END-TO-END RECONSTRUCTION ALGORITHM FOR HIGHLY GRANULAR CALORIMETERS

Philipp Zehetner[1,2] - Jan Kieseler[1,3] - Shah Rukh Qasim[4] - Dolores Garcia[1] - Kenneth Long[1]

[1]CERN, [2]University of Munich, [3]Karlsruhe Institute of Technology, [4]University of Zurich

Work done in context of ML4Reco

MOTIVATION

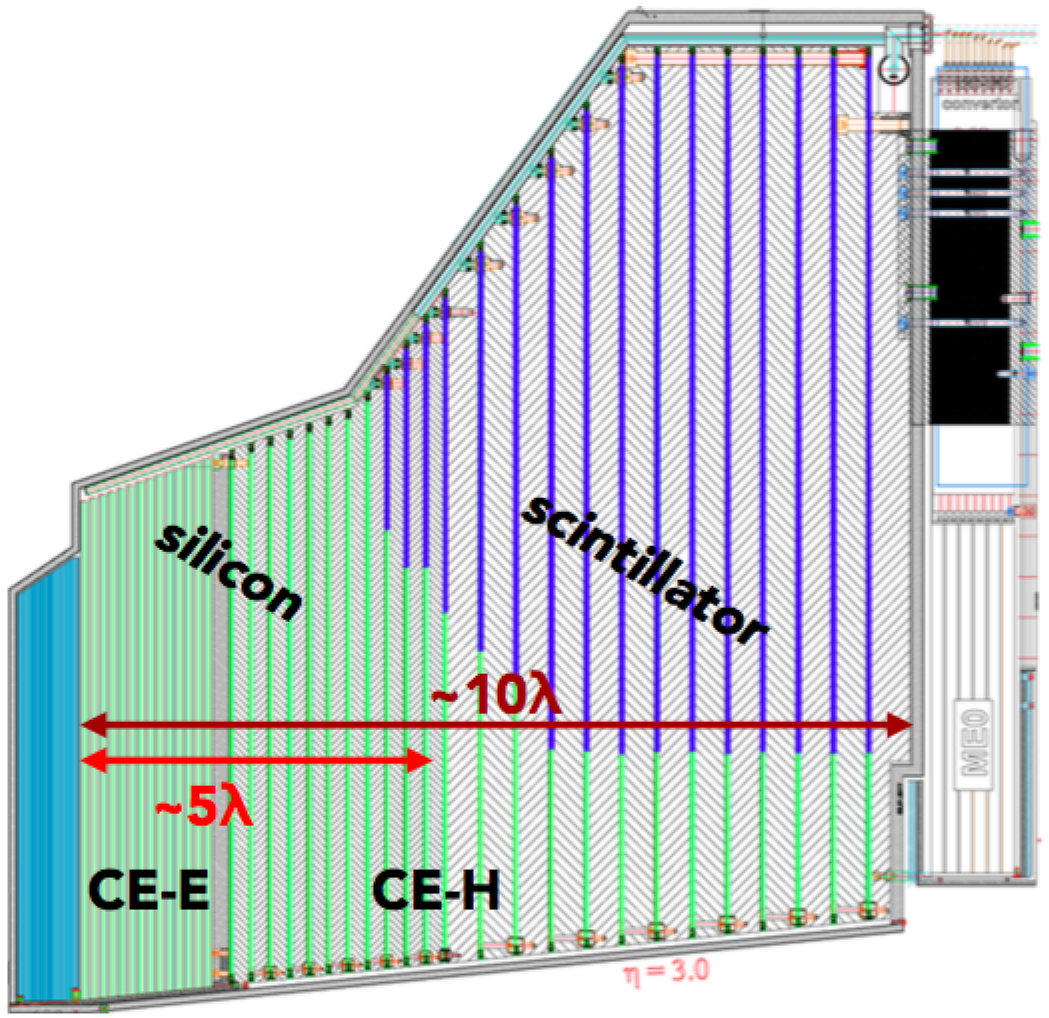
Single step reconstruction of complex events can optimally use all available information

HIGH-LUMINOSITY-LHC

- CMS will observe 200 pile-up (PU) events
- Upgrade for end-cap calorimeter
  High **G**ranularity **CAL**orimeter (HGCAL)
- HGCAL: 2x ~3M readout channels
- Silicon sensors and scintillators

RECONSTRUCTION GOALS

- Combine calorimeter hits with tracks
- **Cluster** hits to build showers
- **Regress** energy of showers
- **Classify** particle ID
- All of this using one **differentiable** network



https://cds.cern.ch/record/2293646/

CMS - HGCAL

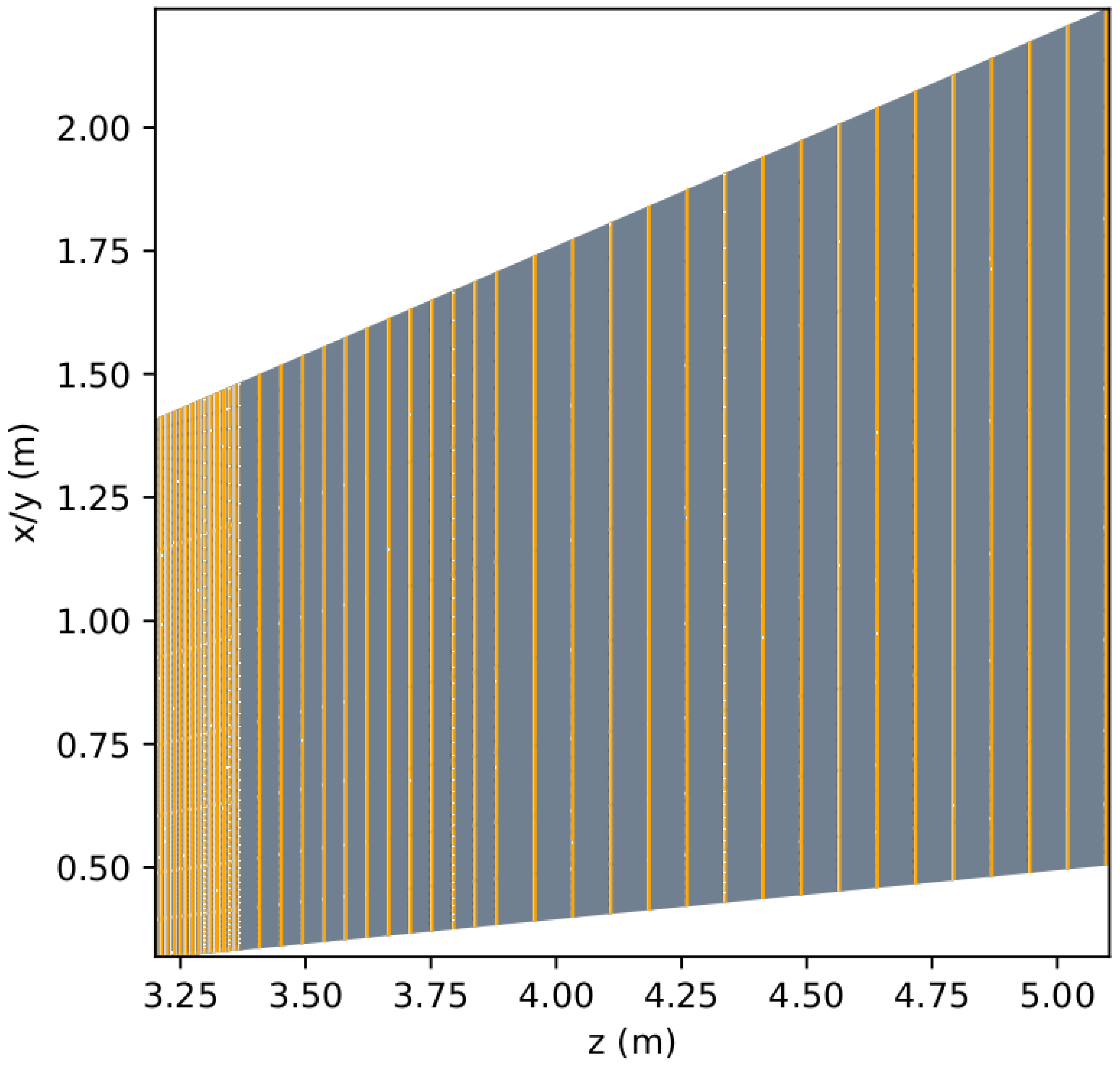
METHODS

- Graph-based neural network
  implemented in TensorFlow
- Object Condensation Loss
  allowing points to represent objects
- GravNet Architecture
  A dynamic GNN that operates on point clouds

PREVIOUS WORK

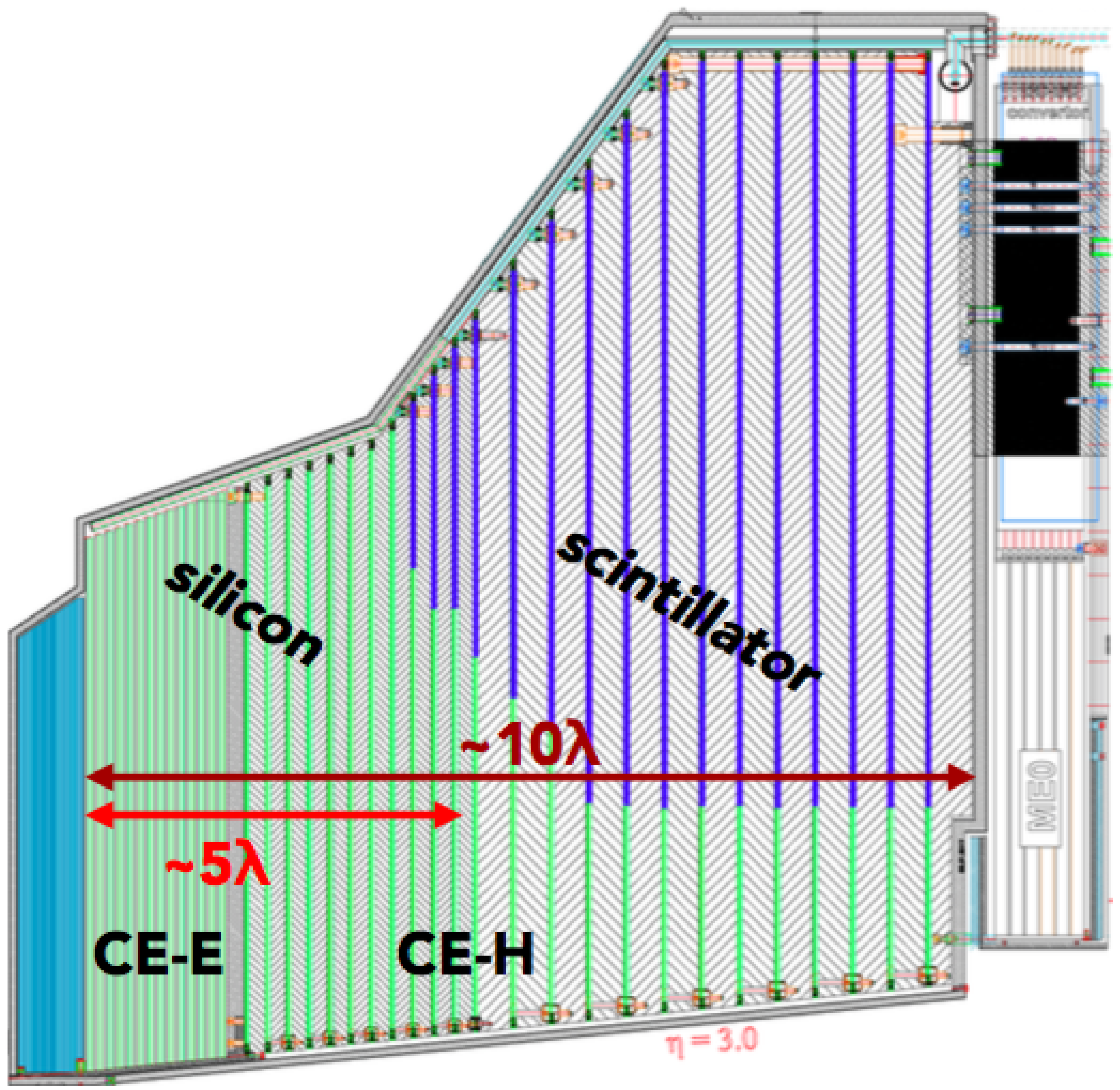
- Connecting The Dots 2023
- ACAT 2022
- ACAT 2021
- End-to-End Reconstruction
- GravNet
- Object Condensation

TOY DETECTOR



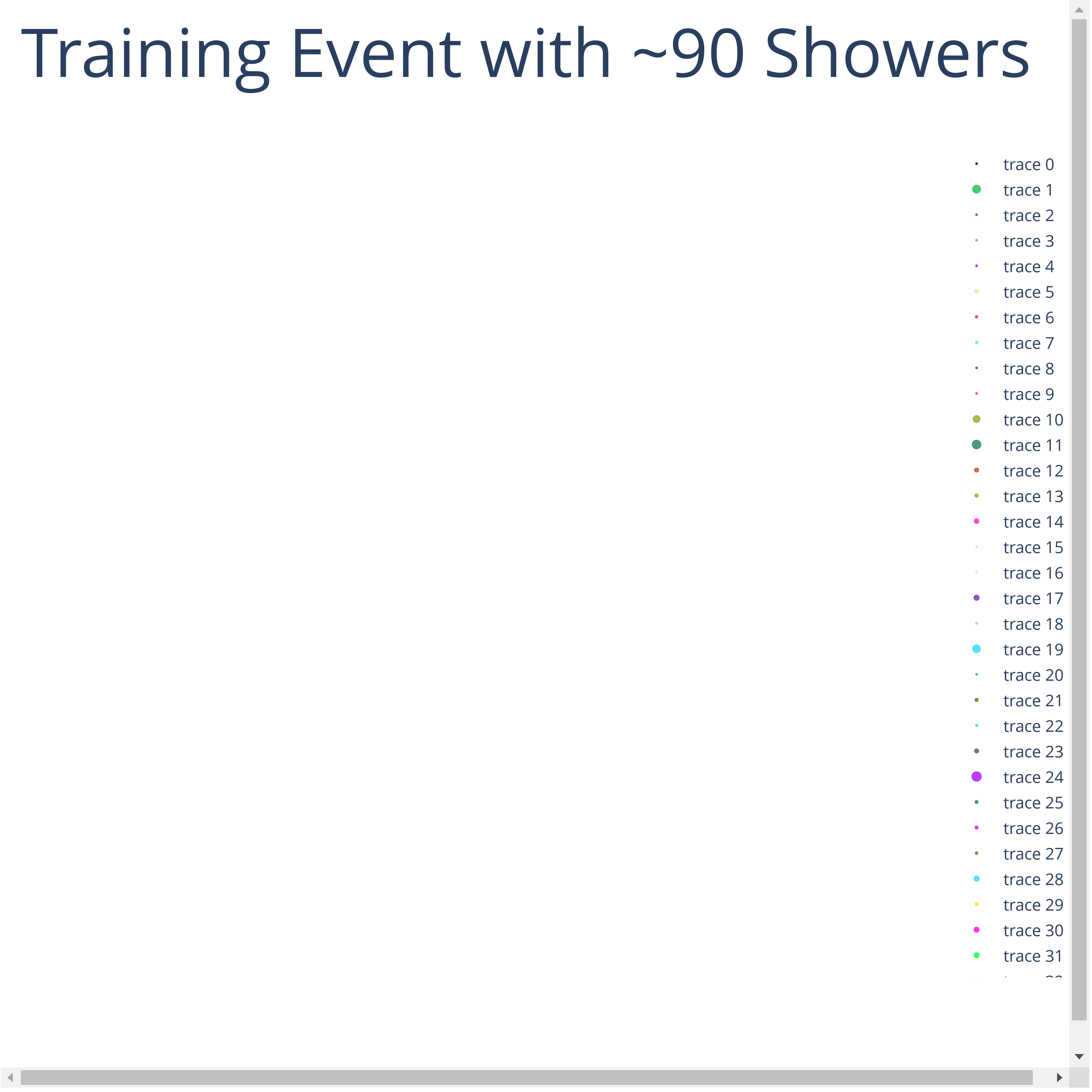
- ~50 Sampling layers with 200 μm silicon sensors as active material
- Sensors square in η and φ
- ~3 Million readout channels per end-cap
- Standalone simulation using Geant

HGCAL



- ~50 Sampling layers with 120 μm, 200 μm, 300 μm silicon and scintillators
- Heaxgonal silicon sensors
- ~3 Million readout channels per end-cap
- Simulation within CMS Software (CMSSW)

TRAINING EVENTS



DATA SETS

- **Particle Showers**
  - Electrons, photons, charged pions, or kaons (K-long)
  - 0.1 GeV < E < 200 GeV
  - $1.5 < \eta < 3.0$

- **Minimum bias**
  - Proton-proton collisions at 13 TeV
  - Simulated with PYTHIA8
  - Used for pile-up

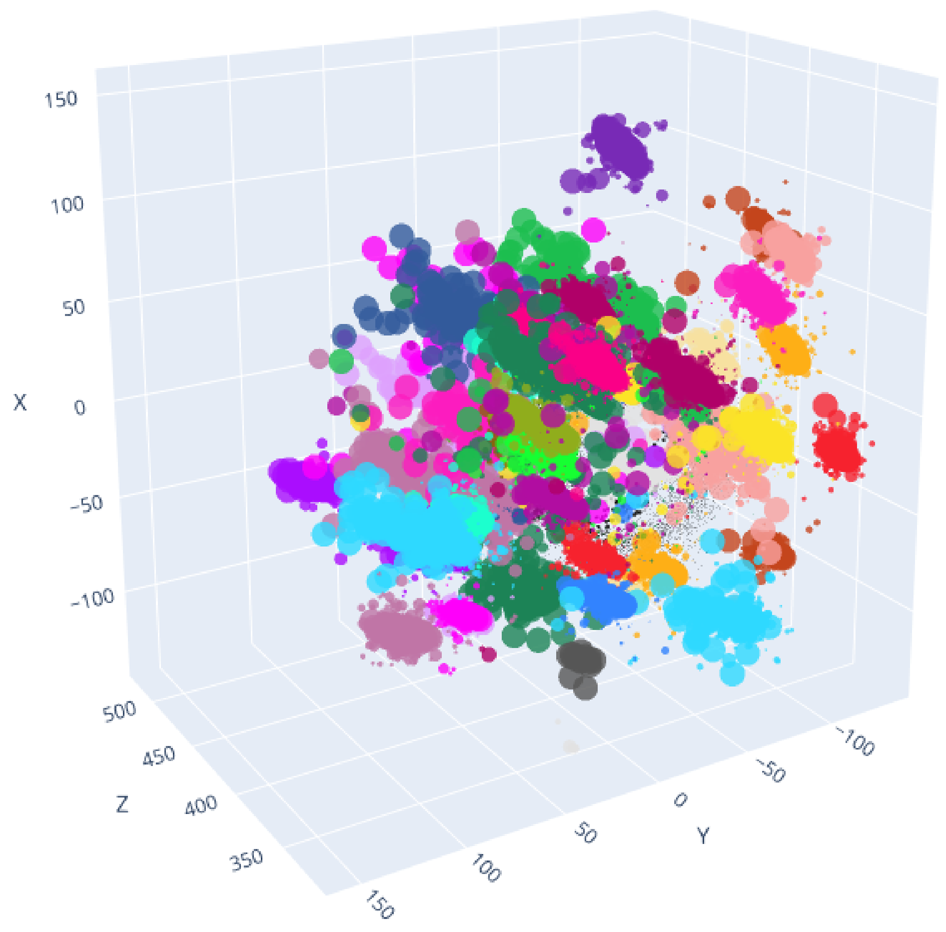
- **Tracks**
  - Tracks are added for all charged particles
  - Tracks are flagged as such and have the particles original smeared out energy

- **Train Data**
  - Multiple showers + Gaussian noise
  - Multiple showers + Gaussian noise + 200 PU in random 30° φ region

- **Test Data**
  - Multiple showers + Gaussian noise
  - Single shower + Gaussian noise + 200 PU in full detector

OVERVIEW

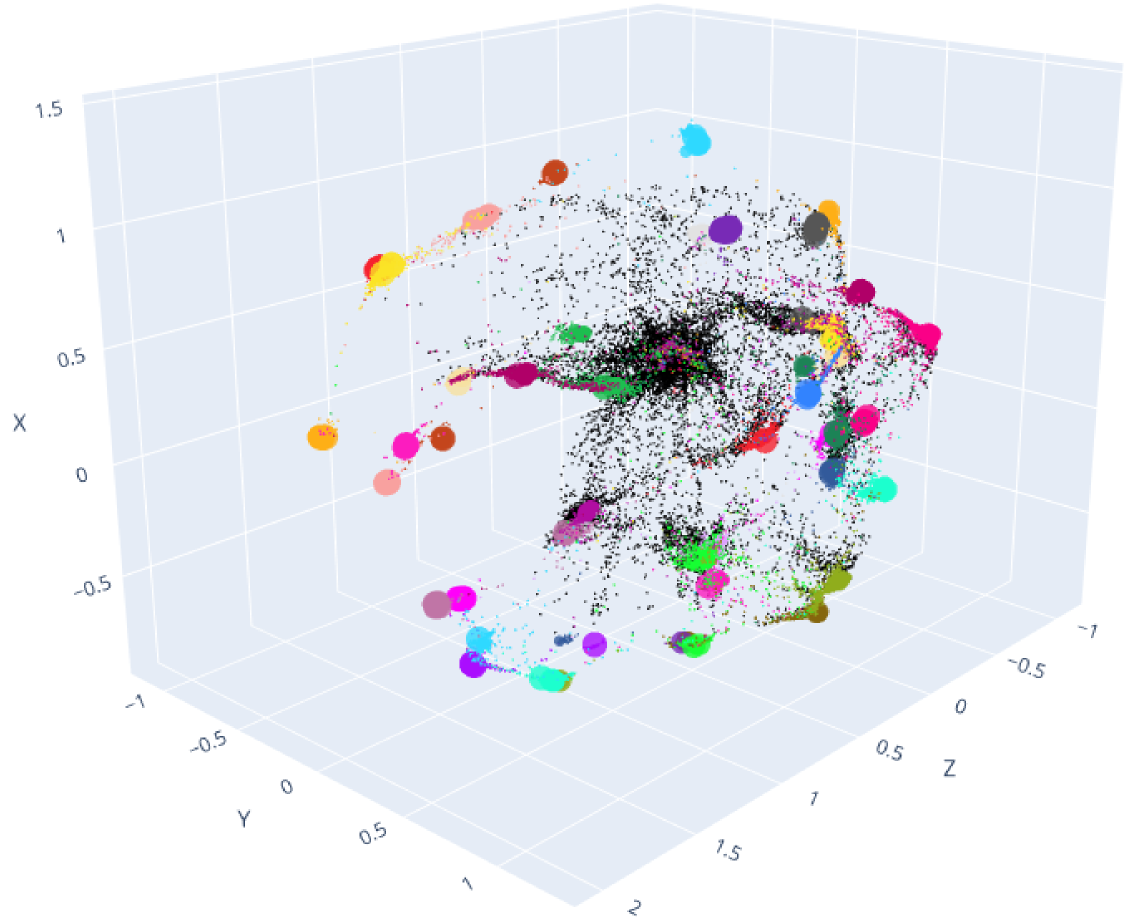
DETECTOR SPACE



CONCEPT

- Colour represents different showers
- Overlapping showers make clustering in detector space difficult
- Learn mapping into clustering space
- Learn confidence β
- β close to 1 $\rightarrow$ hit can represent shower
- In clustering space hits from the same shower should be close
- Every shower should have at least one hit with high β (condensation point)

CLUSTER SPACE



OBJECT CONDENSATION

Highest confidence β for every shower gets assigned as condensation point, indicated by α

Charge $$q_\alpha = \tanh^2(\beta_\alpha)$$

Attractive Potential $$V_{\text{att}} \propto q_\alpha \cdot |x - x_\alpha|^2$$

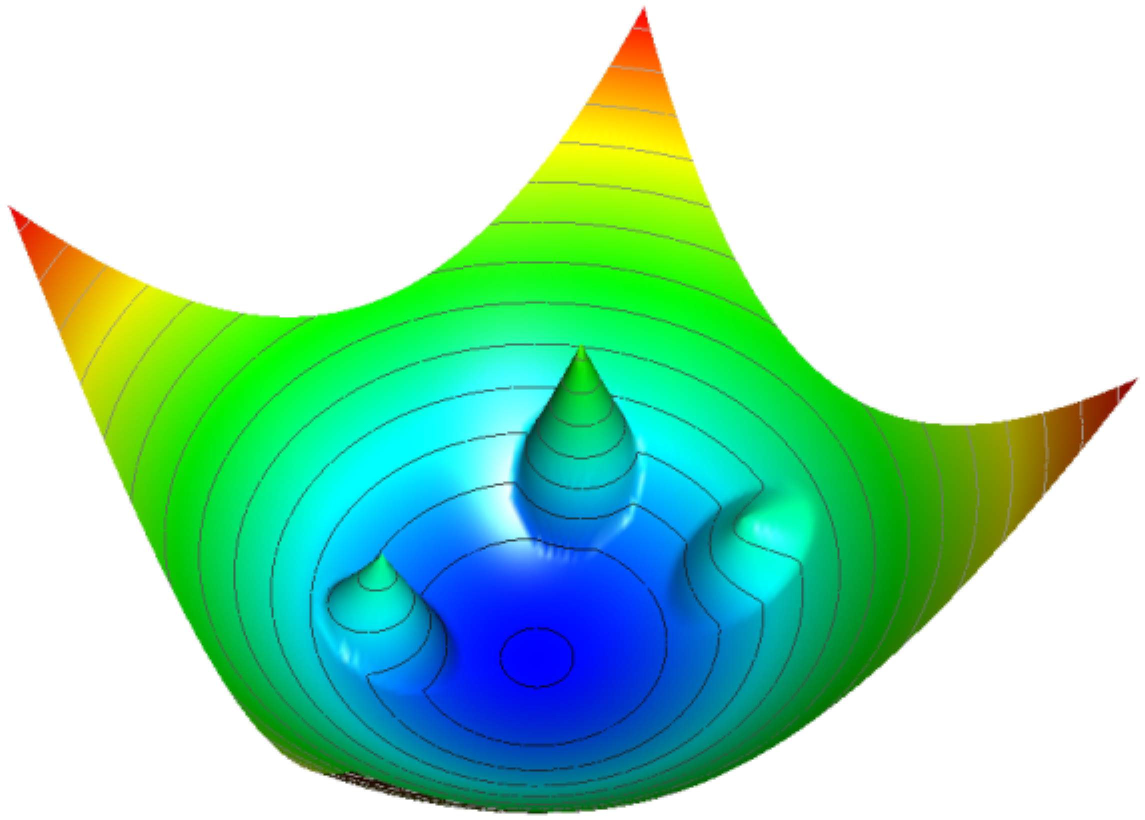
Repulsive Potential $$V_{\text{rep}} \propto q_\alpha \cdot \max(0, 1 - |x - x_\alpha|)$$

Shower Matrix $$M_{jk} = \begin{cases} 1 & \text{hit } j \in \text{ shower } k \\ 0 & \text{else} \end{cases}$$

Loss terms

$$L_V = \frac{1}{N} \sum_{j=1}^{N} q_j \sum_{k=1}^{K} \left(M_{jk} V_{\text{att}}(x_j) + (1 - M_{jk}) V_{\text{rep}}(x_j) \right)$$

$$L_\beta = \frac{1}{K} \sum_{k=1}^{K} \left(1 - \beta_\alpha^k\right) + \frac{1}{N_{\text{noise}}} \sum_{j \in N_{\text{noise}}} \beta_j$$

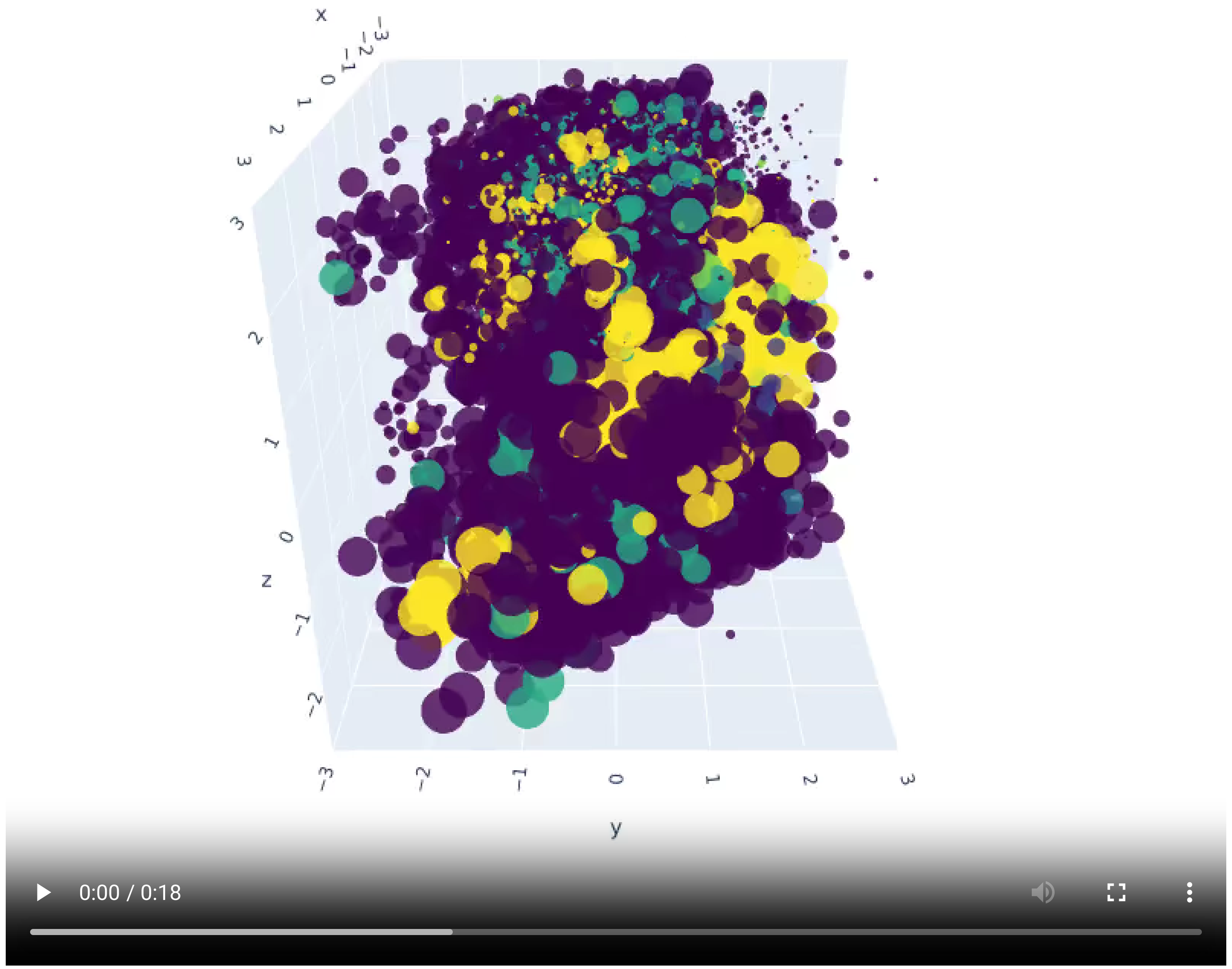


Object Condensation

Potential in cluster space seen by a single hit

- Minimum: Matching condensation point
- Local peaks: Condensation points from 3 other showers

OBJECT CONDENSATION IN TRAINING



CREATING SHOWERS

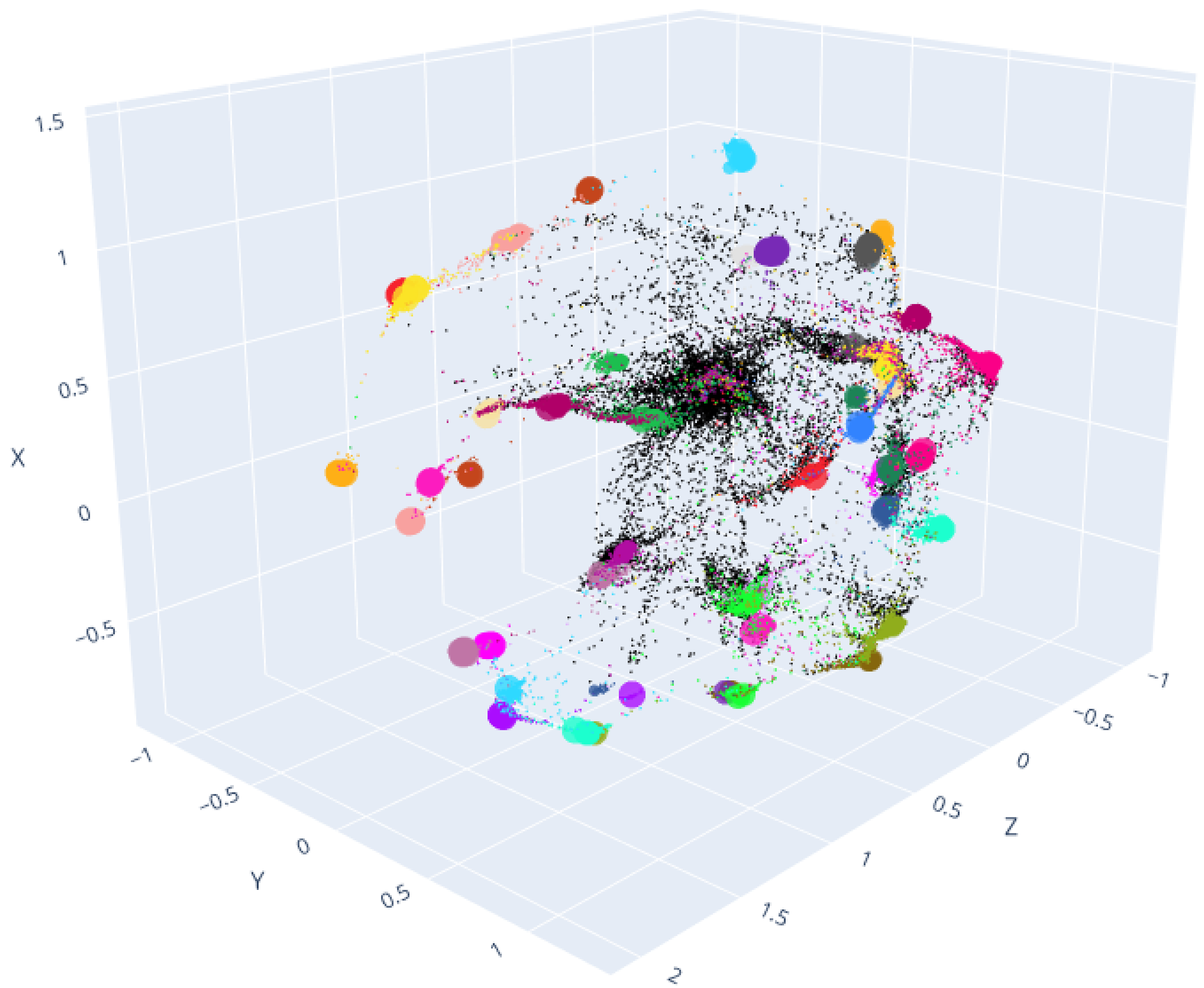
FAST CLUSTERING ALGORITHM

1. Sort hits by confidence β
2. Highest β is first condensation point
3. Hits within distance threshold t_d = 0.25 around β are assigned to first shower
4. Remove already assigned hits from list
5. Repeat steps 2 - 4 as long as highest β value is larger then threshold t_β =0.3
6. Remaining hits are classified as noise

ALTERNATIVE:

More sophisitacted clustering algorithms such as HDBSCAN

CLUSTER SPACE



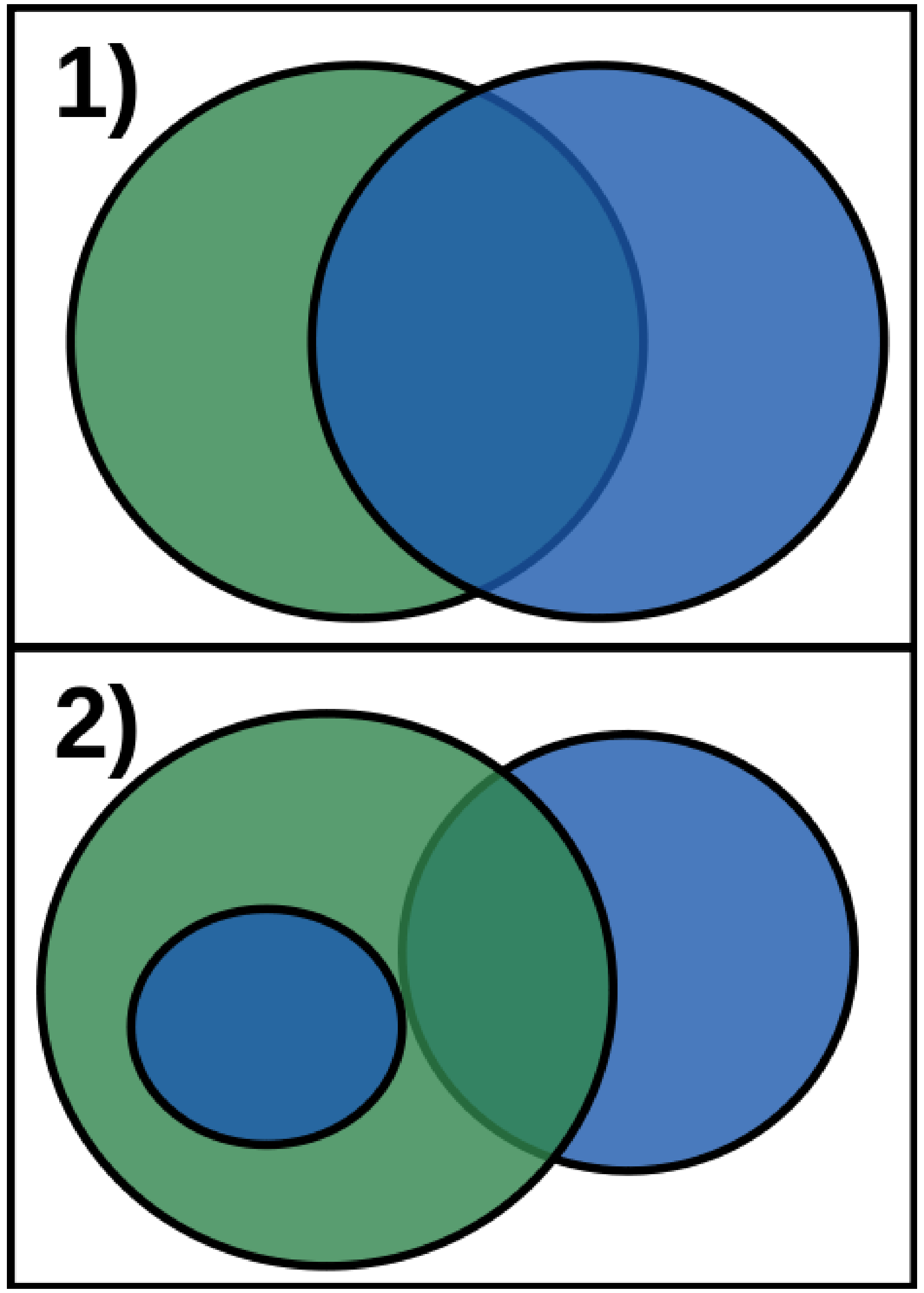
MATCHING SHOWERS

MATCHING CONDITIONS

To evaluate the performance of the algorithm, reconstructed showers are matched with truth showers.

- Reconstructed showers are matched with true showers based on their energy weighted overlap.
- More precisely: The intersection over union between two showers has to be larger than 33%
- If truth shower and reconstructed shower have equal energy, this translates that at least 50% of each shower overlaps

Important:
The matching conditions influence the performance metrics, but do not change the performance of the algorithm.
A low threshold allows to find a match for nearly every shower but comes at the cost of degraded energy resolution and vice versa.



1) True shower and predicted shower overlap
2) More complicated matching scenario

DATASETS

to evaluate

CLUSTERING & ENERGY

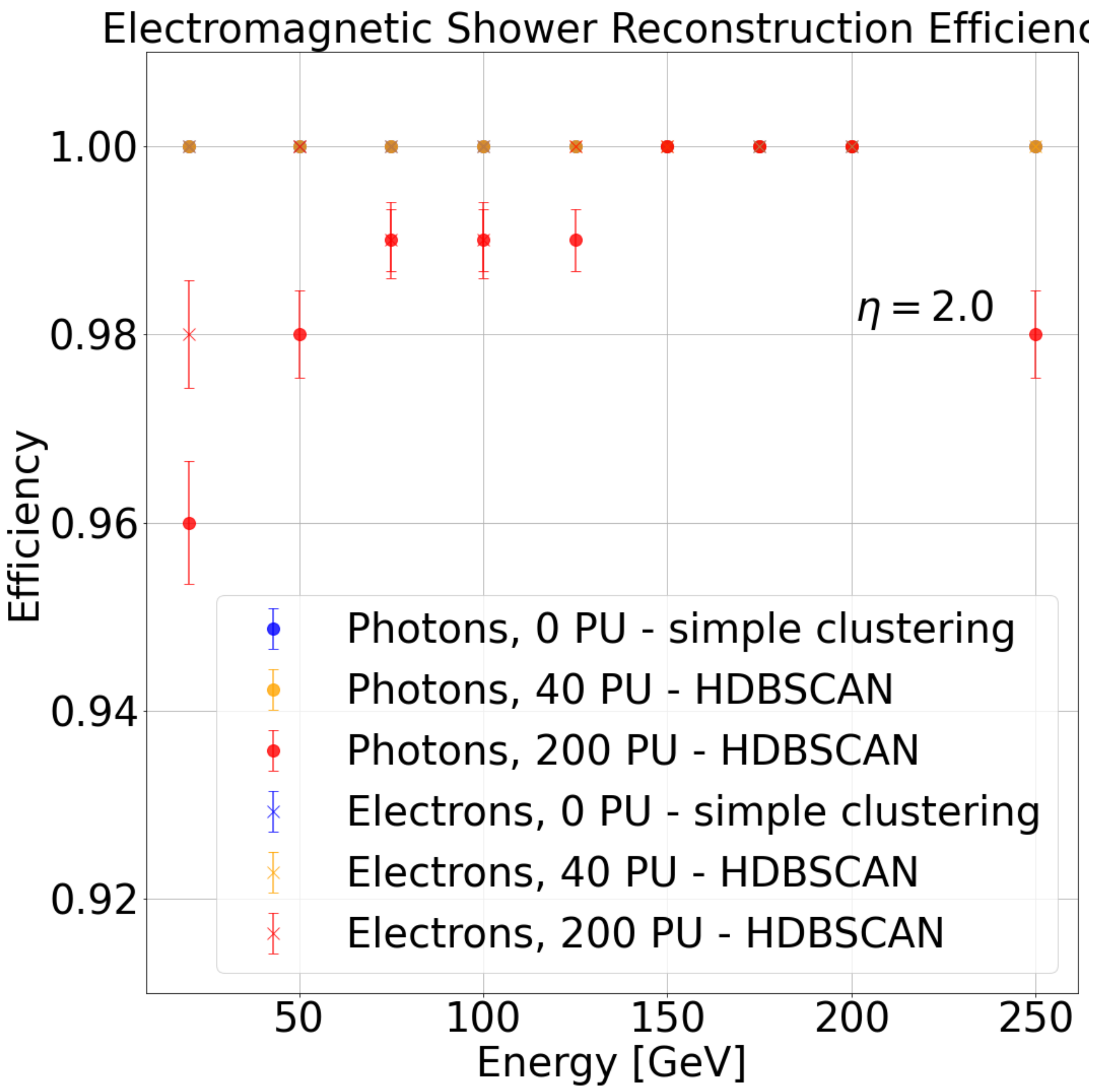
- Single shower
  - Electrons, photons, charged pions, or kaons (K-long)
  - E = 20 GeV, 50 GeV, 100 GeV, 200 GeV
  - $\eta = 2.0$
- Random Gaussian noise
- 200 minimum bias proton-proton collisions

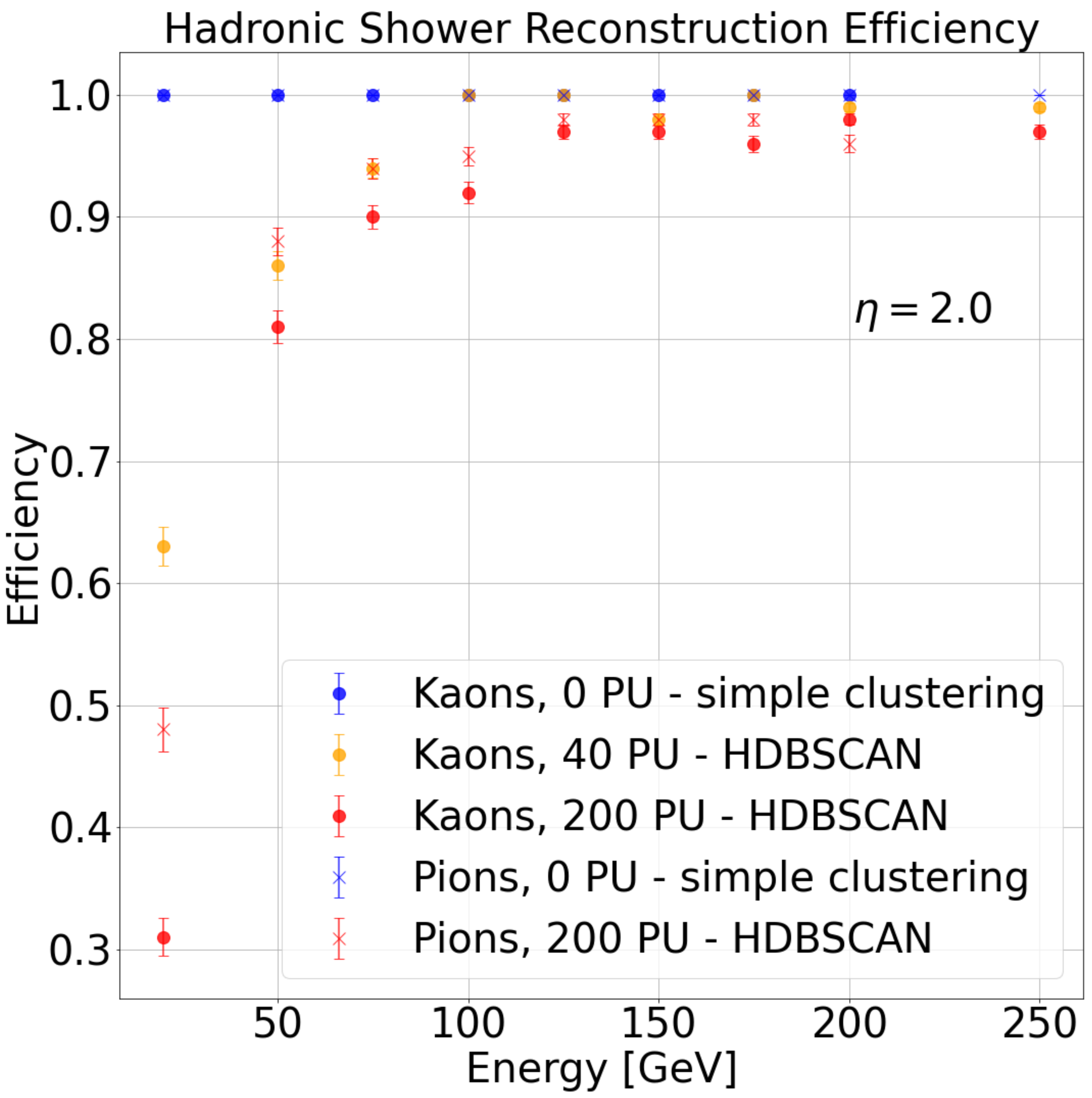
to evaluate

PARTICLE IDENTIFICATION

- 60-90 showers
  - Electrons, photons, charged pions, kaons (K-long)
  - $0.1\ \text{GeV} \leq E \leq 200\ \text{GeV}$
  - $1.5 \leq \eta \leq 3.0$
- Random Gaussian noise
- 0 pile-up

EFFICIENCY





- Efficiency for electromagnetic showers close to perfect even in 200 PU
- Hadronic shower efficiency approaches 100% for higher energies

RESPONSE AND RESOLUTION

Metrics for matched showers

RESPONSE

Mean of predicted energy over true energy

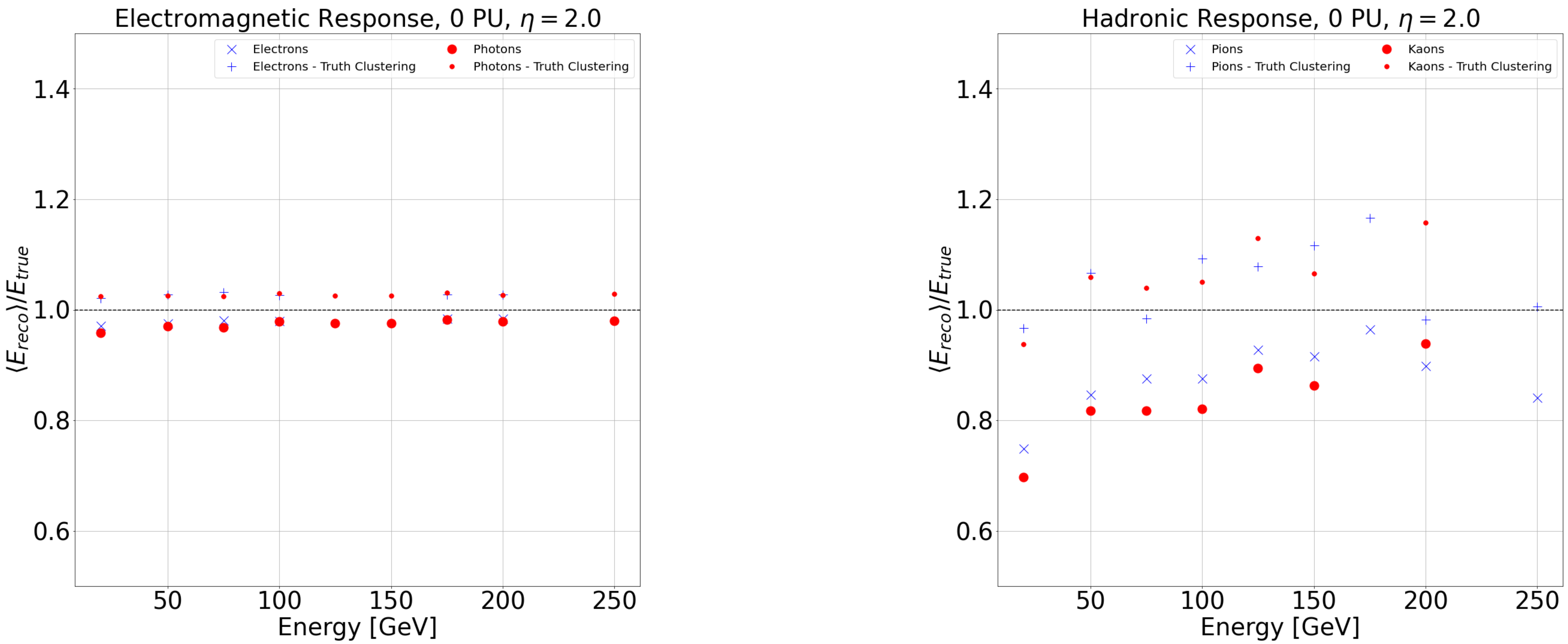
RESOLUTION

Standard deviation of predicted energy over mean of predicted energy

Baseline: Ideal Clustering

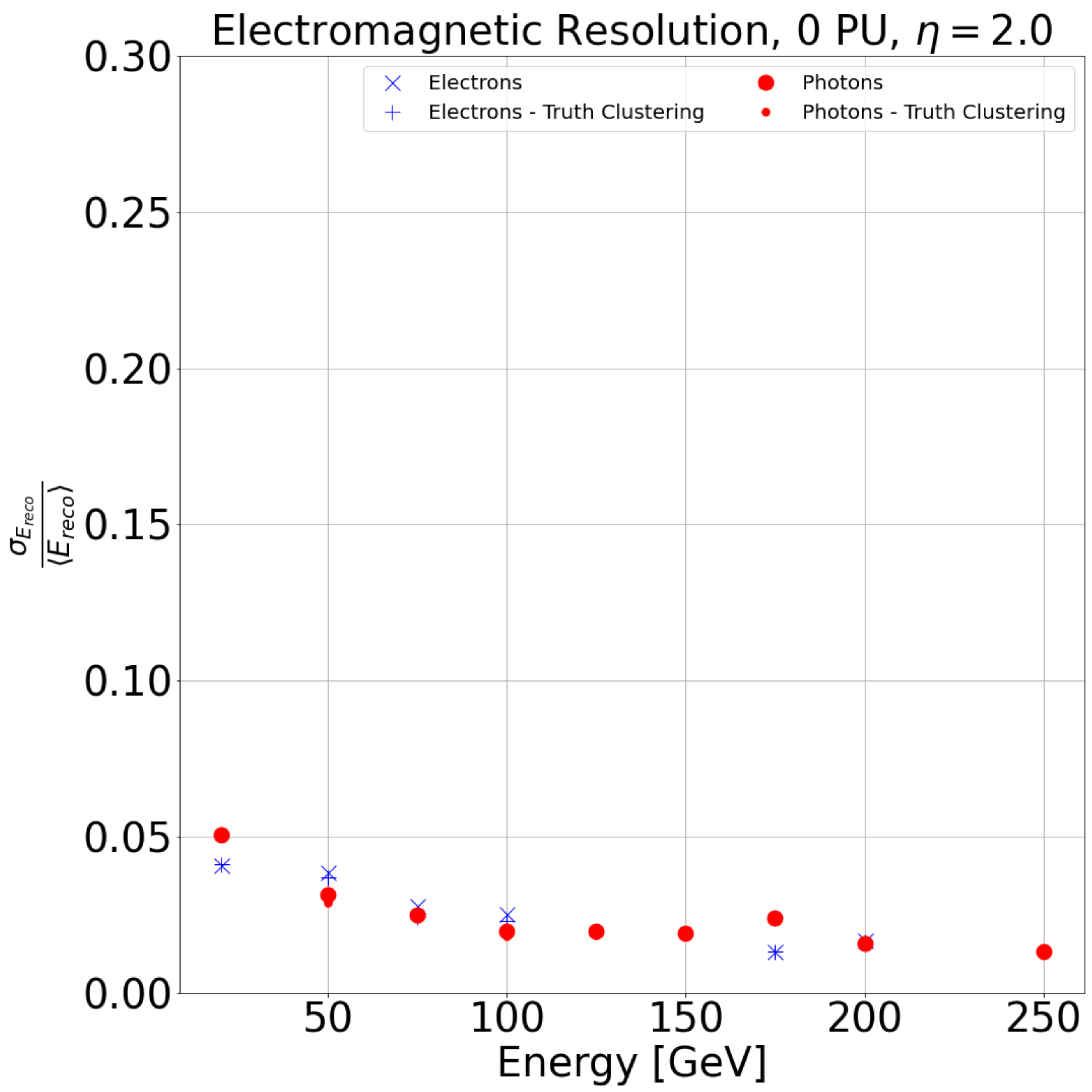
- Use truth information for clustering
- Energy is sum of all hit energies belonging to shower
- Pile-up may contaminate truth information for overlapping hits

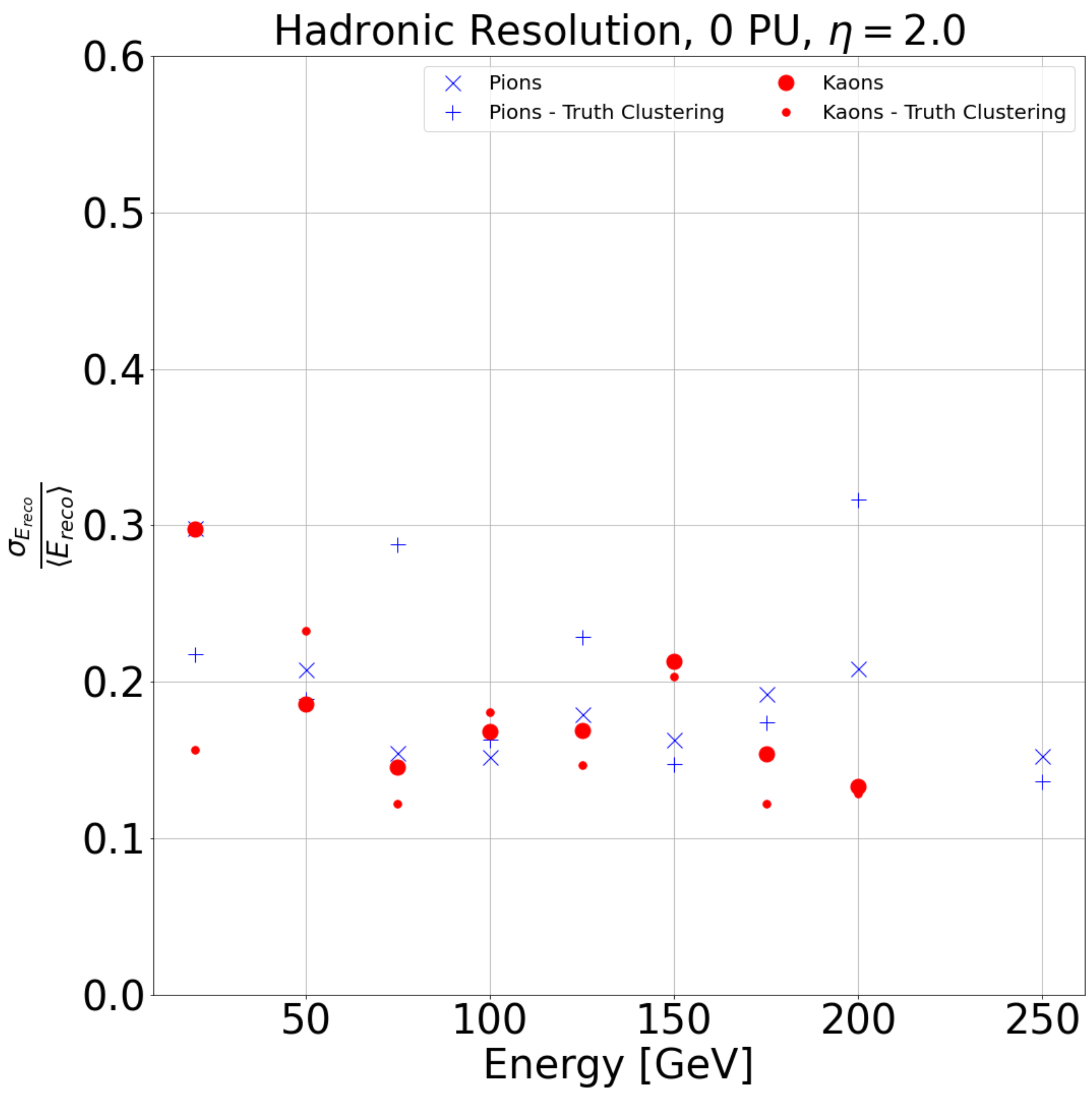
RESPONSE - 0 PU



- Electromagnetic and hadronic response as close to linear as baseline

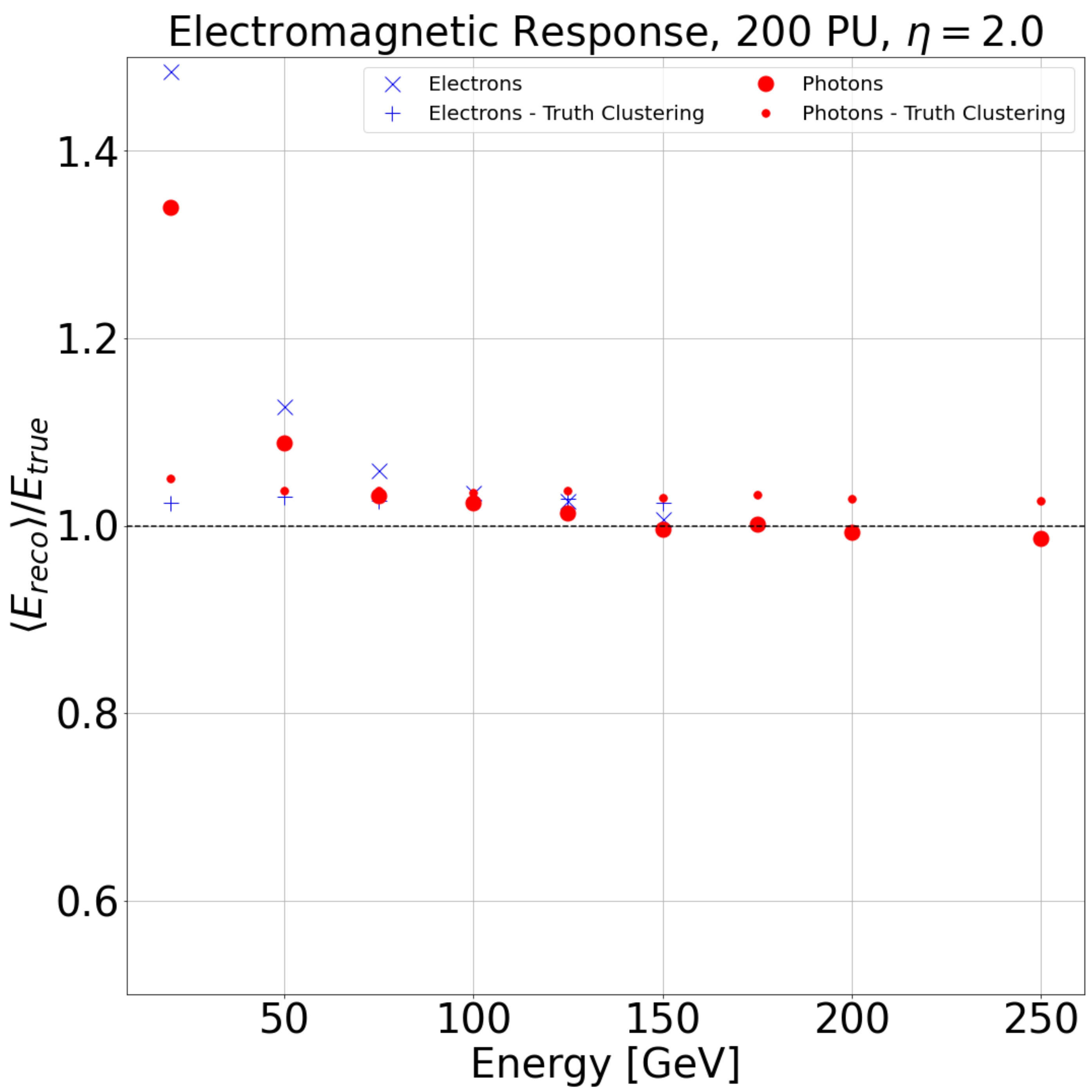
RESOLUTION - 0 PU

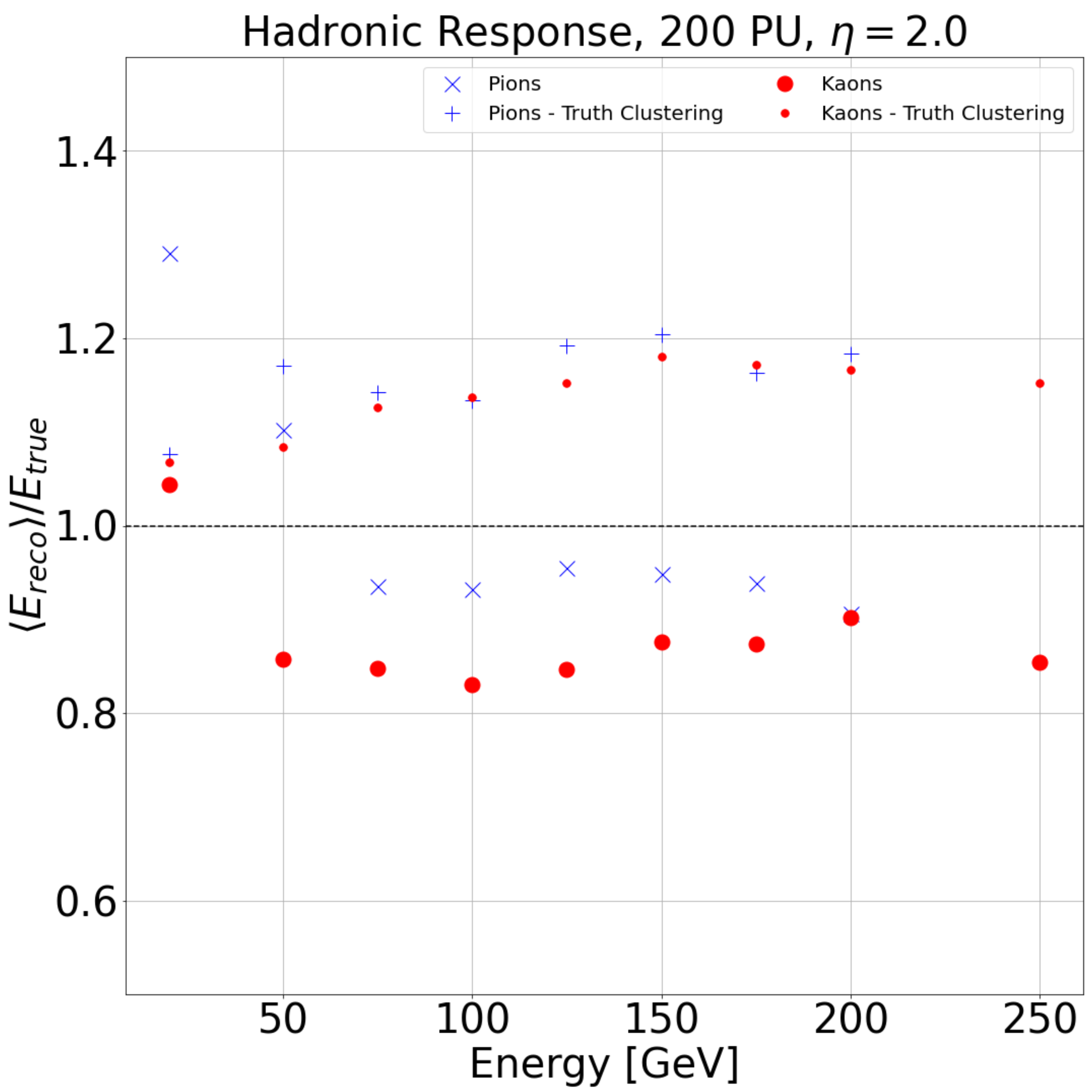




- EM: Reconstructed showers identical to baseline
- Limited by detector resolution

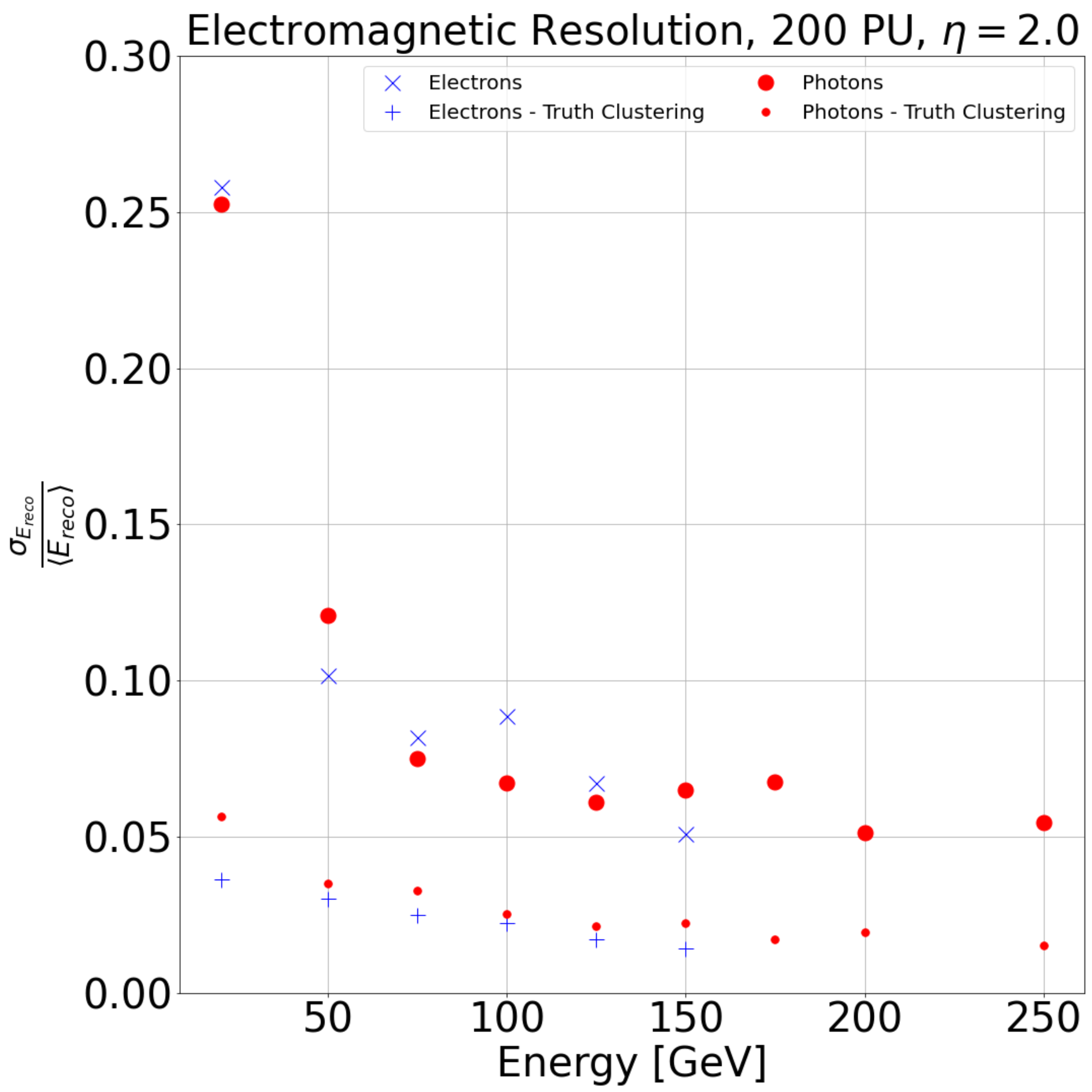
RESPONSE - 200PU

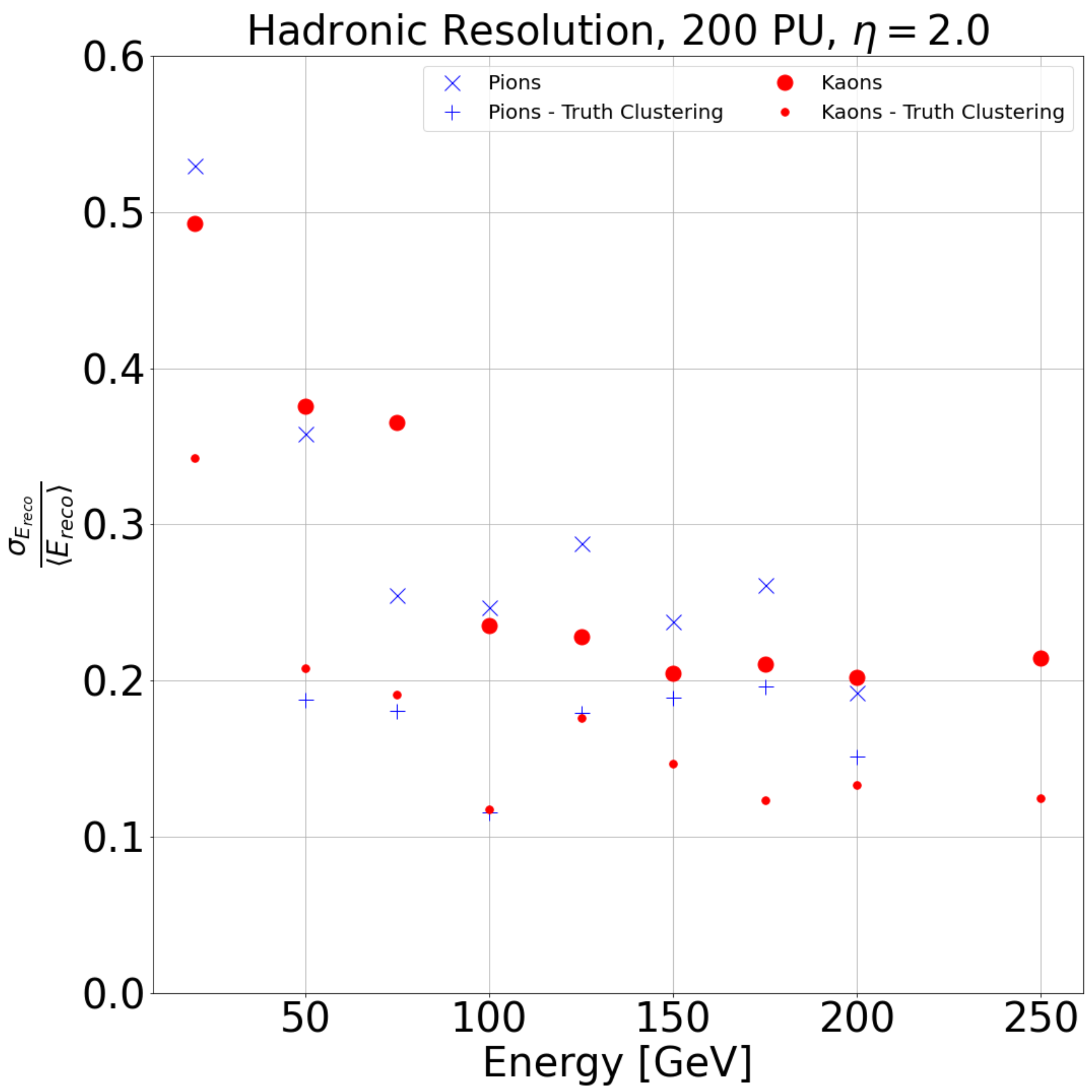




- Contamination from PU more noticable in 200 PU for energies belwo $50\ \mathrm{GeV}$ ($= 13\ \mathrm{GeV}\, p_{\mathrm{T}}$)
- Response close to constant for higher energies

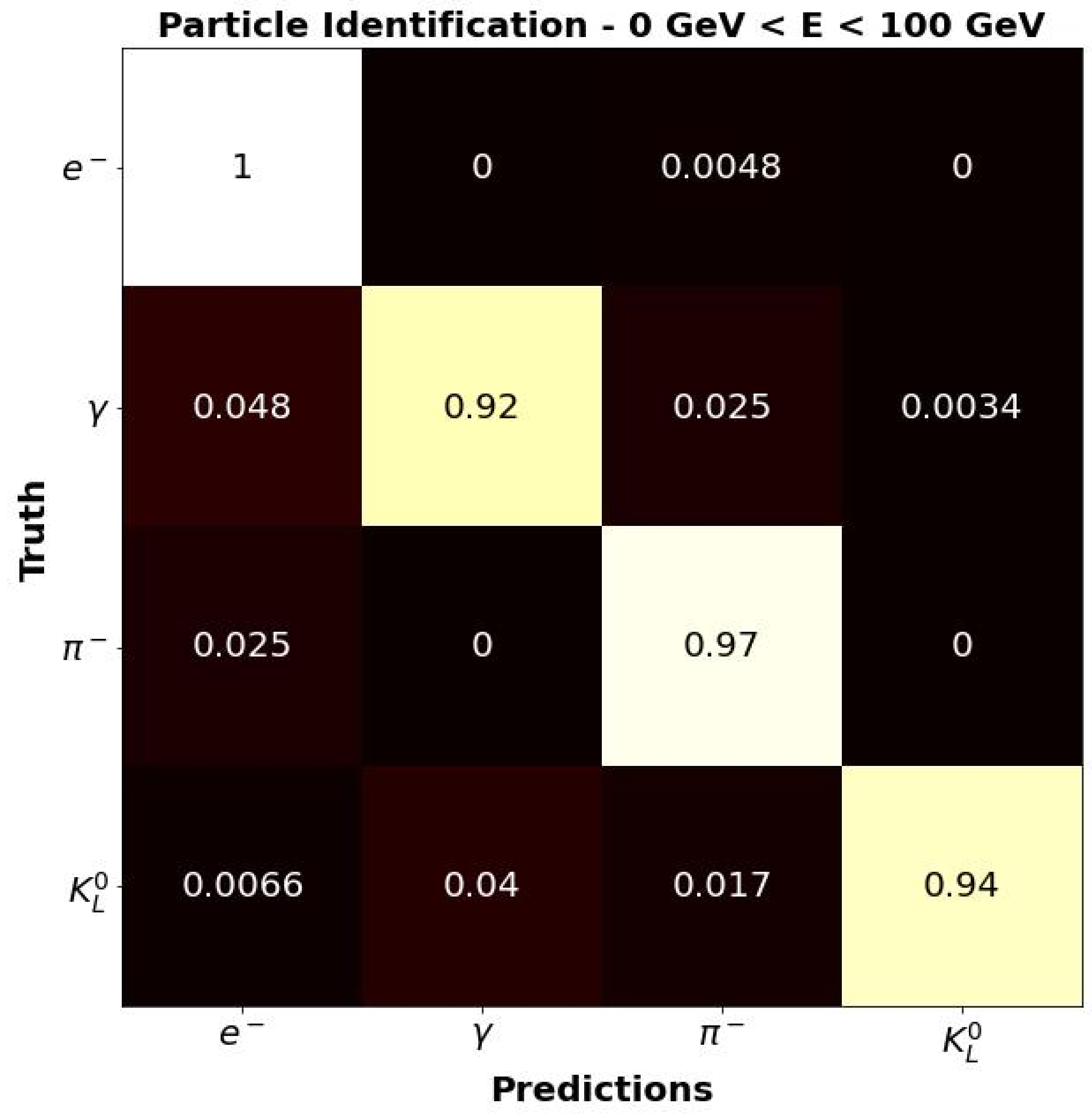
RESOLUTION - 200PU

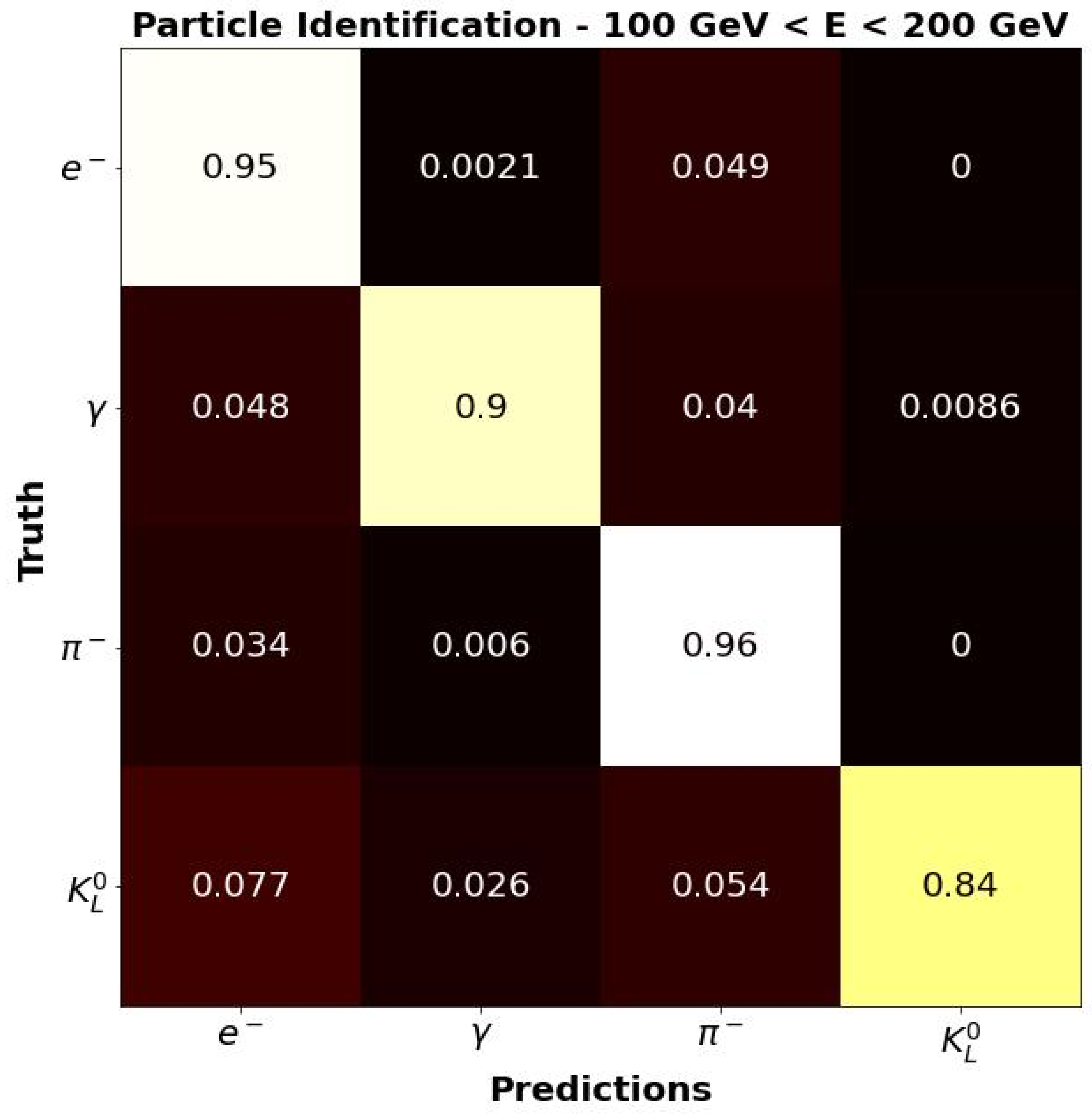




- Offset between optimal clustering and reconstruction more consisten for electromagnetic showers
- Offset generally decreases with higher energy

PARTICLE IDENTFICATION





Accurate particle ID for low and high energies

SUMMARY

- We are able to **efficiently** reconstruct showers within **200 Pilup**
- Learn **energy correction** factors for each shower to improve energy resolution
- Accurately predict **Particle ID** in multi-shower events
- Addition of tracks improves clustering performance
- Step towards an end-to-end differntiable particle-flow algorithm

NEXT STEPS

- Continuing to improve the network architecture
- Exploring other clustering methods
- Particle identification in pile-up events
- Train network on HGCAL simulations
- Integration in CMSSW

Hope to see you at the **poster session on Thursday**

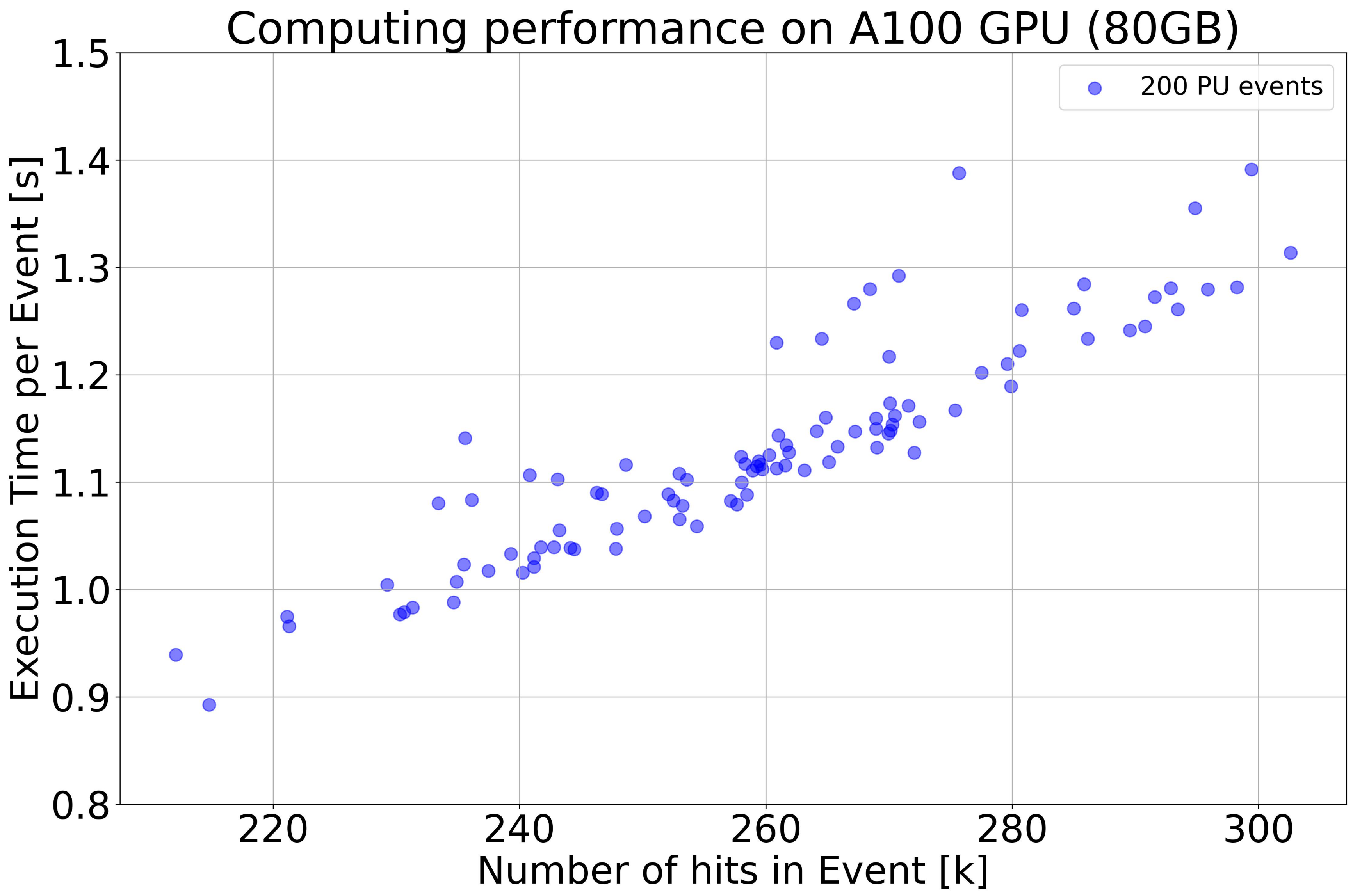
This work has been sponsored by the Wolfgang Gentner Programme
of the German Federal Ministry of Education and Research
(grant no. 13E18CHA)

BACKUP

COMPUTATIONAL REQUIREMENTS

Inference time for 200 PU events only including the network prediction and no clustering (as this can be done in multiple ways).

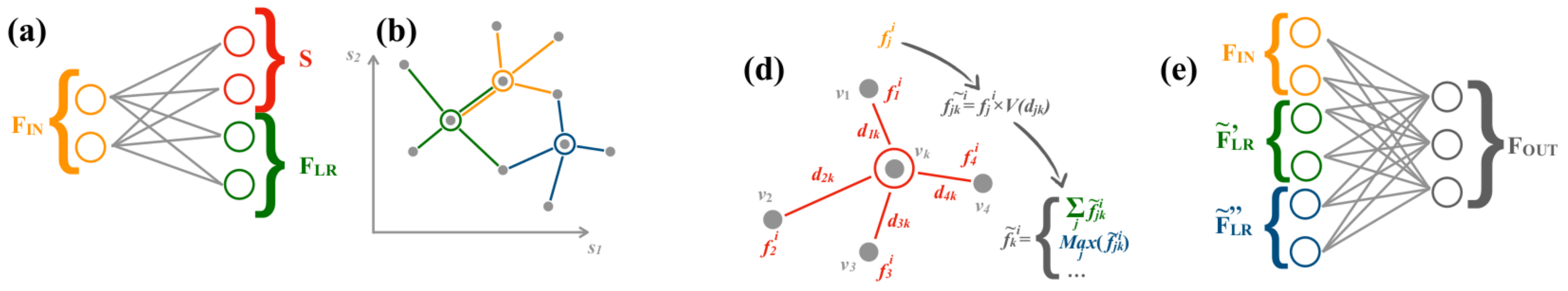
- Inference time scales linear with number of input hits
- In 200 PU events inference needs around one second per event
- **We have yet to explore more options reducing inference time such as:**
  - Downscaling working models
  - Aggressive noise filters or hit reduction via pre-clustering
  - Switching to modern tensorflow versions and updated drivers



GRAVNET

GRAVNET LAYER

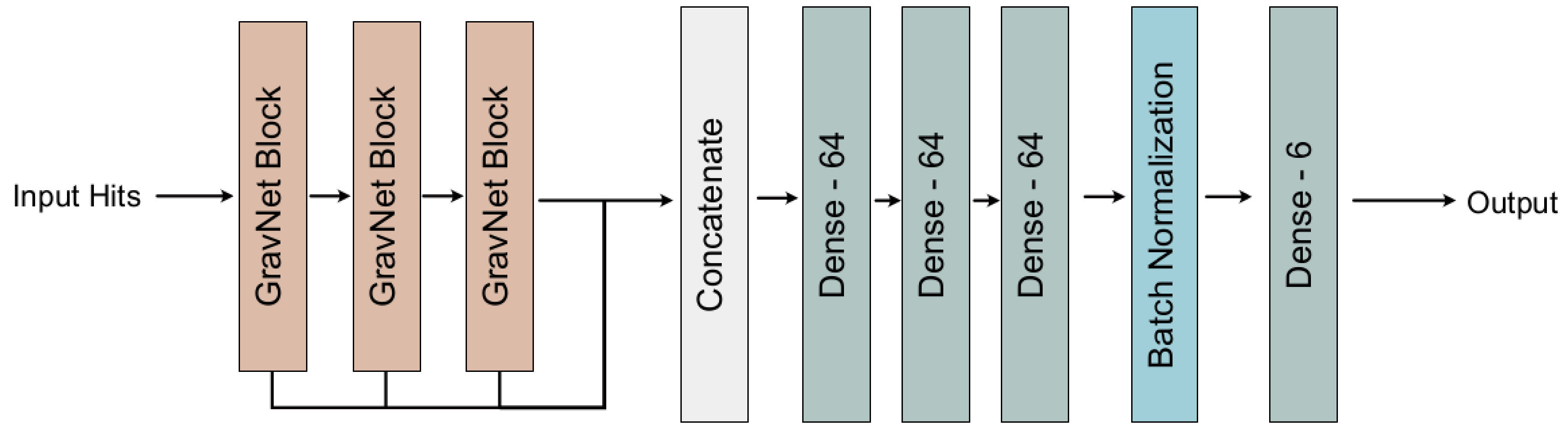
1. Transform input features F_{in} via dense layer into
   - transformed features F_{LR}
   - low-dimensional GravNet coordinates S
2. Use GravNet coordinates to build graph
   $\rightarrow$ connect K nearest neighbours (KNN)
3. Aggregate weighted F_{LR} over neighbours
   - Weights depend on distance between nodes
   - Aggregation is mean and max value of all neighbours
4. Concatenate to produce output F_{out}



https://arxiv.org/abs/1902.07987

NETWORK ARCHITECTURE

1. Transform and normalize inputs
2. Use several GravNet layers to exchange information among neighbours
3. Create ouputs using information from all Gravnet layers
   - Cluster coordinates
   - Confidence β
   - Energy correction factor
   - Particle ID



https://arxiv.org/abs/2204.01681v2