5D Calorimetry: Recent Results

Uli Einhaus, Bohdan Dudar

LCWS Valencia

15.10.2025

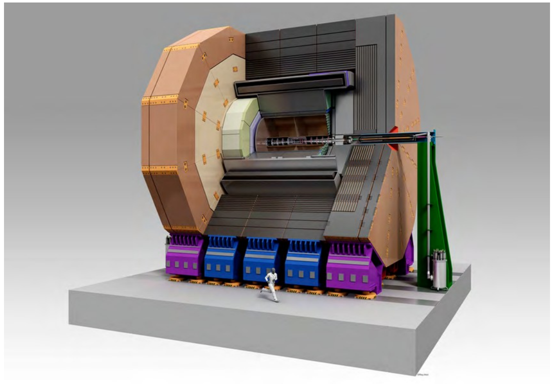
Context

- 5 dimensions of calorimeter data: x, y, z, E, **t**
- Aim: study of implementation and utilisation of ps-timing capability in high-granularity calorimeter to improve performance
- Calo5D project working on various aspects from hardware over reconstruction simulation to physics impact

- This talk: using neural networks / ML and timing to enhance calorimeter performance
- Cooperation with Jan Kieseler (KIT) on ML part

ILD detector setup

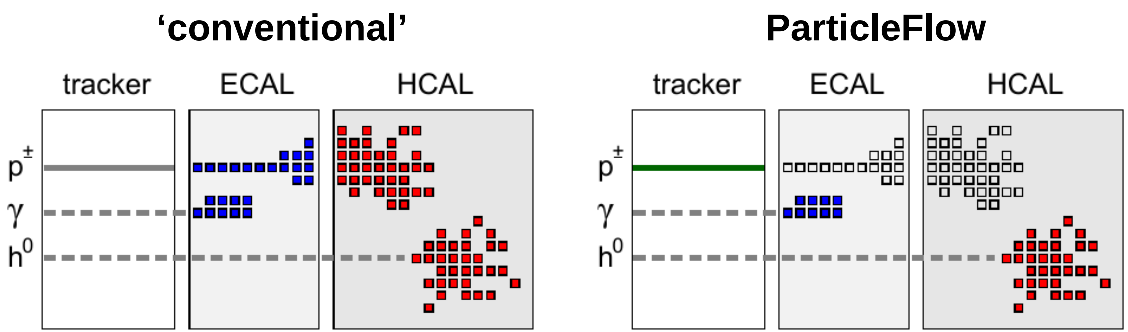
- International Large Detector, developed for ILC, now also proposed for FCC-ee
- Full-sim in DD4HEP, simulation and reconstruction chain 'battle-proven' in large MC productions
- Setup:
  - Si vertex + gaseous tracker, calorimeter inside 3.5 T magnet
  - ECal: 5x5 mm² in 30 layers
  - HCal: 3x3 cm² in 48 layers
- Designed for ParticleFlow with low-material tracker and high-granularity calorimeter



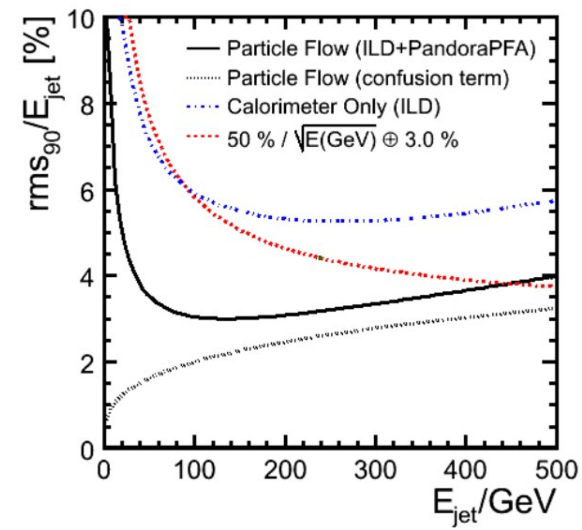
arxiv:2003.01116

ParticleFlow overview

- Idea: utilise the best subdetector for each particle / jet constituent
  - work horse: highly granular calorimeter
  - future collider detectors (e.g. ILD) developed with PFlow in mind, also applied to existing detectors (e.g. CMS HGCAL for HL-LHC)



M. Thompson

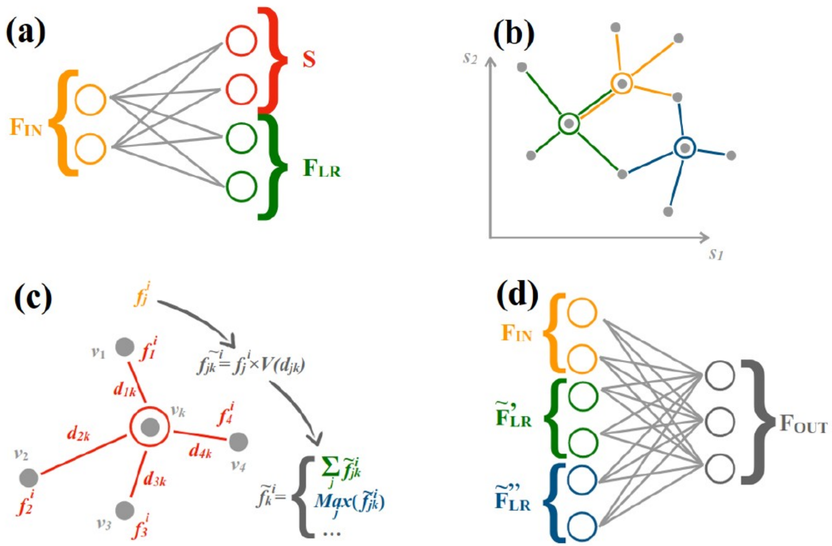


$$\sigma_{jet} = f_{charged} \cdot \sigma_{Tracker} \oplus f_{\gamma} \cdot \sigma_{ECal} \oplus f_{neutral} \cdot \sigma_{HCal} \oplus \sigma_{confusion} \oplus \sigma_{leakage}$$

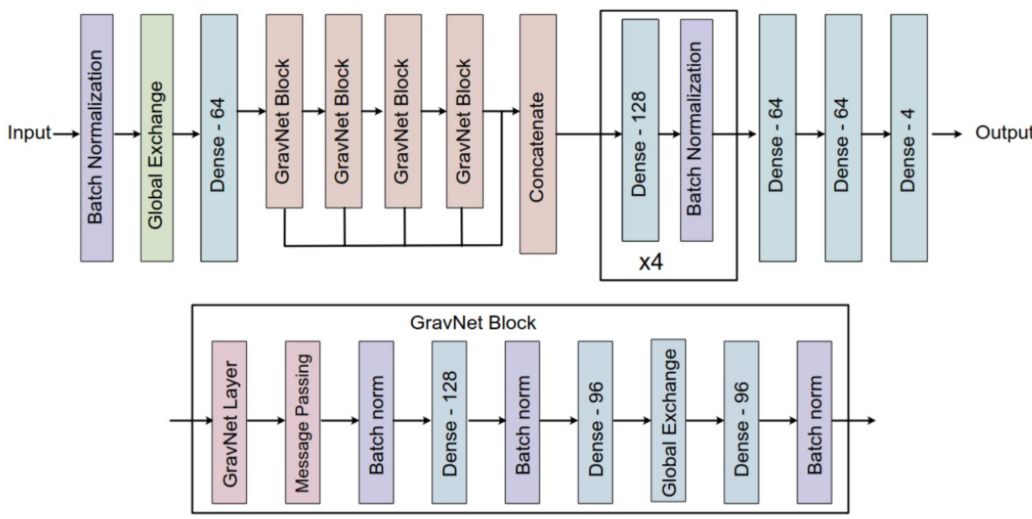
- Use neural networks and timing to reduce confusion term
- Issue with number of channels for fixed-geometry NN (e.g. CNN) → use GNN

Overview GravNet + Object Condensation

- ‘Holistic’ approach with end-2-end reconstruction - clustering and features reconstruction (position, energy, PID) in one integrated Graph Neural Network
- Calorimeter hits are clustered in (low dimension) latent space S
- Feature properties F_{LR} of nearest neighbours in S are combined with different aggregator function to predict object features, weighted with exp(-d²) → Gra**v**Net



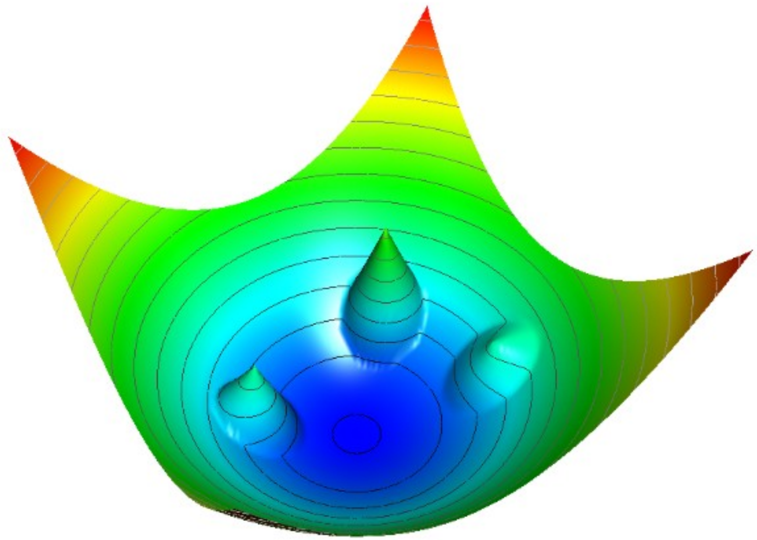
visualisation of the GravNet steps
arxiv:1902.07987



GravNet model setup we used

Object Condensation

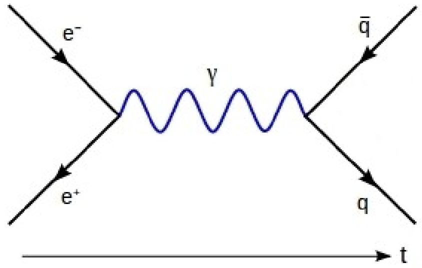
- For clustering introduce attractive potential ~ d^2 for all hits belonging to an object towards each other, and a repulsive potential ~ -d with short range towards all other hits
- To reduce computational effort, introduce β quantifying similarity to object properties, only use hit with highest β per object to generate combined potential
- Additional loss per object of 1-β of the condensation point encourages one hit per object to become representative of object and ‘gather’ related hits into a cluster



Object condensation attractive + repulsive potential visualisation
arxiv:2002.03605

Data set

- Use ILD MC production data set of $e^+e^- \rightarrow (\gamma)q\bar{q}$ at 250 GeV (2fZhad), 45k events
- O(a few 1000) hits per event
- Truth: properties of MCParticle that entered calo and caused energy depositions
  - full MCParticle chain before calo from geant4 is available; in case of multiple contributions to the same hit, largest energy fraction is chosen
  - index, end point in calo (x,y,z), (energy, PID)
  - issue: particles that backscatter from the calorimeter sometimes are not registered, and hits are associated with the original particle that entered the calorimeter
- Simplification cut: all hits with $r_{\varphi,\theta}$ > .5 m from true object are ommitted



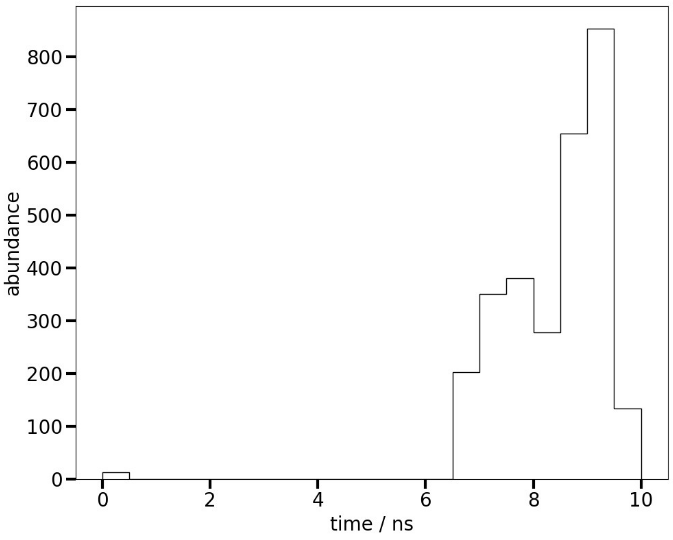
Properties used

- Input: tracks, ECal and HCal hits with 9 properties each
  - x, y, z, $r_{x,y}$, θ, η, E (p for a track), time, ID (isTrack)
  - three timing 'resolutions' are compared: perfect, smeared with 100 ps, none (t==0)
  - time tail cut: t [t>20ns] := -1

- Output: 12
  - target: β, cluster coordinates (x,y,z), position (x,y,z), energy, PID (4 channels)

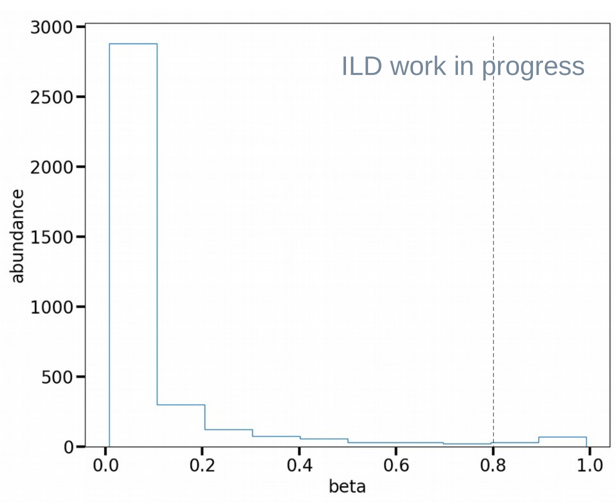
- Loss:
  - object condensation → β, cluster coordinates
  - hit loss → position (energy and class not used atm)

- $$L_{pos} = Huber\left(\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2 + 10^{-6}}, 0.1\right)$$

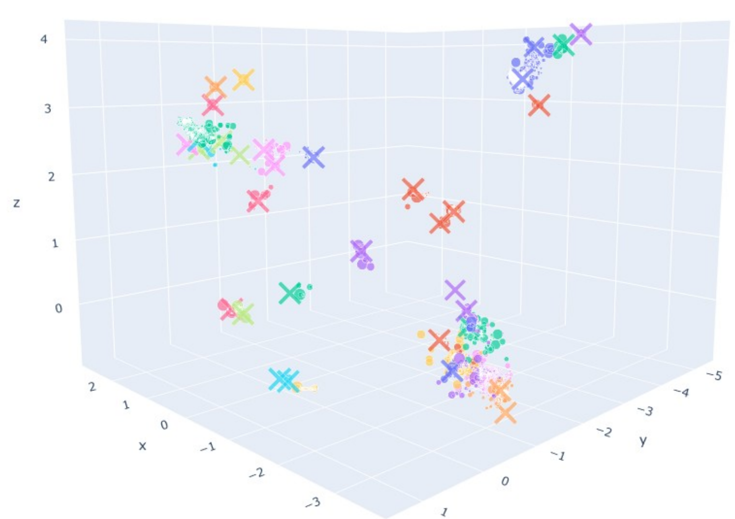


Post-inference thresholding

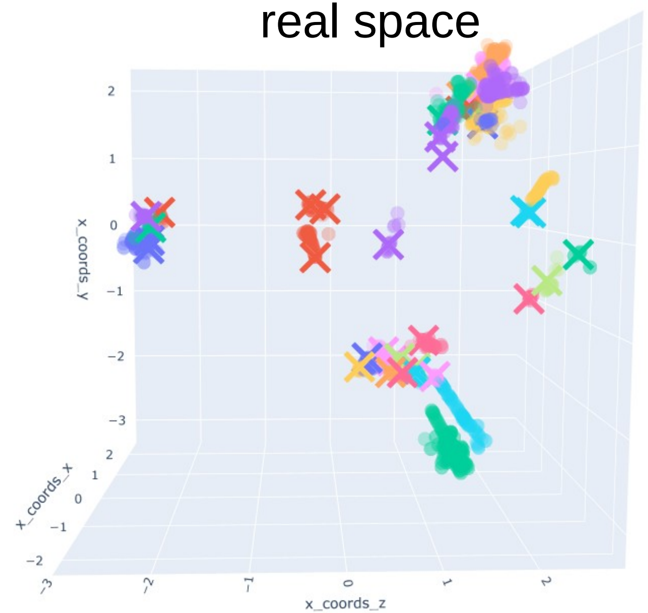
- For predicted objects, i.e. PFOs, thresholds of min beta and min dist to next PFO are chosen after running inference
- beta: few entries with values close to 1 represent the well reconstructed PFOs
- In space projections: matching colours belong to the same MCParticle, X mark predicted PFOs, here for $\beta > 0.8$



cluster space

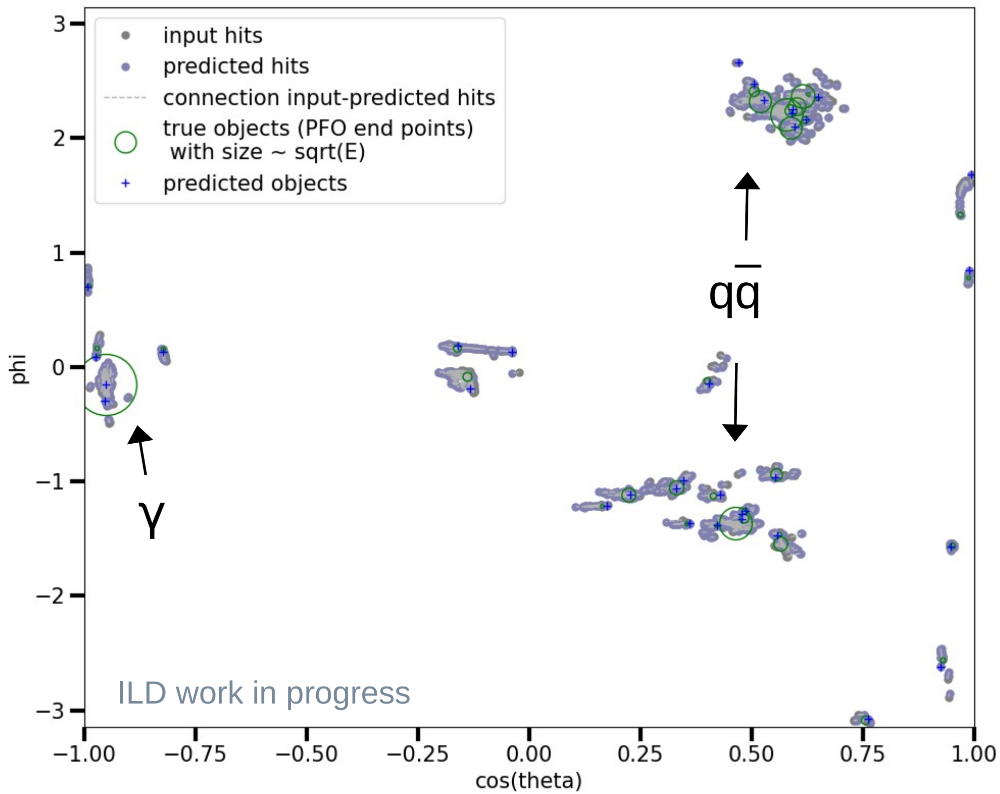


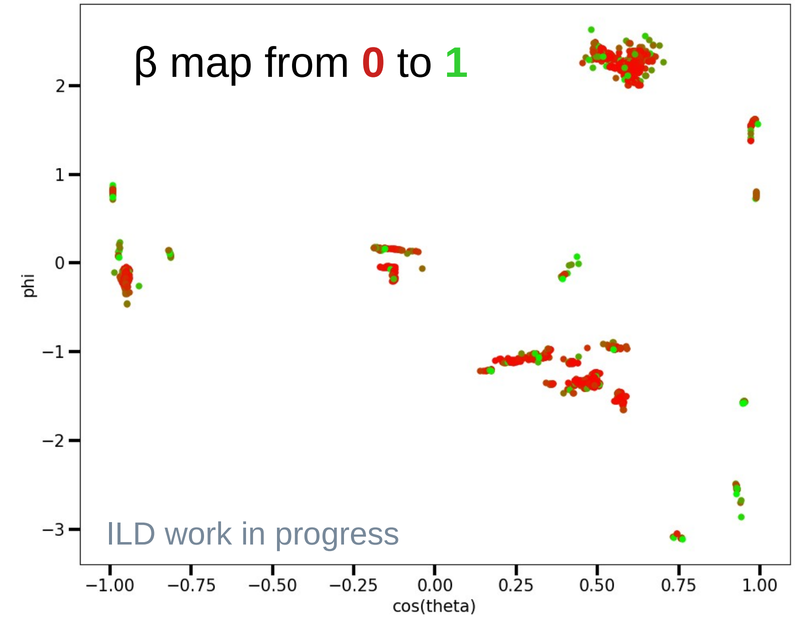
real space



Angular projection of example event

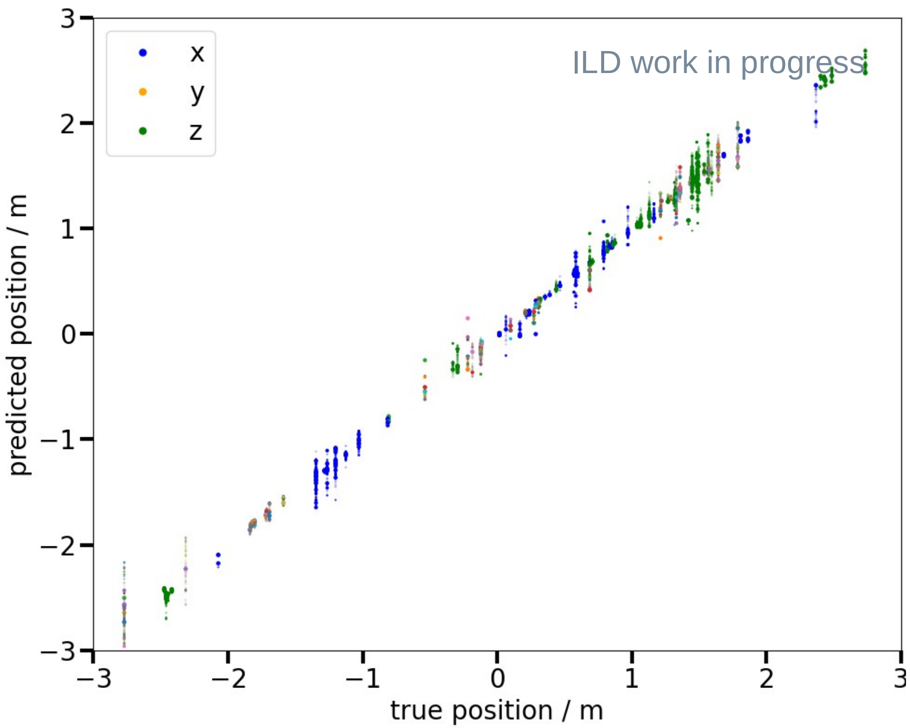
- Predicted positions excellently clustered around targets
- In many cases one predicted PFO (+) per MCParticle (o), here: 35 and 34

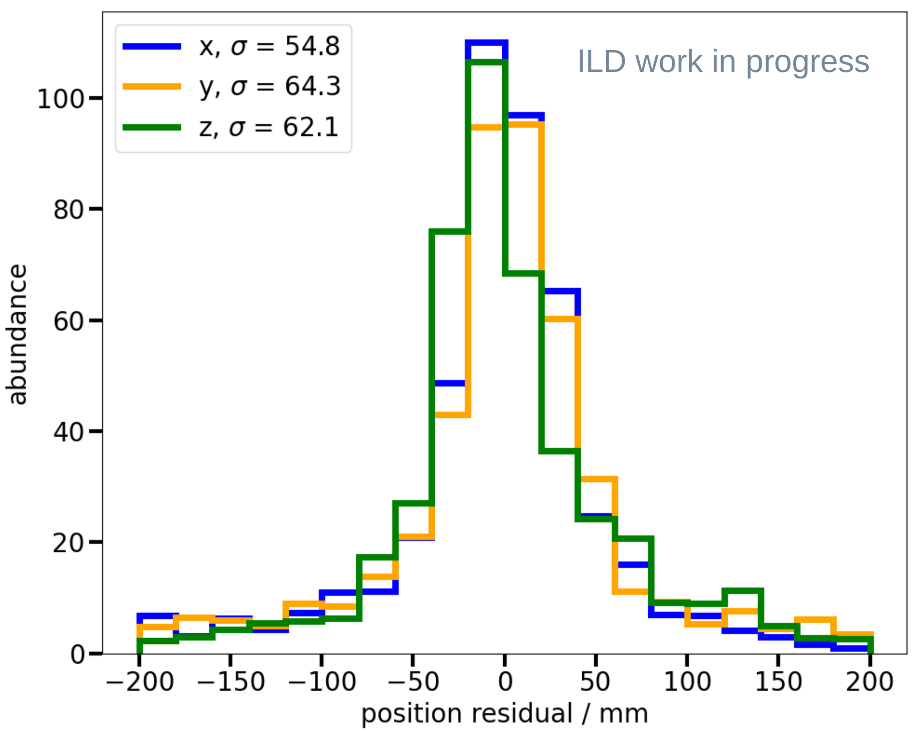




Position reconstruction

- Position per hit, weighted with beta
- Good correlation, hit position prediction uncertainty about 6 cm with current training setup



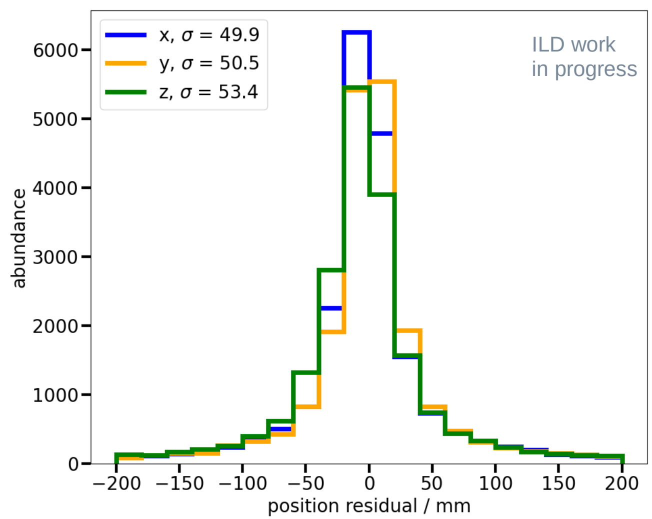


Impact of timing

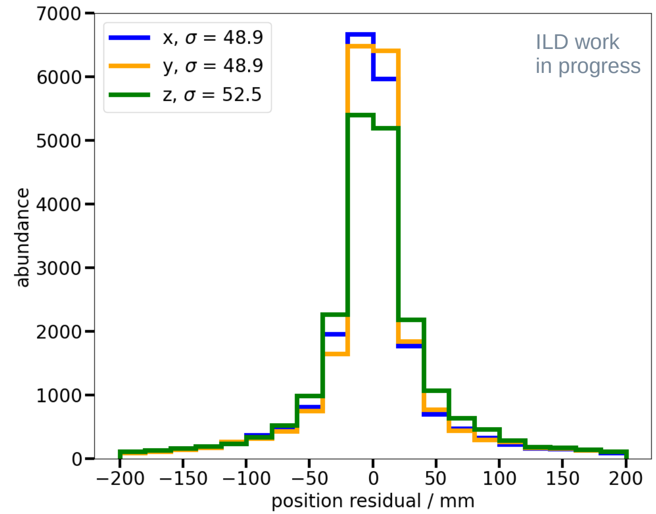
Work in progress!

- PFO position prediction, statistics for 400 test events, ~ 50 mm position uncertainty
- Slight improvement (~ 1 mm; $\Delta_\sigma \approx 0.3$ mm) with adding time

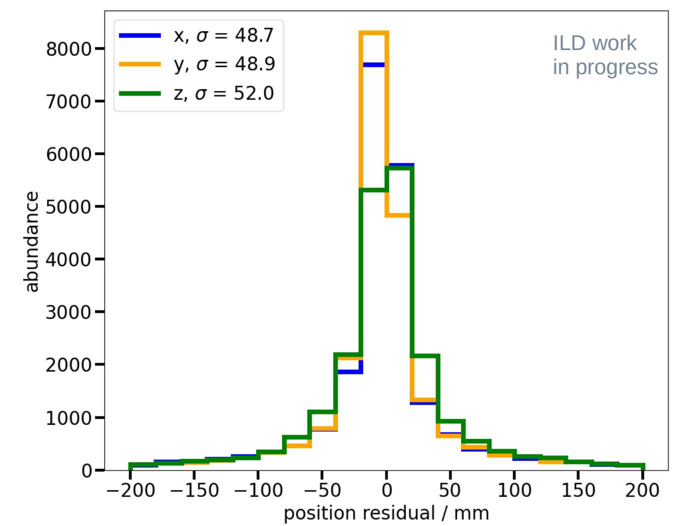
no timing



smearing with 100 ps

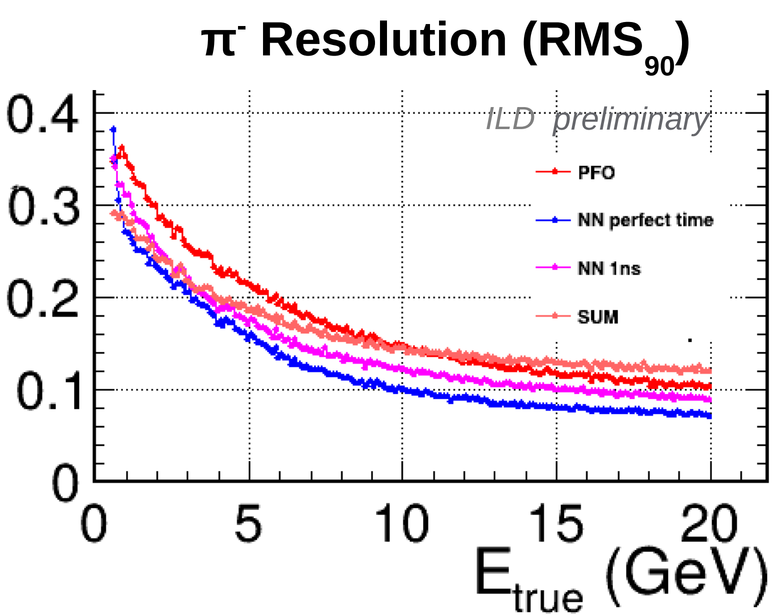


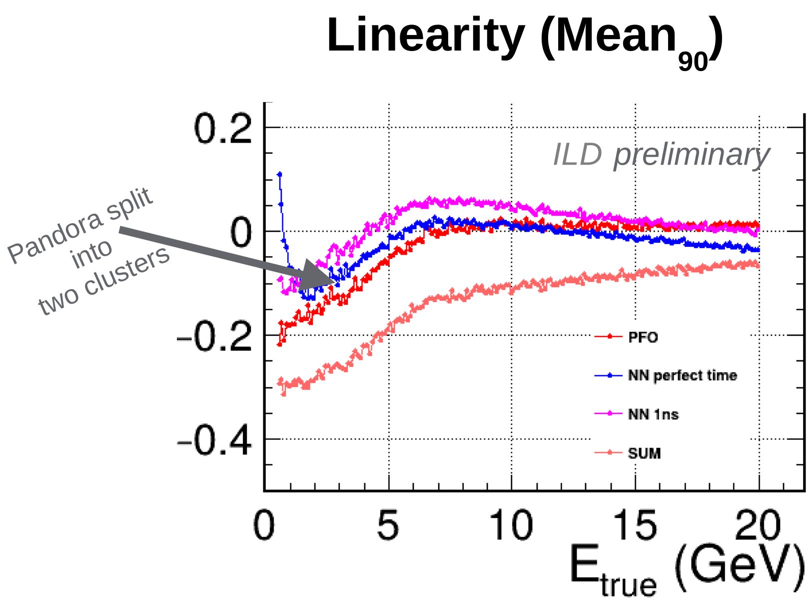
perfect timing



Optimising software compensation with ML

- Starting point: single particles – ML software compensation works well here!





Results

- NN outperforms Pandora on single pions
- Better timing → better resolution!
- Could be further optimized.

Two big buts...

- Pandora is optimized for physics, not single particles!
- Migrating "single-particle" trained NN on physics events is challenging due to confusions

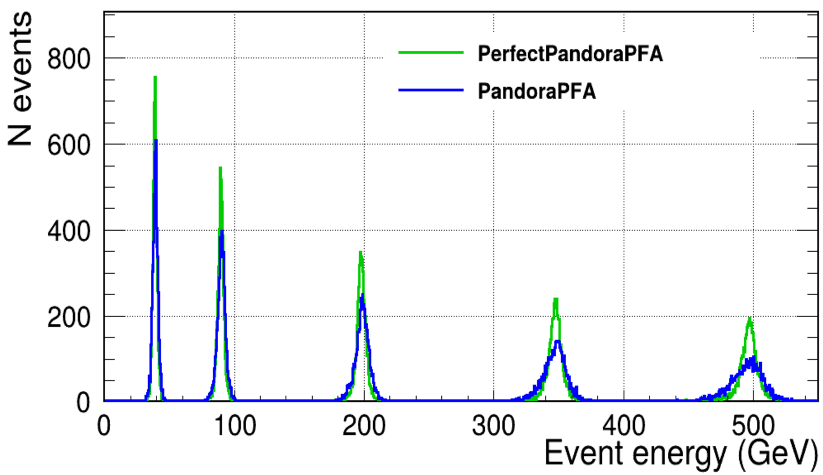
Application to full $q\bar{q}$ events

Setup:

- $Z' \rightarrow q\bar{q}$ (u, d, s)
- ILD full reconstruction
- E_{CM}: 40, 91, 200, 350, 500 GeV

Challenge:

- NN that outperforms Pandora PF
- Estimate role of timing



Current approach

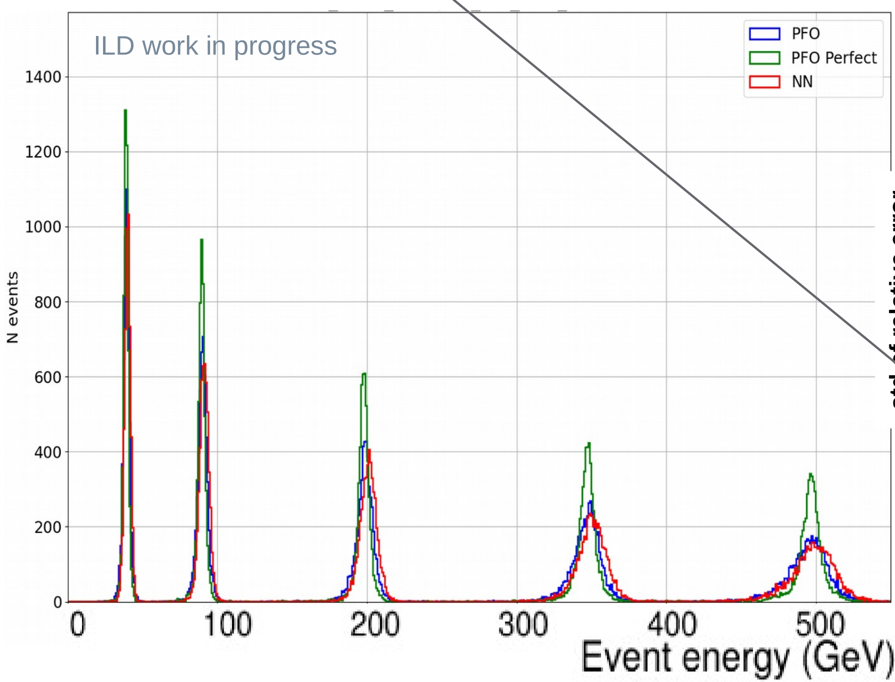
Input	Hit features (details in the back-up): positional, energy, time (perfect time res.)
Model	Dynamic Graph CNN (DGCNN)
Output	Corrected per-hit energies (E_{hit})
Target	Generator level $E_{CM} = \Sigma_i E_{hit, i}$
Loss	MSRE (Mean Squared Relative Error)
Train data	E_{CM}: 40, ~~91~~, 200, ~~350~~, 500 GeV (total 25k evts)
Val data	E_{CM}: 40, 91, 200, 350, 500 GeV (total 25k evts)

First results: slight improvement with NN compared to Pandora

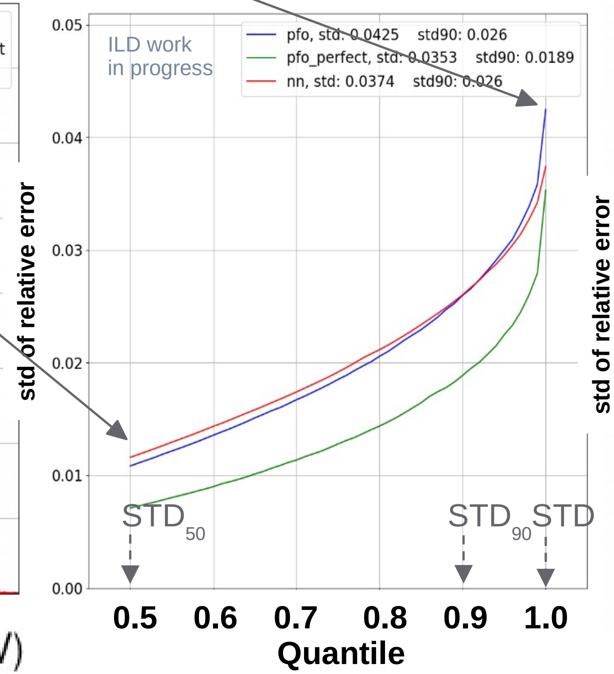
Work in progress!

Results

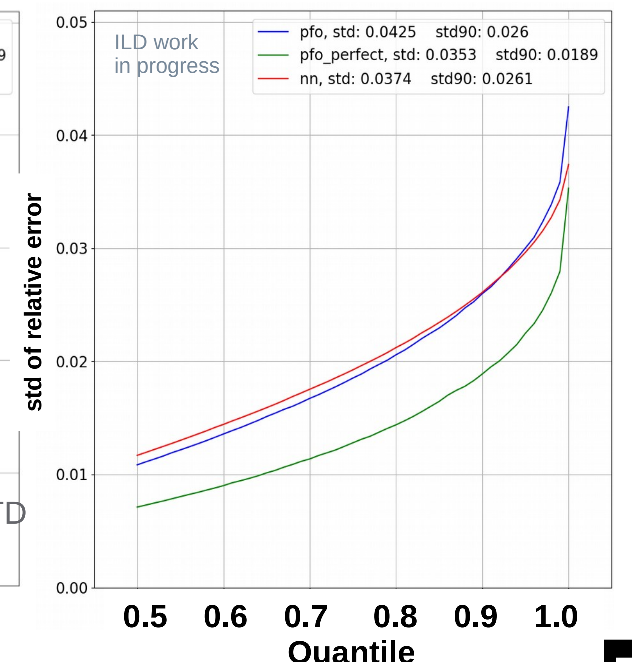
- NN is significantly better than Pandora on the full distribution (fixes bias+outliers)
- NN is similar to Pandora on the core of the distribution (still far from ideal)
- So far, timing shows no impact on performance, but it is too early to draw conclusions



perfect timing



no timing



Conclusion & Outlook

- **Promising results** with GravNet + Object Condensation for **ParticleFlow**: clustering and position prediction work well, optimisation ongoing
- **Additional features** energy and PID to be added next to go for full jet energy resolution
- In **software compensation** DGCNN performs similar to Pandora; with some improvement for the full distribution → working well to correct bias and outliers
- **Timing** is of small impact in both cases
- Work ongoing, lots of space to still explore!

Input features of DGCNN for SWC

Position hit features	x, y, z,	*Cartesian detector coordinates*
	rho, r, phi, theta,	*Spherical detector coordinates*
	d_long, d_perp,	*Distance from hit to the shower CoG*
	is_charged, is_neutral, is_undefined,	*Charged of Pandora PFO*
	is_ecal, is_hcal, is_yoke, is_lcal, is_lhcal, is_bcal,	*ILD subdetectors specific*
	is_barrel, is_endcap, is_ring,	
	layer,	
Energy hit features	e,	*Hit's energy (note: for hits associated to tracks, ignore actual hit's energy and split track's energy evenly among all Pandora cluster hits. To be improved.)*
	e_frac	*Hit energy fraction from the total energy of all hits within Pandora cluster*
Time hit features	t,	*Absolute hit time*
	dt,	*Time relative to the earliest time in the cluster*
	t_minus_c	*Absolte hit time – r/c*
		(For now perfect time resolution assumed)