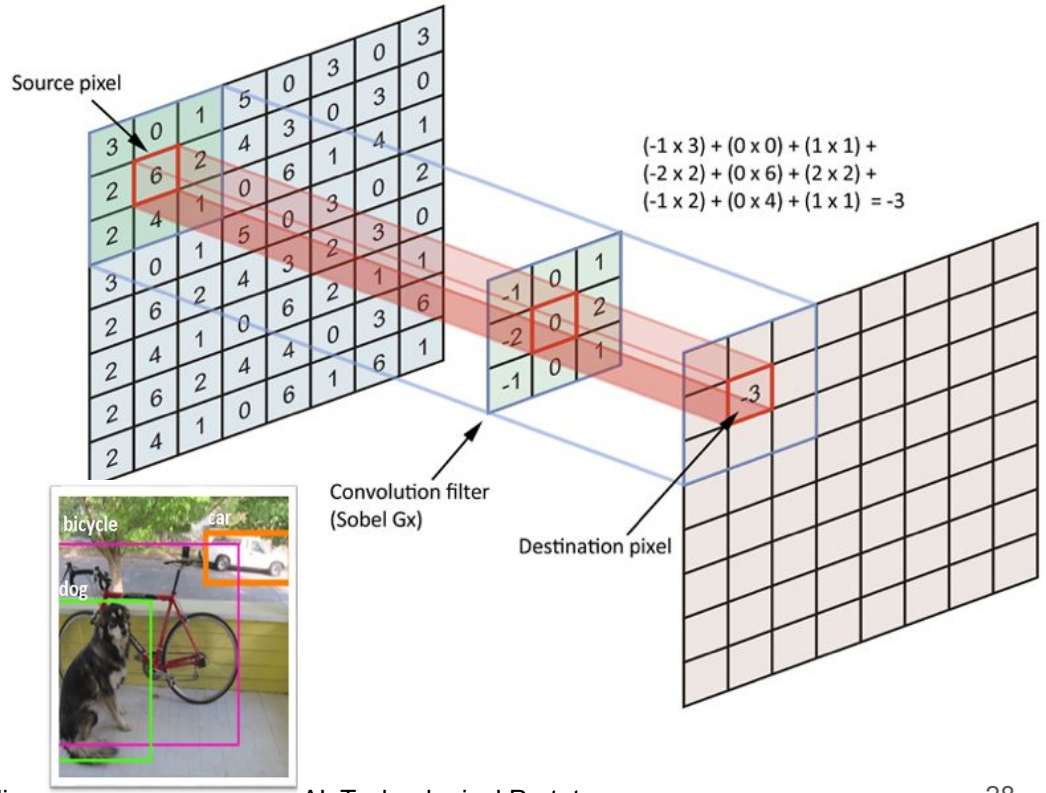
f_{sat} : SiPM saturation function

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

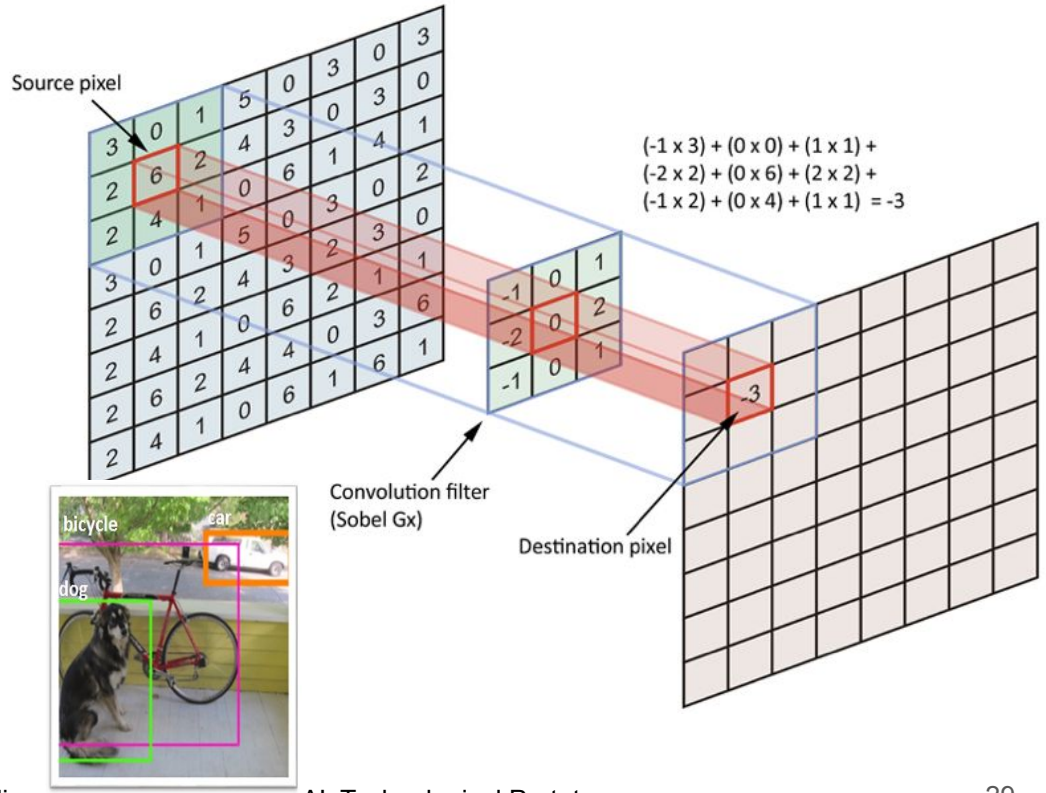


Locally Connected Layer

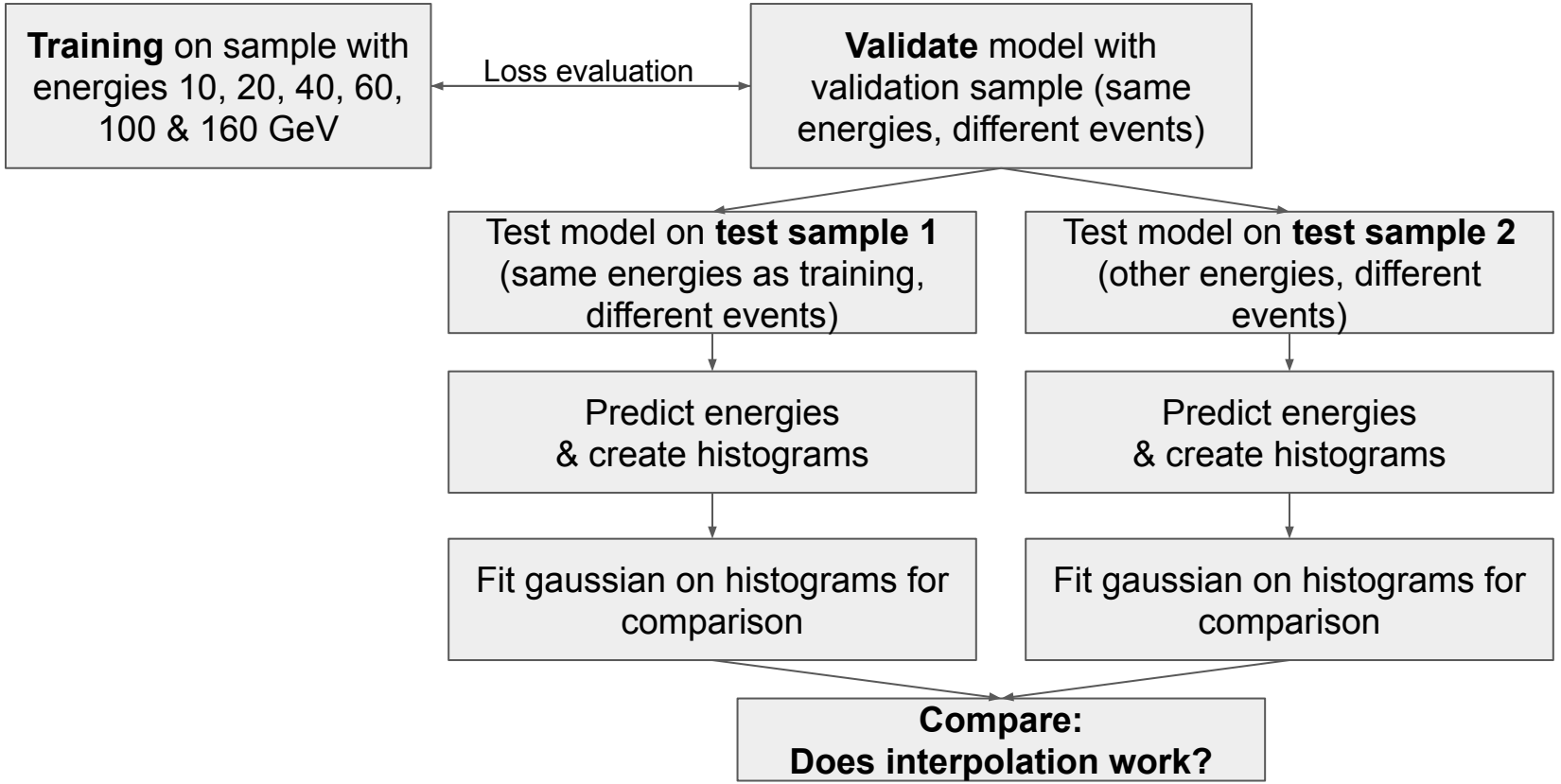
- Output value of convolutional kernel:

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

- **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



Training & Testing Process



5D calorimetry

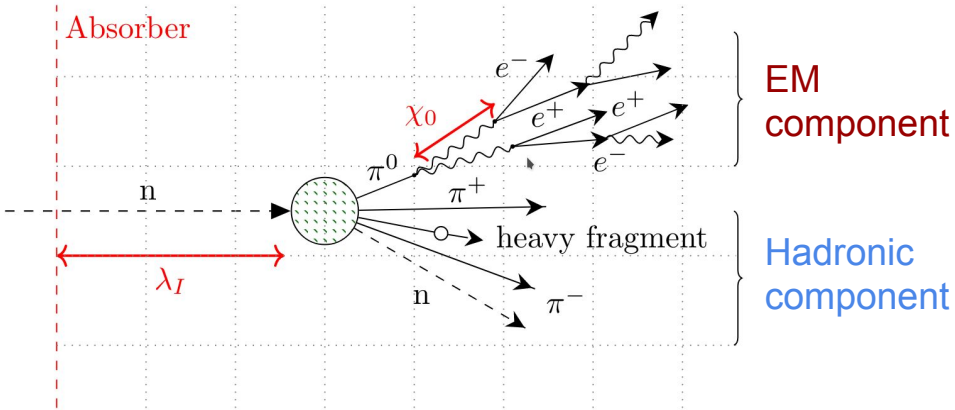
5D = 3 spatial dimensions + hit energy + hit timing

Hit energy: high for EM hits, low for hadronic

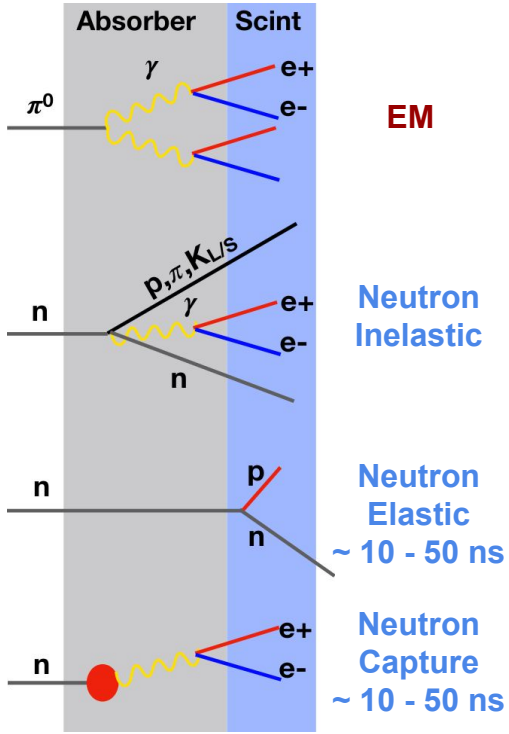
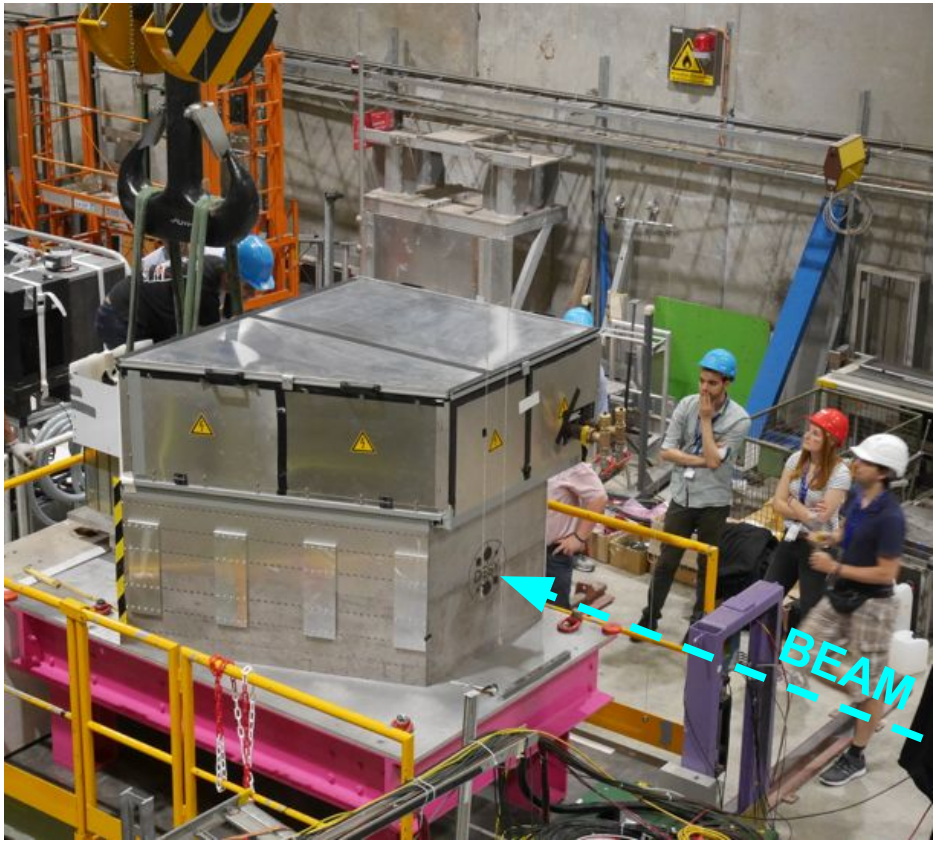
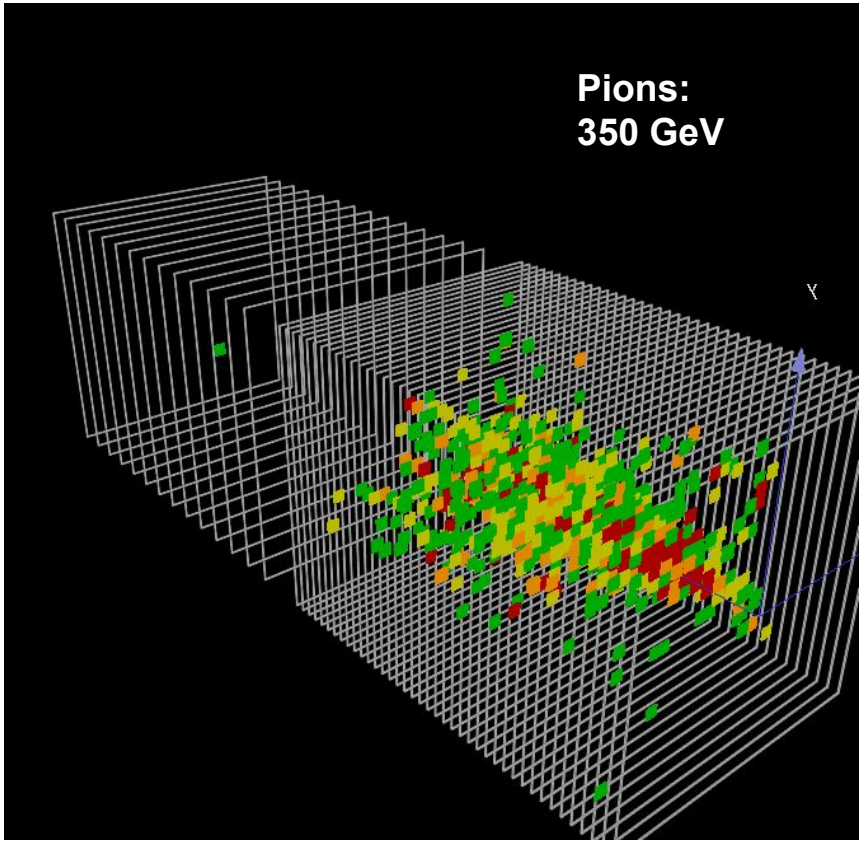
Hit timing: early for EM hits, late for hadronic

→ improved energy reconstruction (“*software compensation*”)

Hadr. shower:



Timing of EM and hadr. processes:



Test Beam Setup: Technological Prototype @ SPS