Fast simulation of the HGCAL with generative models

A newly started project - Preparation and first results on energy regression

Soham Bhattacharya, Sam Bein, Engin Eren,
Frank Gaede, Gregor Kasieczka, William Korcari,
Dirk Krücker, Peter McKeown, **Moritz Scham**,
Moritz Wolf

HamGen

Terascale, 23.11.2021

The DeGeSim Project
MS, SB, K. Borras, W. Fedorko,
J. Jitsev, J. Katzy and DK

HELMHOLTZ AI
ARTIFICIAL INTELLIGENCE
COOPERATION UNIT

RWTH AACHEN UNIVERSITY

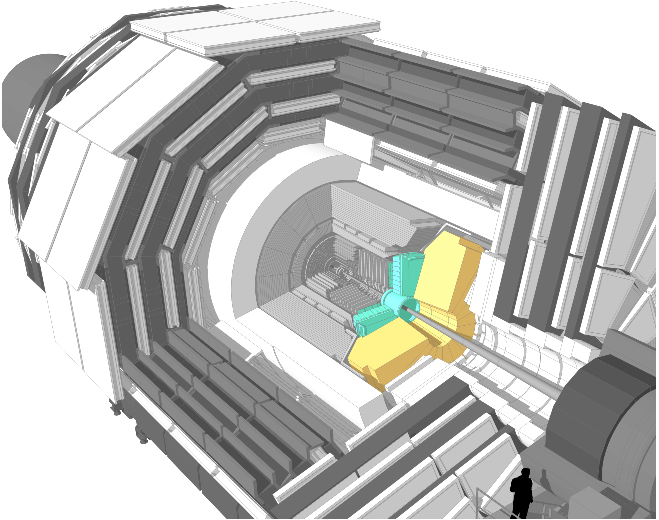
JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

DESY.

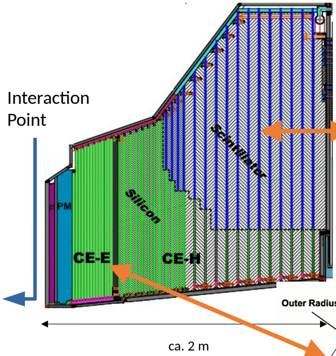
The HGCAL

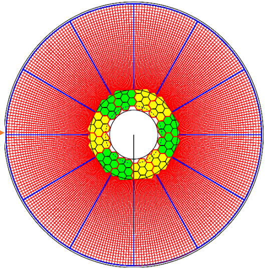
Upgrade for the High-Luminosity LHC in the endcap region

- > Replacing the present forward calorimeter (right) to meet the challenges of the HL phase
- > A high granularity imaging calorimeter
- > Greatly improved spatial resolution and timing
    - Better discrimination of pileup
    - Better shower separation etc.

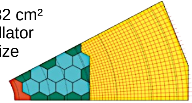


The Geometry of the HGCAL



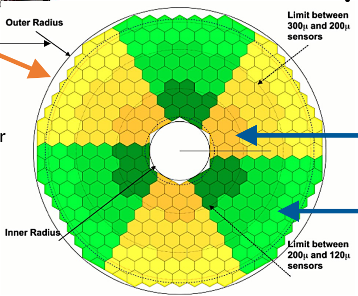


4 to 32 cm² scintillator cell size



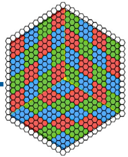
One mixed layer of scintillator and silicon detectors in CE-H (left). Each hexagon represents a wafer with multiple detector cells (pictures below).

- The outer scintillator tiles form a square grid with different edge sizes.
- Note the cut hexagons that define the outer/inner borders (above).

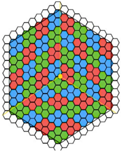


One silicon layer in CE-E (right).

- Each hexagon represents a wafer with multiple detector cells (pictures to the right).
- Higher number of cells in the inner sections.



0.52cm² silicon cell size

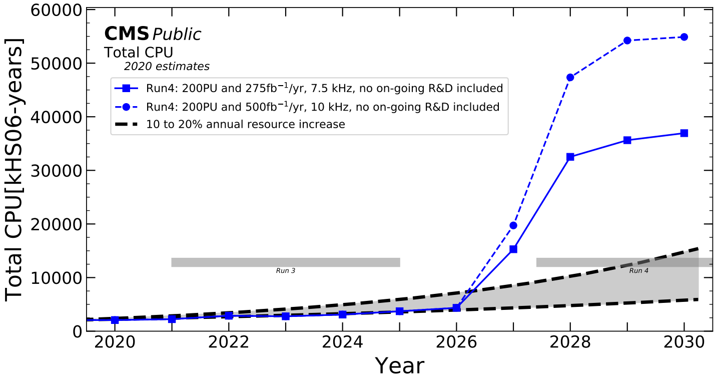


1.18cm² silicon cell size

All Pictures: CMS-TDR-019

Computing Challenge

- High Luminosity phase
  - More particles to simulate
- HGCAL – More cells and channels
  - Complex and time-consuming simulations



CPU time requirements (link)

⇒ Increase in computing time beyond the expected increase in resources.

Vision

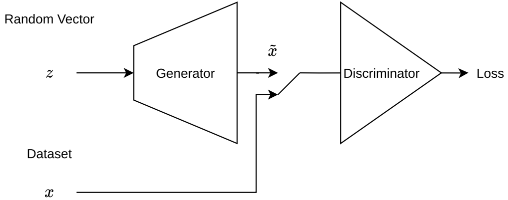
Save CPU time by using a **Neural Network** to simulate the **HGCAL**.

Vision

Save CPU time by using a **Neural Network** to simulate the **HGCAL**.

- Multiple proof-of-principle demonstrations for GAN/VAE-based fast calorimeter simulation
- But no working CMSSW implementation yet
- Use Graph Neural Networks (‘GNNs’) to deal with sparsity and irregular geometry

Generative Adversarial Networks



Loss eg. Minmax:

$$\mathbb{E}_x\left[\log D(x)\right] + \mathbb{E}_z\left[\log\left(1 - D(G(z))\right)\right]$$

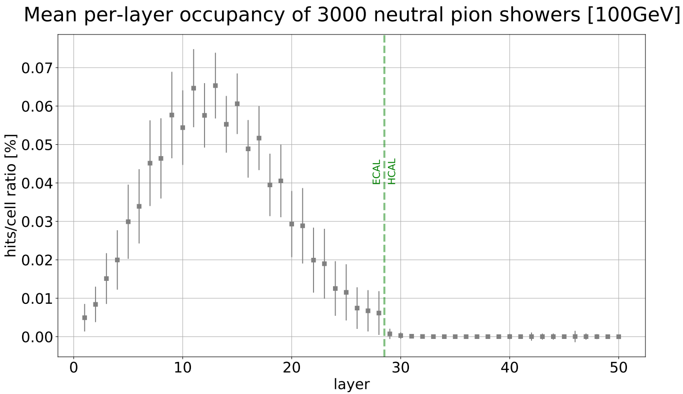
Maximize for the generator, minimize for the discriminator

First proposed by Goodfellow et al (`arXiv.org` 1406.2661)

Necessary Steps

Starting the project

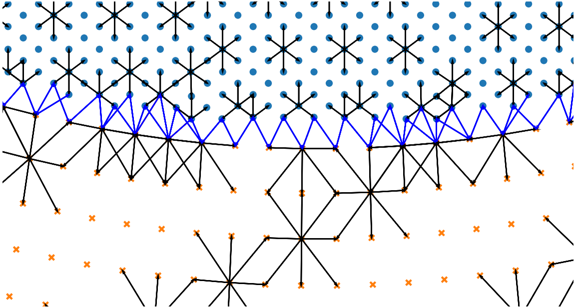
- Training sample production
  - Training will be on single particles (photon, leptons, hadrons)
  - Downside of high granularity $\Rightarrow$ **Sparse data** (right)
- Make the geometry and cell properties accessible to the deep learning frameworks
  - Construct the neighborhood between the cells for the graph representation
- Eventually, development of a generative model to handle sparse data and large graphs



Geometry Extraction

Make the HGCAL geometry available to the Deep Learning frameworks

- CMSSW provides
  - Cell positions
  - Cell properties, e.g. type
    - Silicon
    - Scintillator
  - Neighboring cells within silicon or scintillator
- Full detector neighborhood construction only once
  - Min. distance (x, y) neighbors beyond subdetector borders
  - Nearest cell in $\pm z$ direction



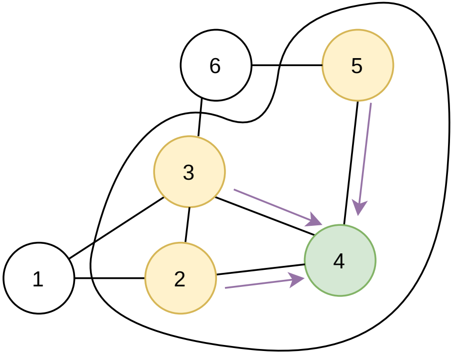
Part of a layer in the hadronic part of the HGCAL

- Black: Some connections within the silicon/scintillator parts
- Blue: The connections between silicon and scintillator cells
- Max distance 0.5 cm between the corners of the cells

ML on graphs

Message Passing: Information propagation for GNNs

Update Node 4
(w/ Neighbors, Messages)



1 Message:

$$\mathbf{Msg}_{j\to i} = \mathrm{Msg}\left(\mathbf{x}_i, \mathbf{x}_j, \mathrm{edge}_{j\to i}\right)$$

2 Aggregate:

$$\mathbf{Aggr}_i = \underset{j\in\mathcal{N}(i)}{\mathrm{Aggr}}\, \mathbf{Msg}_{j\to i}$$

3 Update:

$$\mathbf{x}_i \leftarrow \mathrm{Upd}\left(\mathbf{x}_i, \mathbf{Aggr}_i\right)$$

Energy Regression

Discriminator and a regressor have similar tasks:

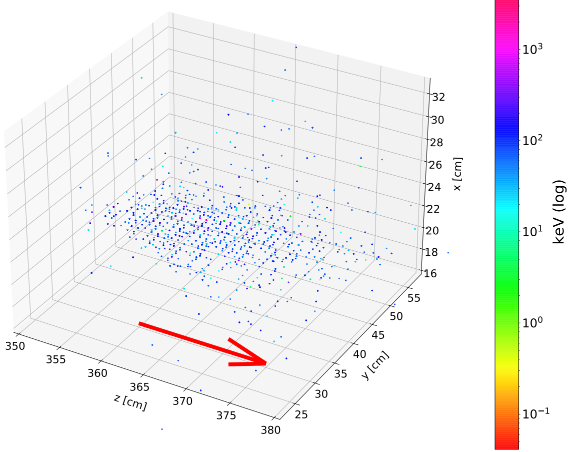
- > Discriminator maps the graph to a probability
- > Regressor maps the graph to a regressed variable

⇒ Technical implementation similar

⇒ Capabilities of GNN architectures measurable in a regression task

Idea : Test technical capabilities and the power of GNN architectures with energy regression on simplified datasets.

3D development of a 100GeV neutral pion shower

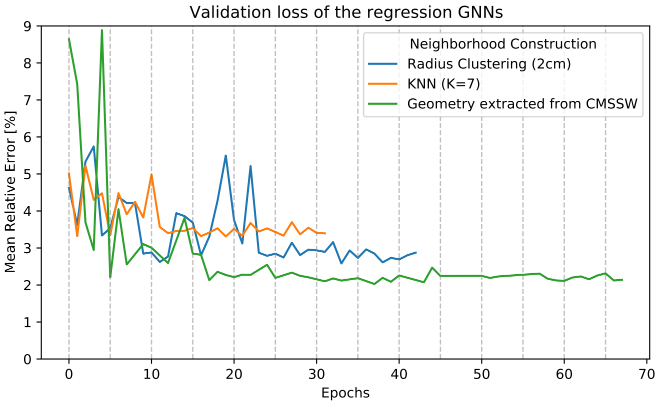


W. Korcari

Energy Regression GNN on HGCAL showers

Collaboration with W. Korcari, G. Kasieczka (UH)

- Simulate only the relevant part of the HGCAL
- Photons with $[50, 100]\,\mathrm{GeV}$ and $\eta = 2$
- Loss: $L = \frac{1}{n}\sum_i \left|1 - \frac{\hat{y}_i}{y_i}\right|$
- Early stopping



⇒ Technical capabilities proven

⇒ GNN shows very promising performances

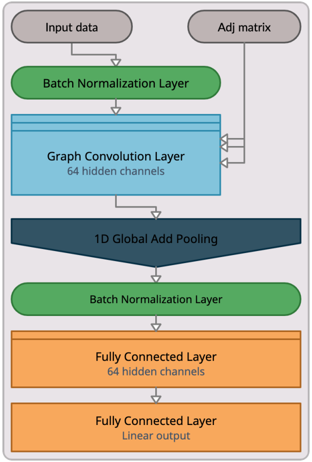
⇒ Geometry extracted from CMSSW leads to the best results

Architecture of the GNNs on HGCAL showers

Collaboration with W. Korcari, G. Kasieczka (UH)

- PyTorch implementation:
  - 3 GCNConv layers
  - 2 fully connected layers
  - 2 Batch normalization layers
- Hyperparameters:
  - Learning rate 0.001
  - Batch size 64
  - 300 epochs

W. Korcari (UH)

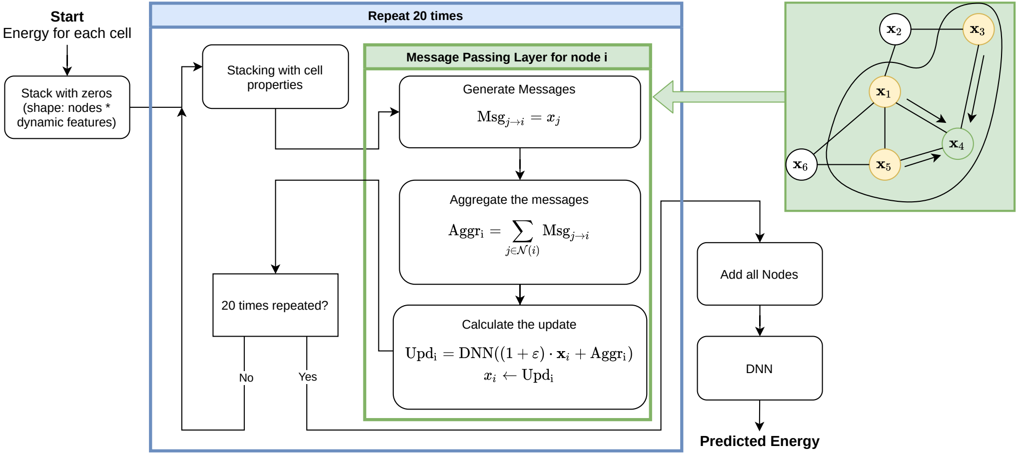


Energy Regression GNN on a Toy Dataset

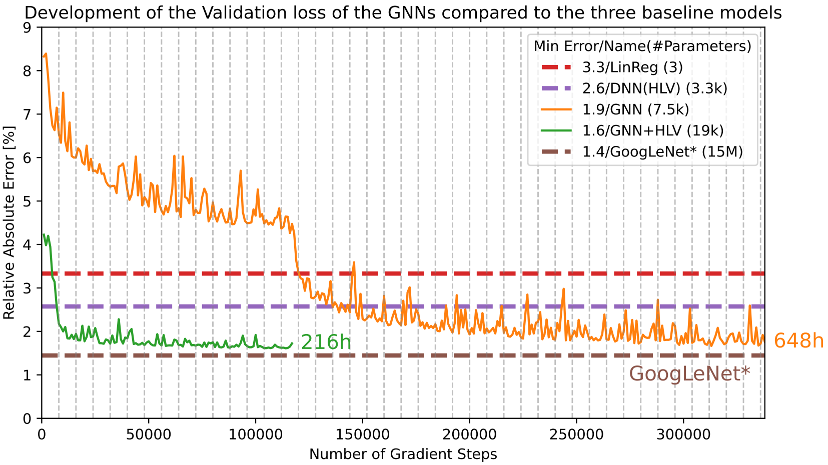
Simplified dataset for Performance Studies

- A CLIC inspired toy dataset (`arXiv.org` 1912.06794):
  - ECAL: 25 layers of 51 × 51 cells
  - HCAL: 60 layers of 11 × 11 cells
  - Electron showers, $p_\mathrm{T} \sim \mathrm{U}_{[0\,\mathrm{GeV},500\,\mathrm{GeV}]}, \varphi \sim \mathrm{U}_{[0,0.35\mathrm{rad}]}, \eta \sim \mathrm{U}_{[-0.524\mathrm{rad},0.524\mathrm{rad}]}$
- Some Models are provided with following variables ("High Level Variables"):
  - E_elmag Sum of cells in the electromagnetic part
  - E_hadron Sum of cells in the hadronic part
  - η (Energy weighted average)
  - ϕ (Energy weighted average)
- Loss is the mean relative error: $L = \frac{1}{n}\sum_i \left|1 - \frac{\hat{y}_i}{y_i}\right|$
- Early stopping

The Architecture of the GNNs on the Toy Dataset



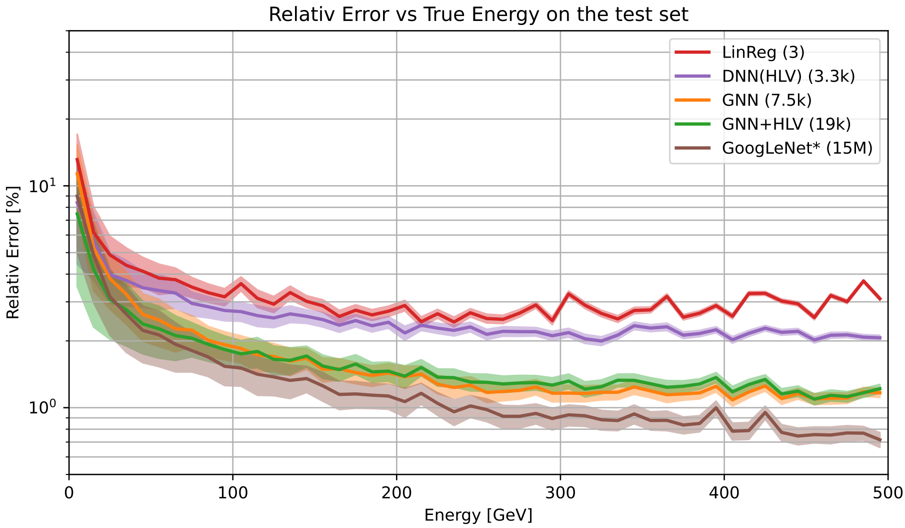
Training of the Energy Regression GNNs on the Toy Dataset



⇒ GNNs approach the performance of state-of-the-art Models

⇒ HLV help convergence and shorten training time

Performance Comparison



Summary

- Geometry was extracted from CMSSW
- Technical infrastructure has been set up
- GNNs show good performance in a regression task
- Studies on generator architectures ongoing

Thank you!

DESY. Deutsches
Elektronen-Synchrotron
www.desy.de

Soham Bhattacharya, Sam Bein, Engin Eren,
Frank Gaede, Gregor Kasieczka, William Korcari,
Dirk Krücker, Peter McKeown, **Moritz Scham**,
Moritz Wolf
CMS-E
moritz.scham@desy.de