Weekly report

Fangyi Guo

中国科学院高能物理研究所
Institute of High Energy Physics Chinese Academy of Sciences

Working status

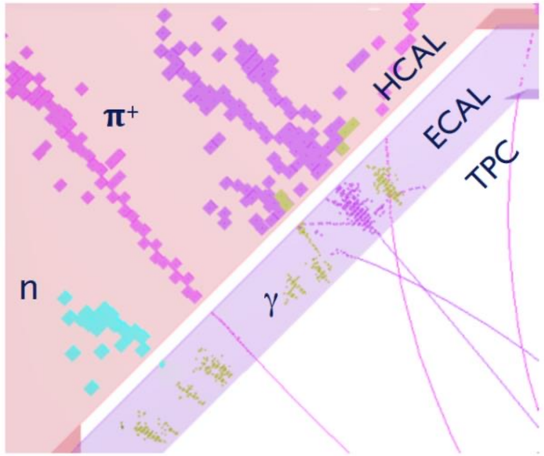
- **For defense:**
  - Preparing the slides. 70% finished.
  - Both anonymous review are received, no critical comments.

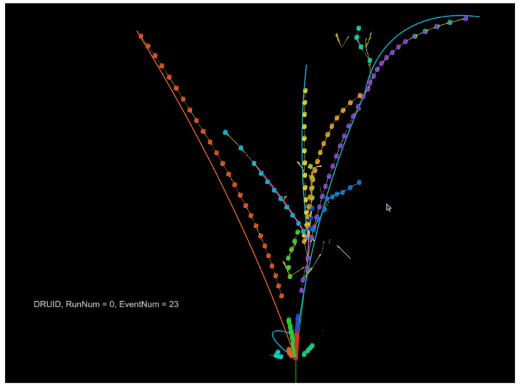
- **ATLAS HH combination:**
  - Discussed with Hideki and Mingshui, connected with ATLAS HHComb contact.
  - Touching the HHComb and HHMLComb framework.

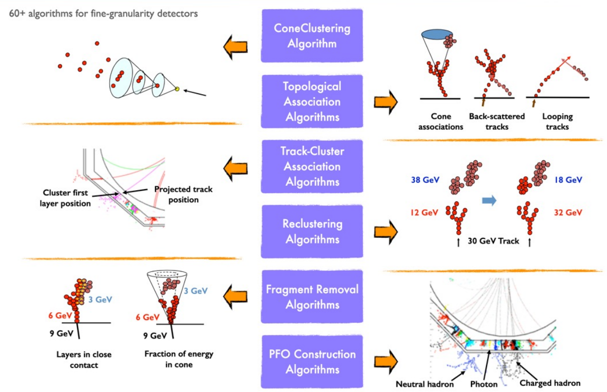
- **Calorimeter reconstruction with DGCNN:**
  - Got some preliminary results.

Introduction

- **Particle Flow approach for the future collider:**
  - Combine the info of tracker + calorimeter.
  - High granularity calorimeter:
    - Precise 3D points measurement.
    - Reconstruction: pattern recognition for clusters.
    - $\sigma_{Jet} = \sqrt{\sigma_{track}^2 + \sigma_{EM}^2 + \sigma_{Had}^2 + \sigma_{confusion}^2}$
  - Particle Flow Algorithm:
    - PandoraPFA: hand-tuned algorithms for clustering. **Artificial recognition.**
    - ArborPFA: arbor-structure of shower structure. **Much Smarter and simpler.**

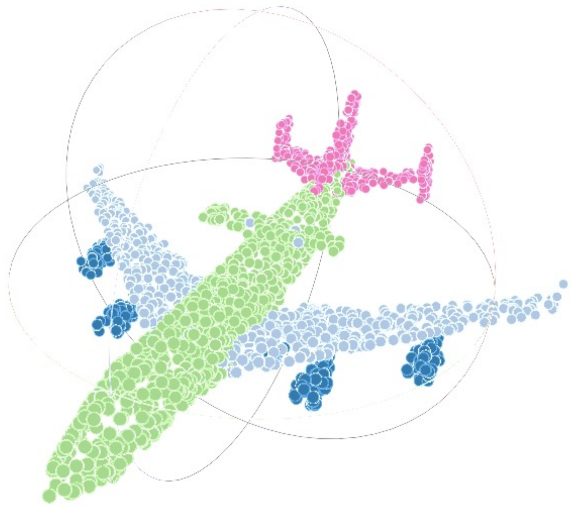


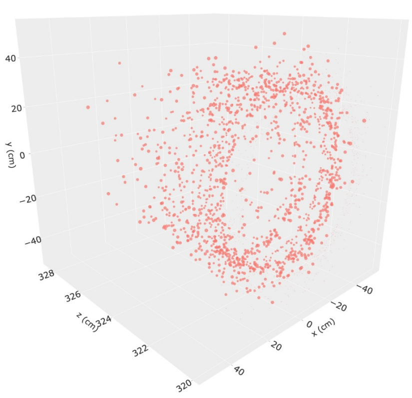


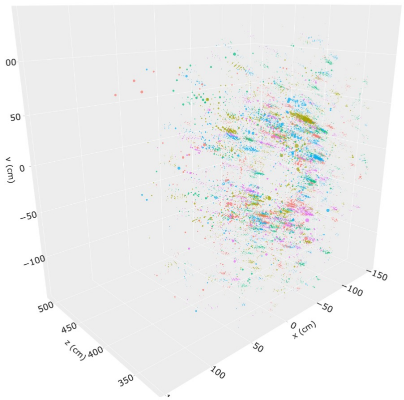


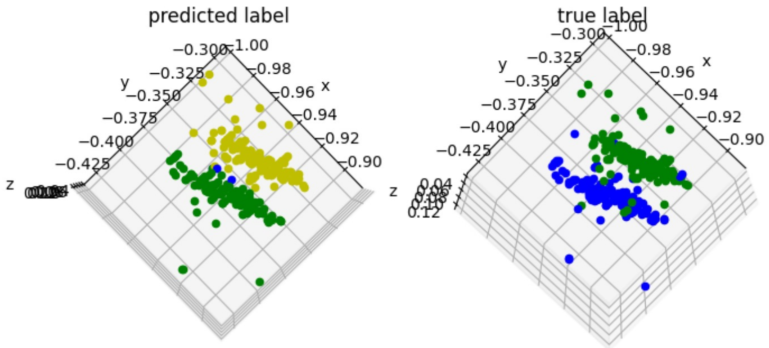
Introduction

- **Particle Flow approach for the future collider:**
  - In 2020s AI era: **point cloud**.
    - HG calorimeter hits are naturally point clouds.
    - **Why not try AI models for the pattern recognition?**
      - Better performance, more flexible model for detector optimization, etc.
  - Reconstructing the clusters in HG calorimeter:
    - [CMS for HGCAL]
    - [Calice for ILD (Si-W ECAL)]



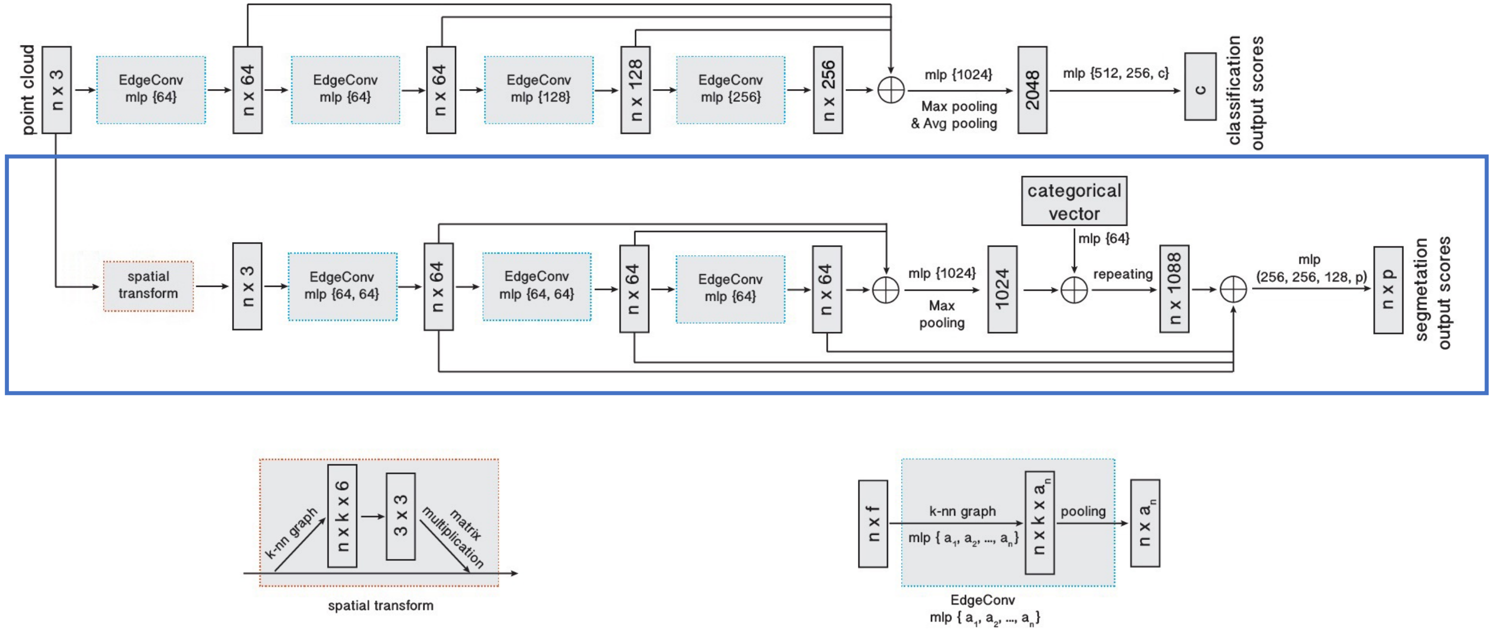


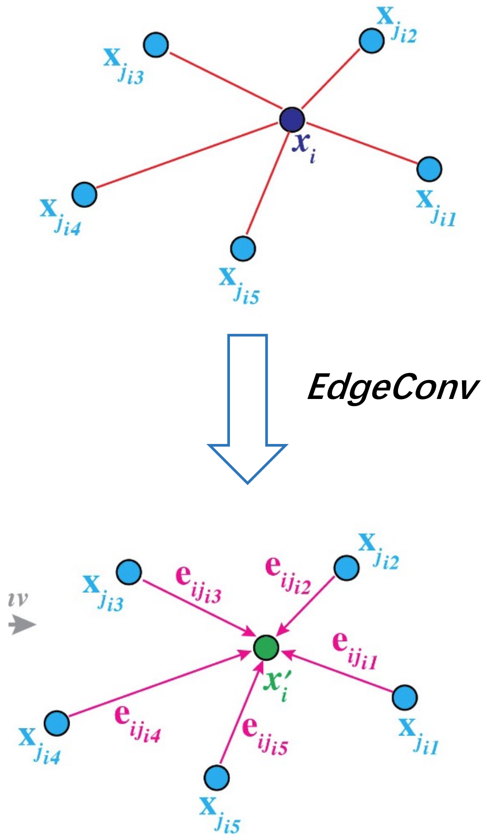




Deep learning clustering

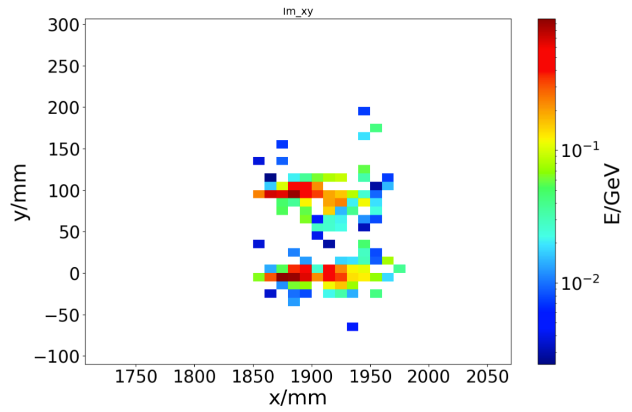
- **Model: Dynamic graphic CNN ([DGCNN])**
  - CNN-based model for graphic dataset.
  - Proposed the ***EdgeConv*** to handle the graph structure.
  - Is able to handle classification and **segmentation** tasks.
  - Application: ParticleNet for c-tagging in CMS.

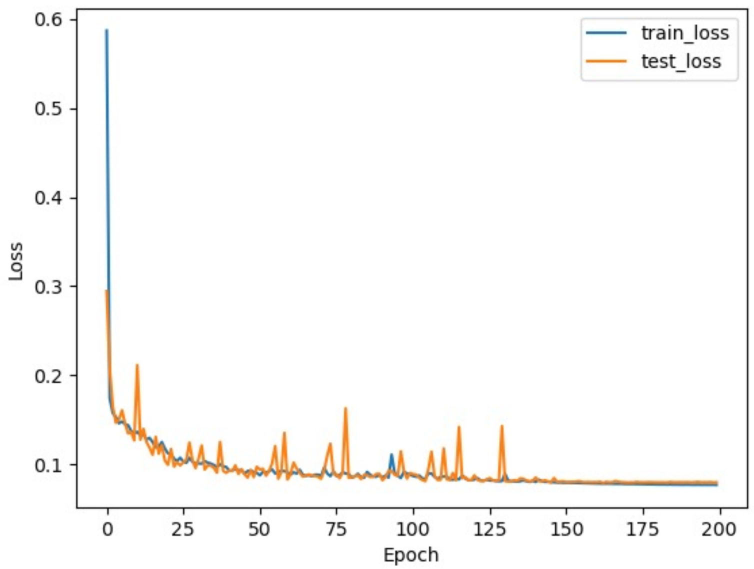


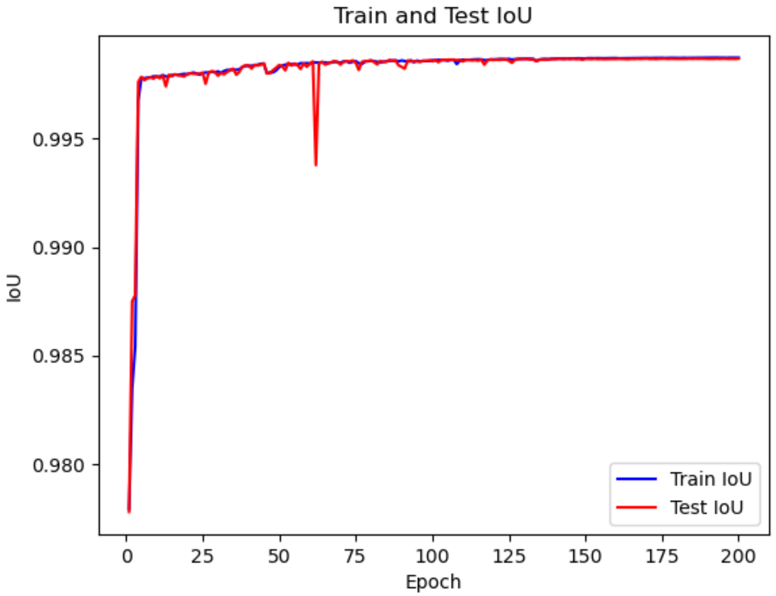


Deep learning clustering

- **Model training for di-photon separation:**
  - Simulated samples: CEPC-v4, 2 nearby photons, 10k events.
    - γ_1: $E_\gamma \in [3,\ 7]$ GeV, $\theta_\gamma \in [88°, 90°]$, $\phi \in [0°, 3°]$
    - γ_2: $E_\gamma \in [3,\ 7]$ GeV, $\theta_\gamma \in [90°, 92°]$, $\phi \in [0°, 3°]$
  - 8k for training, 2k for test, 200 epoch. Run with 4 CPU (8G mem/CPU)+4 GPU (V100). Time consuming: <2h.





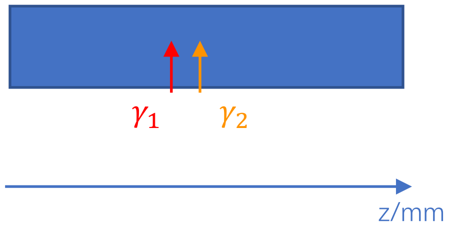


Input: (x, y, z), hit tag.
Loss: cross entropy

Now using the model as a black box, but we are able to tune it if necessary.

Deep learning clustering

- **Application: di-photon separation.**
  - 2 photons, E=5 GeV, $\theta = 90°$, $\phi = 0°$, generate @ ECAL surface, scan the distance.
  - Photon reconstruction with DL + energy splitting:

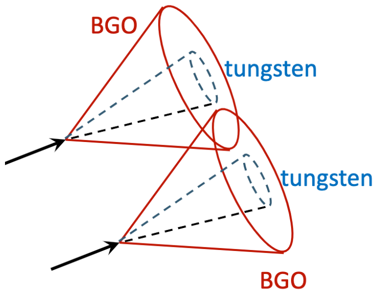


- For hit [a]:

P[1]	P[2]	···

Traditional: hit[a] belongs to particle [i] with max(P[i]).

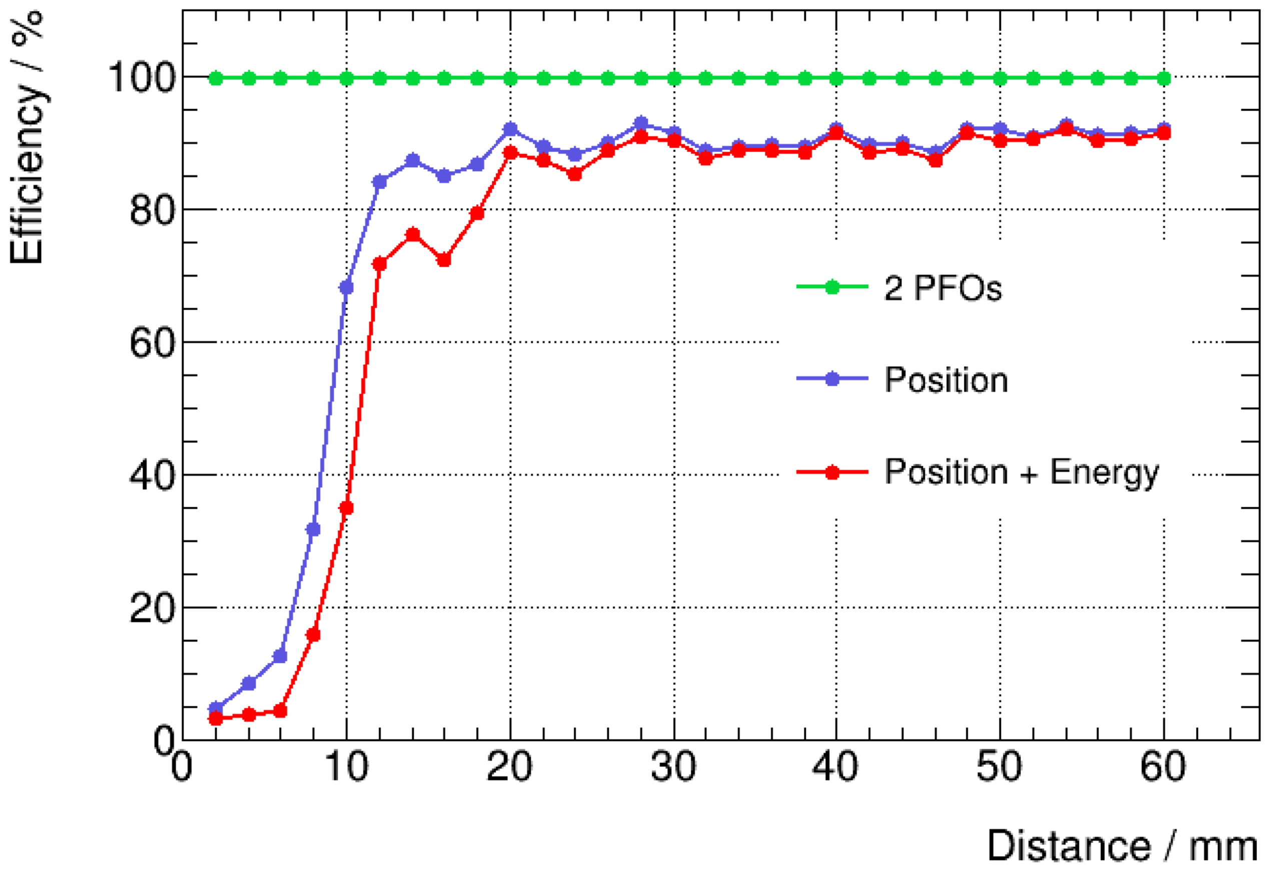
P[i]: probability that hit [a] belongs to shower (particle) [i].
→ Particle [i] contributes P[i] to hit [a].
→ **Split hit[a] to several parts.**

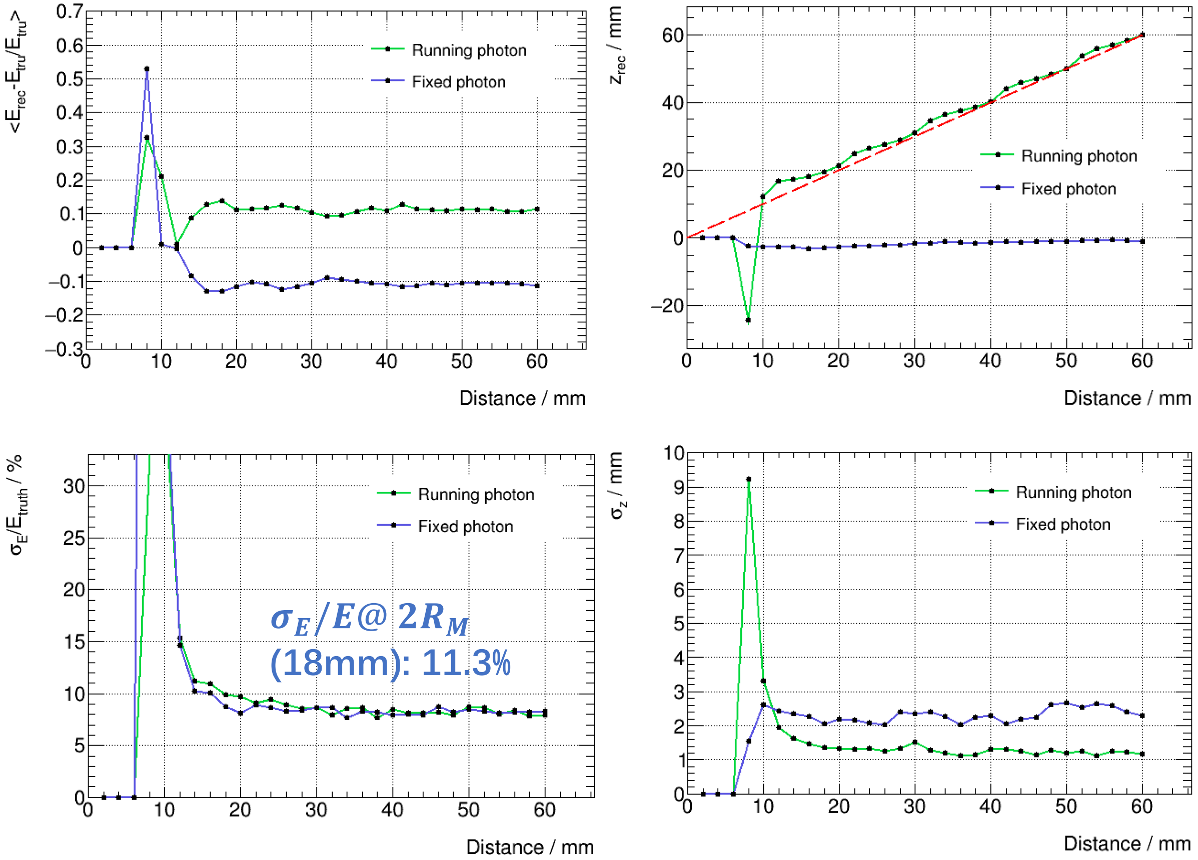


Would be useful for dealing with large overlap showers.

Performance

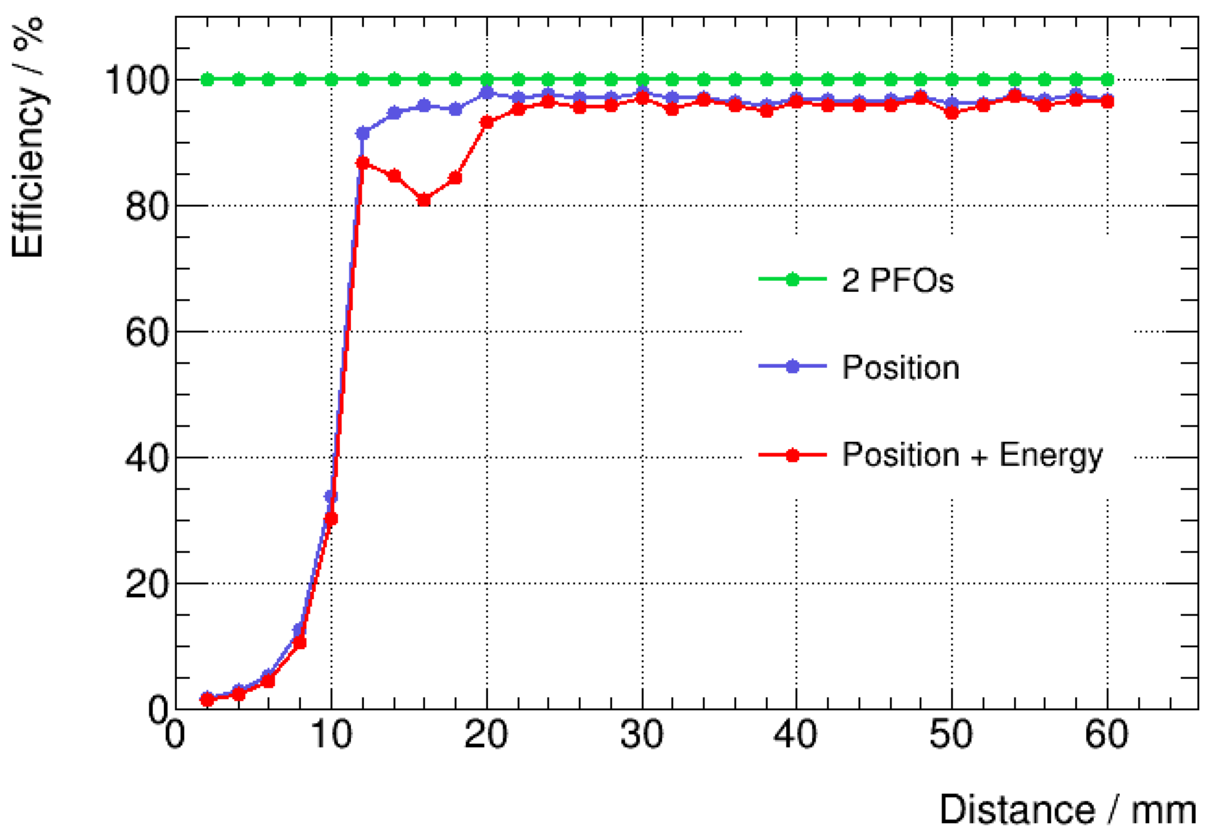
- **Application: di-photon separation (no splitting)**

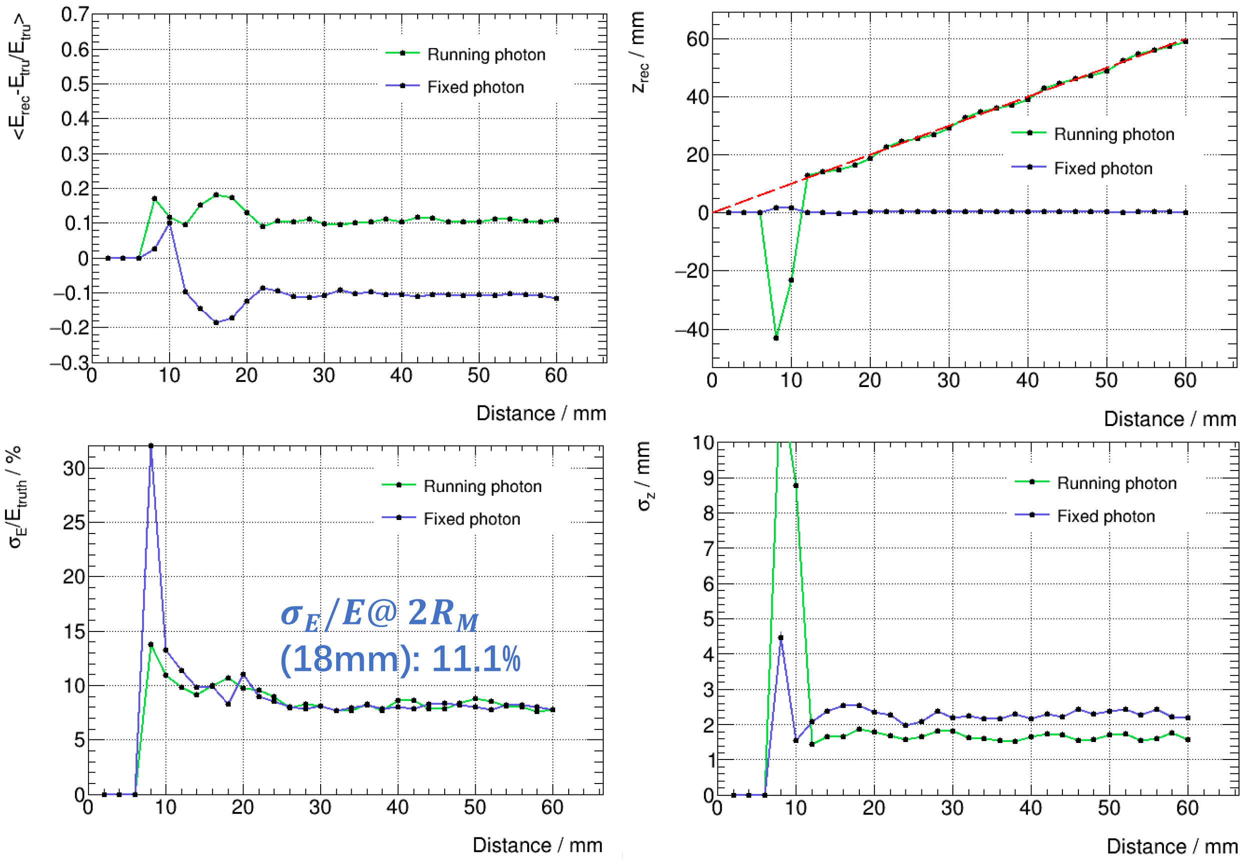




Performance

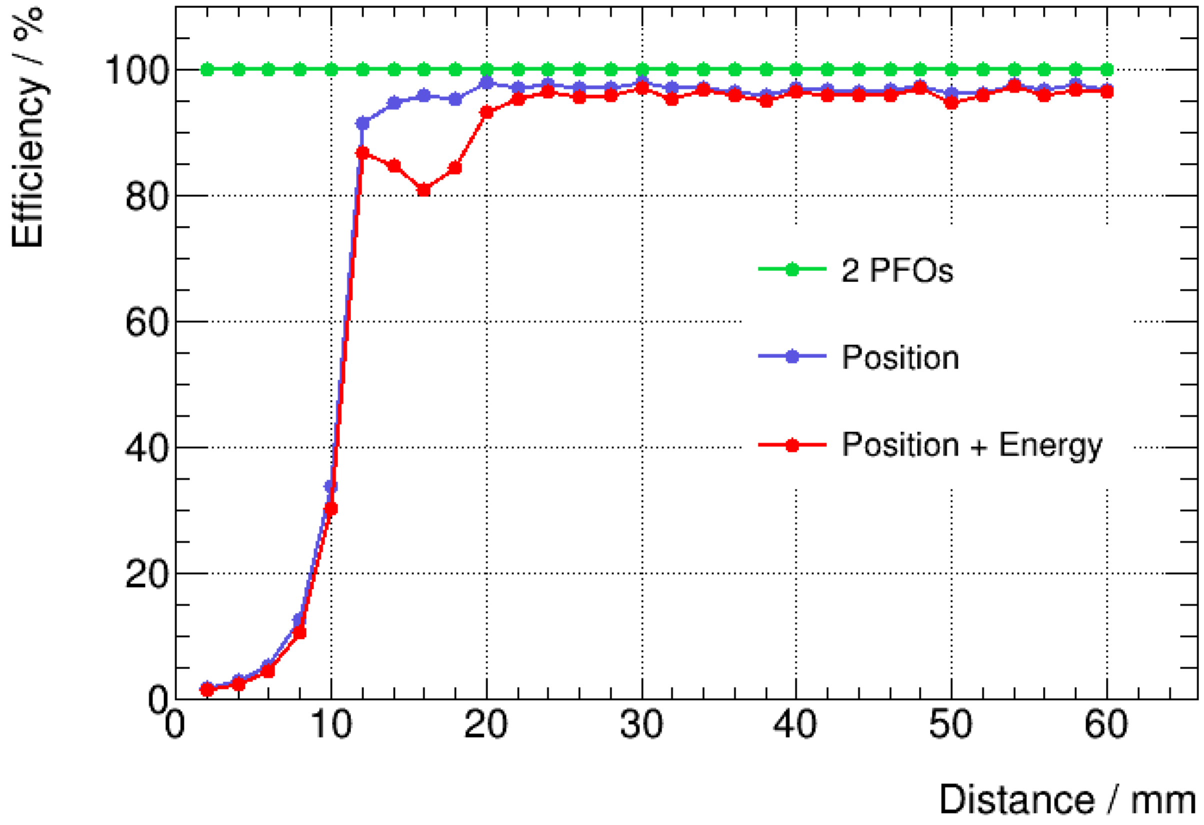
- **Application: di-photon separation (splitting)**

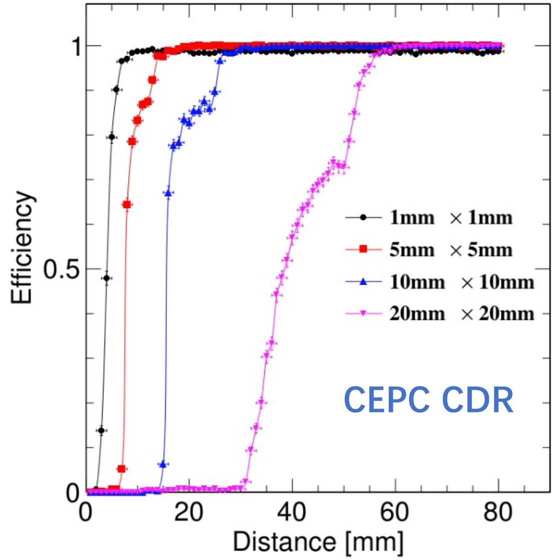




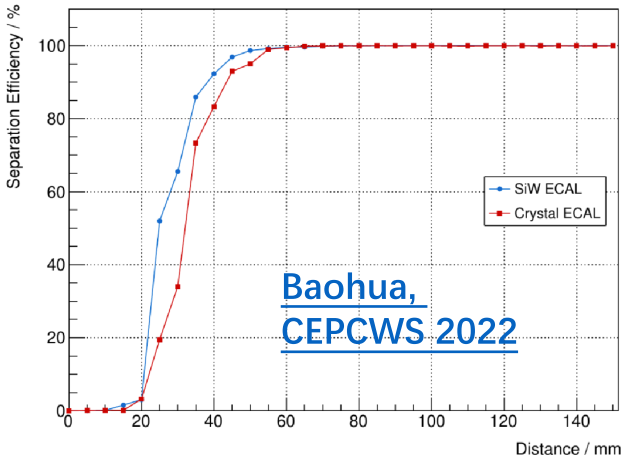
Performance

- **Application: di-photon separation (splitting)**





γ/γ separation with barrel ECAL



Baohua, CEPCWS 2022

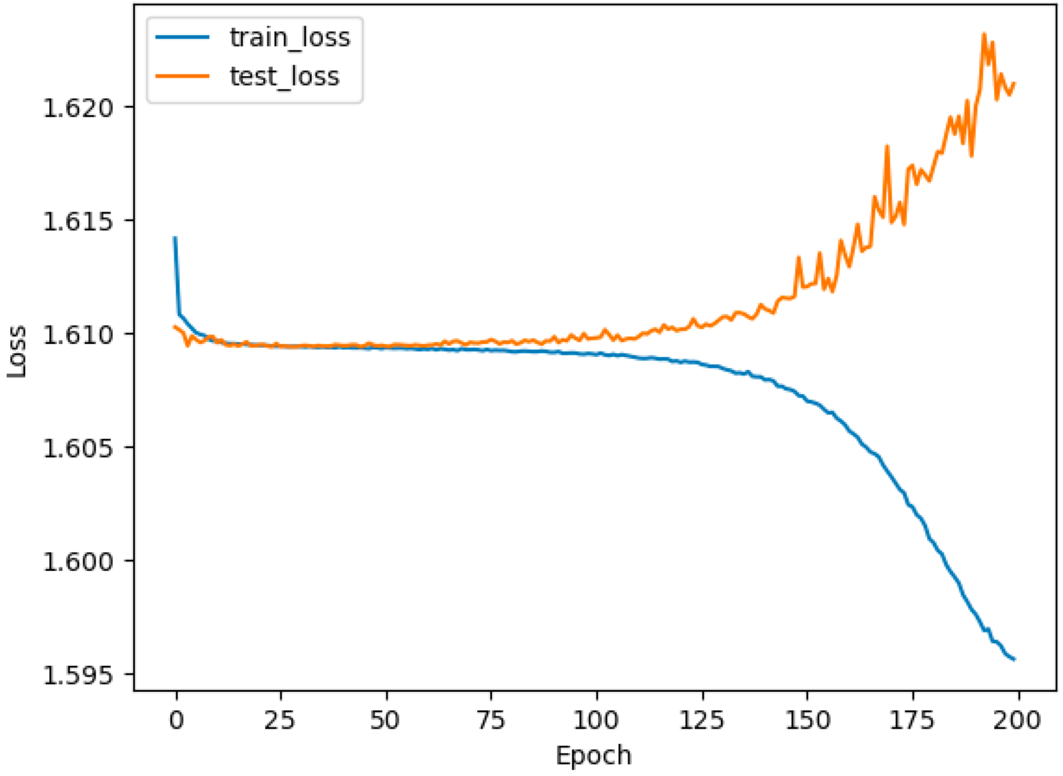
Summary and outlook

- **Calorimeter clustering with deep learning:**
  - Showed the feasibility of separating 2 photons, obtained better performance than CDR.
  - Overtraining problem
    - Tried to train with float number of photons, but the model did not converge.
- **Next step:**
  - Tune the model to overcome the overtraining and un-converge.
    - Model structure, loss, etc.
  - Add energy and time info as point feature.
    - Simply tried but did not converge either.
  - Try other models, e.g. GravNet that CMS and Calice used.
- **Future:**
  - Add track info as bias.
  - Final target: a deep learning based PFA.

Backup

- **Loss function**

With 5 Gam in event



With floated photon number (1~5)