# Jet Energy Corrections with GNN Regression using Kubeflow

Daniel Holmberg, CERN IT

# Kubeflow at CERN

Centralized ML platform to improve resource utilization across CERN

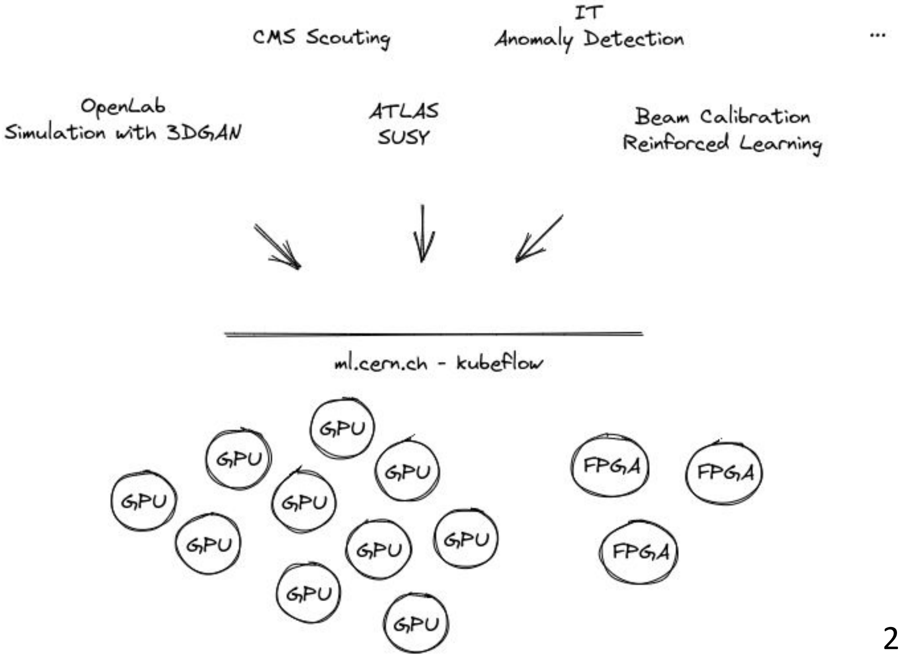
- Reduce maintenance work for researchers
- Easier access to GPUs
- Scaling capabilities

On-premise cluster, using Openstack

Integration with CERN services

- SSO, Harbor registry, CSI, Gitlab CI, EOS fs

In production since April 2021



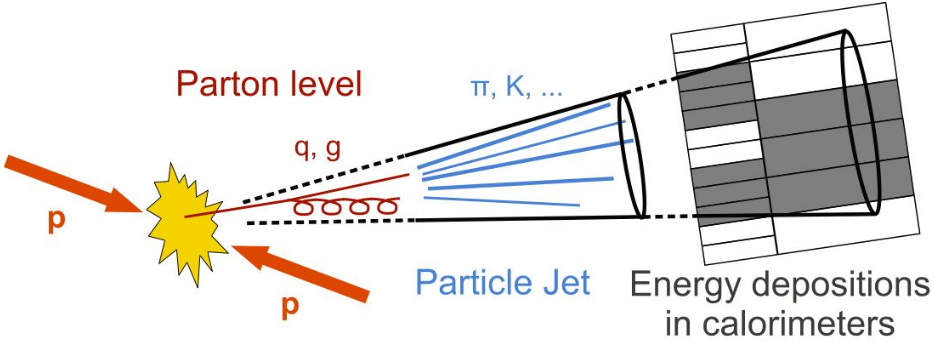
# Jet Energy Corrections

Colliding **protons** at high energies produces color-charged **partons**

Hadronization gives rise to a spray of color-neutral particles that are clustered into a **jet**

**Measured energy differs from theory** due to detector inaccuracies, invisible particles etc.

Can machine learning help with energy calibration?

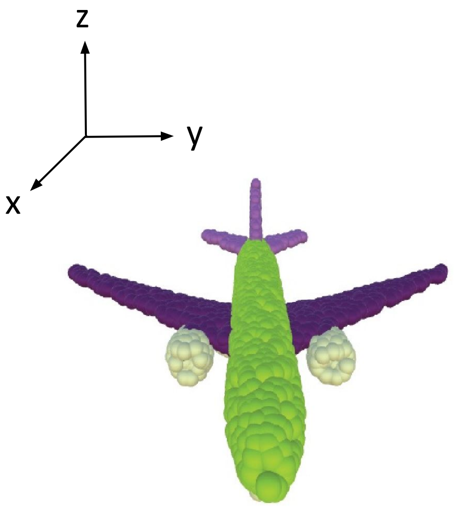


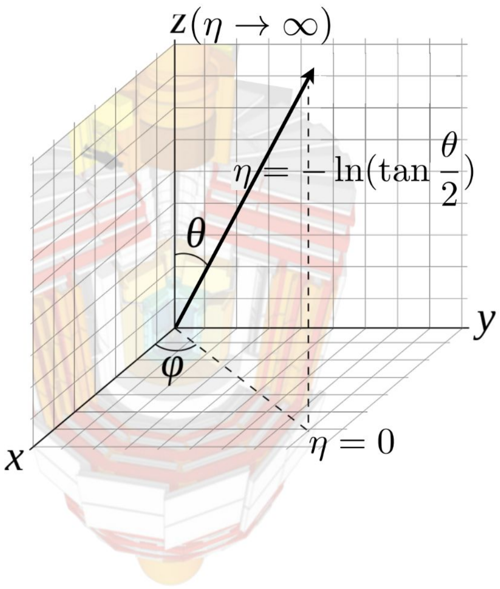

https://cms.cern/news/jets-cms-and-determination-their-energy-scale

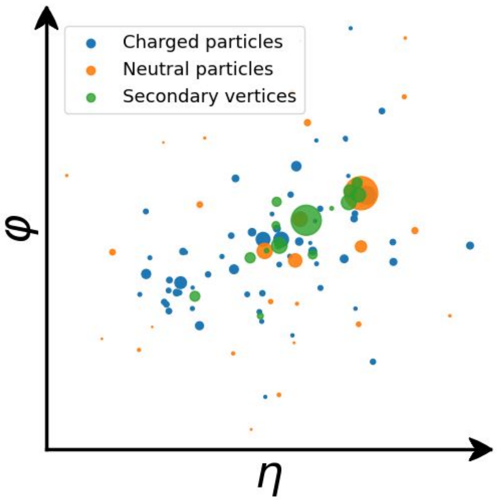
# Representing Jets as Particle Clouds

Use detector coordinates to represent jets as **particle clouds**

Analogous to **point clouds** in computer vision problems

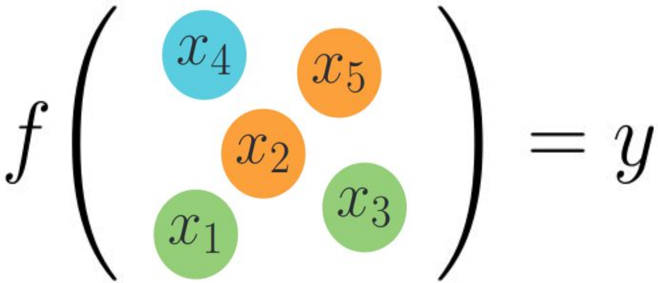


# Learning on Particle Clouds

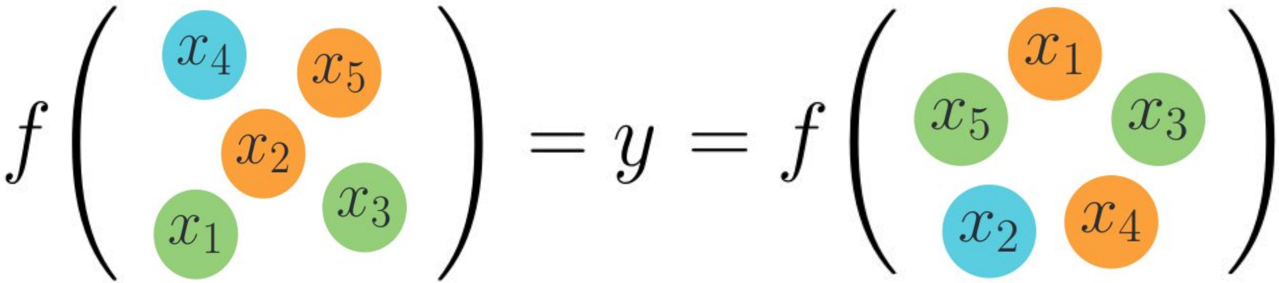
Map set of particle feature vectors $x_i$ towards correction factor target $y$



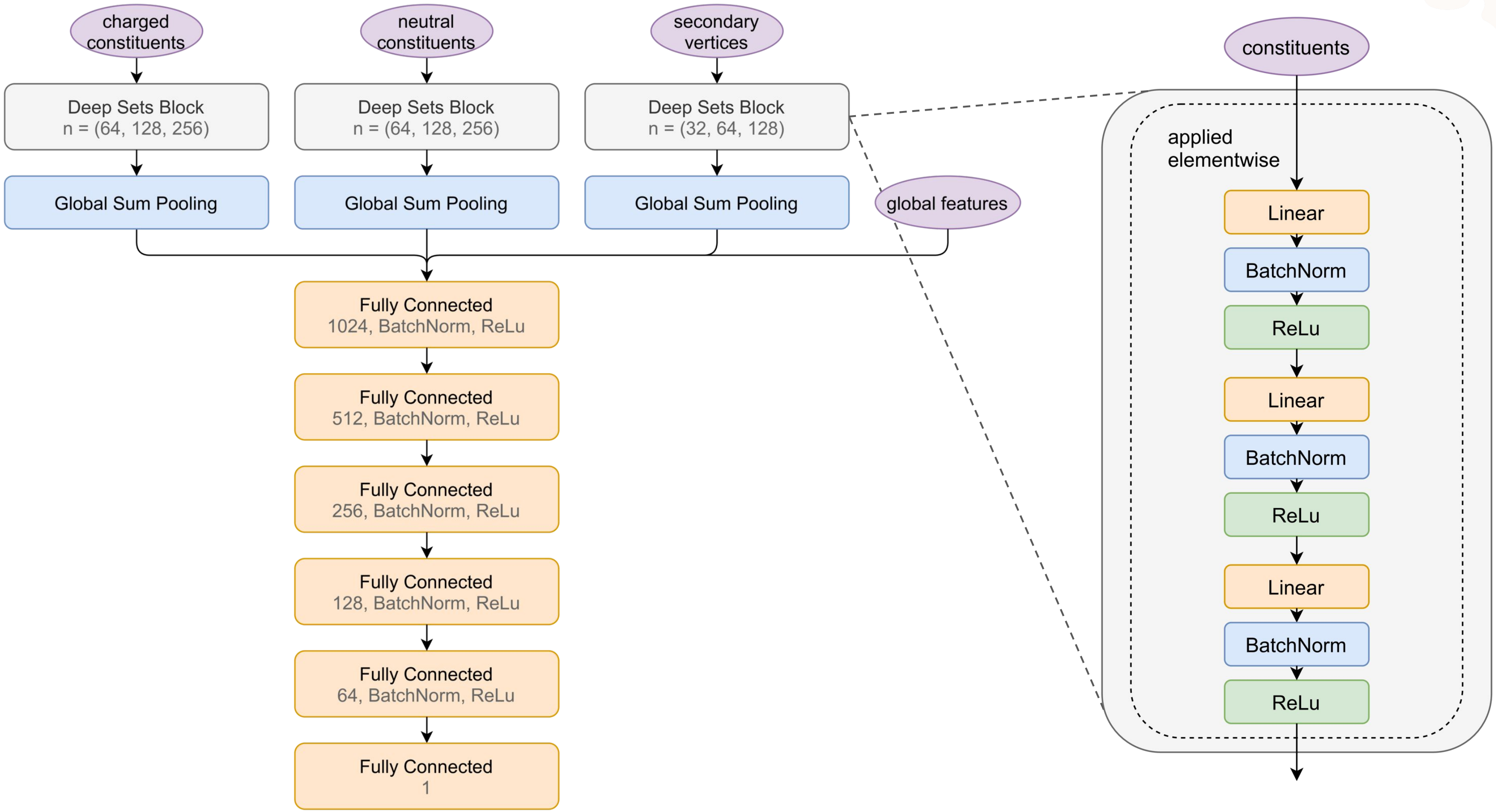
# Learning on Particle Clouds

Map set of particle feature vectors $x_i$ towards correction factor target $y$

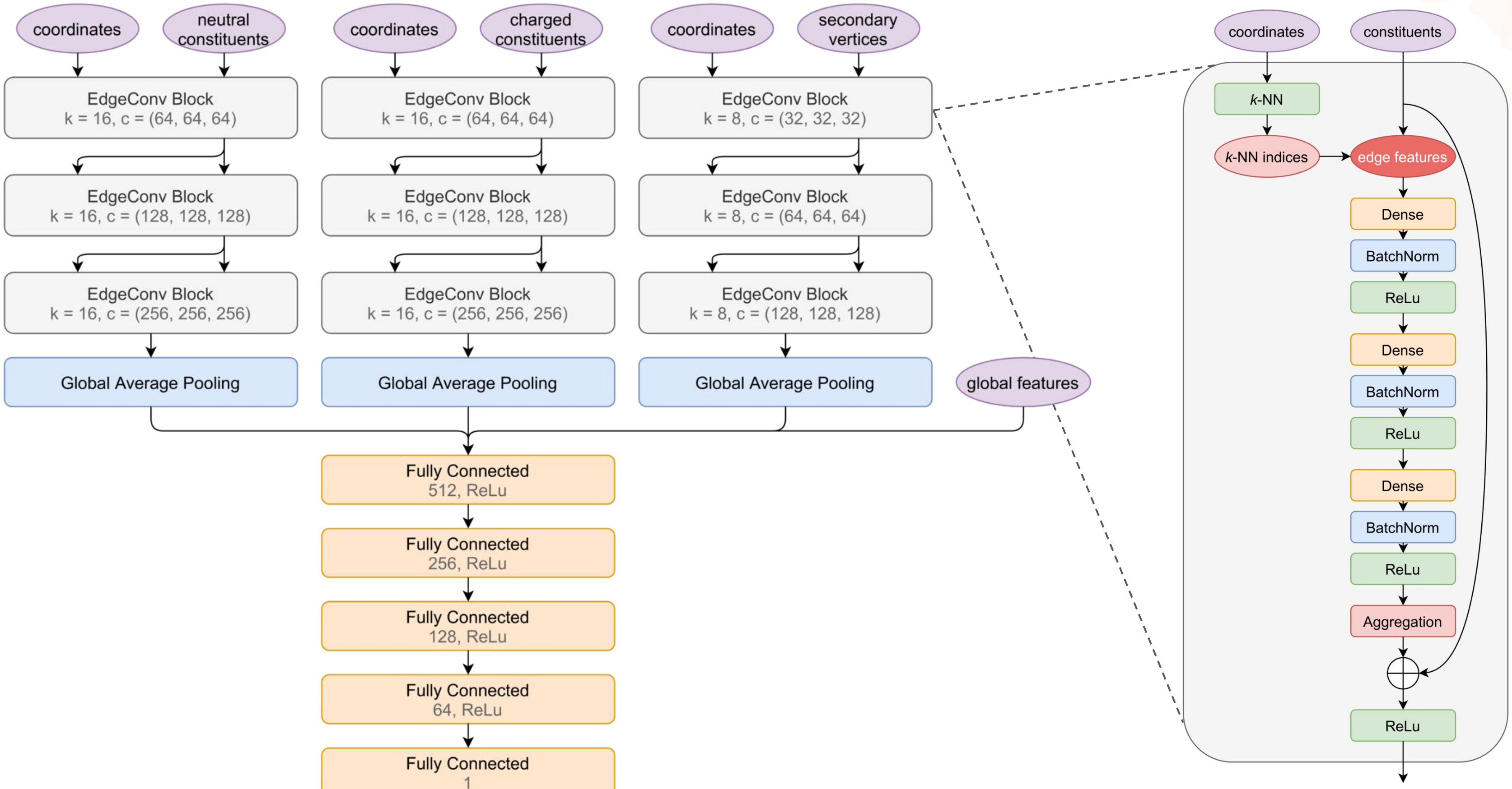
The model must be invariant to the order of the jet constituents



# Particle Flow Network



# ParticleNet

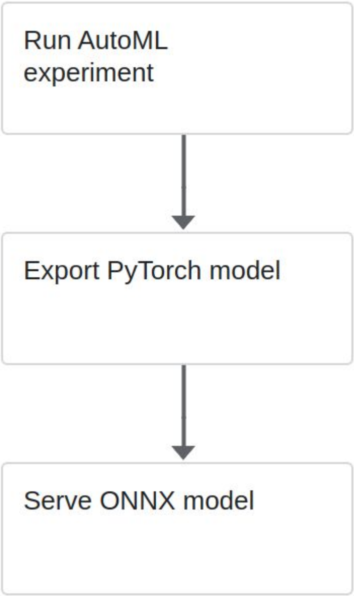


# ML Pipeline

Kubeflow Pipelines: “Engine for scheduling multi-step ML workflows”

Define end-to-end ML pipeline as a directed graph

- Start with running a Katib AutoML experiment
- Export the optimal model
- Finally serve using KServe



# Training

Dataset with 14 million jets = 10GB stored on S3

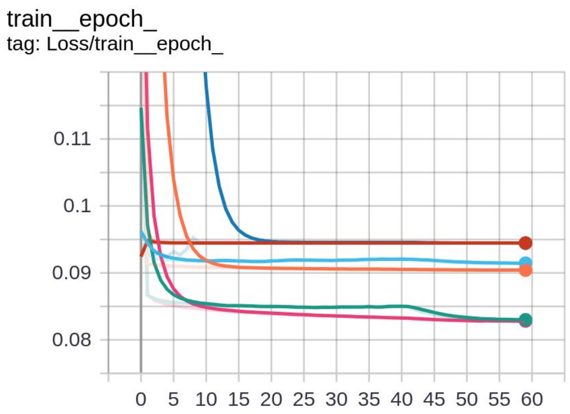
Minimize mean absolute error (MAE) loss

Tune hyperparameters with Katib using Random Search to reach a lower loss

Scalability

- Multi-node training by using the PyTorchJob operator
- Multiple CPU workers can read data simultaneously
- Additionally, many Katib trials can be run in parallel

Monitor training with Tensorboard component



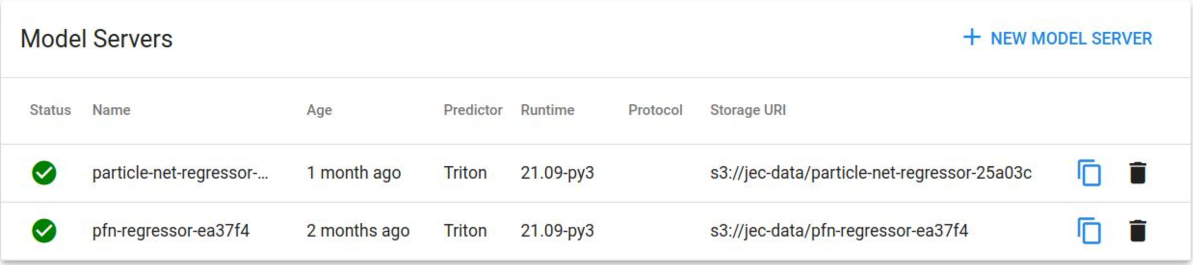
# Inference

Export best PyTorch model to [ONNX]

Serve model with [Nvidia Triton inference server]

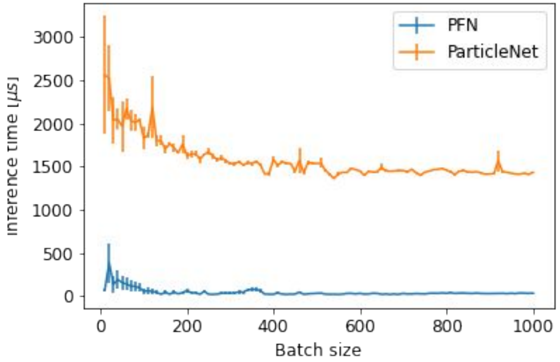
Use Triton's Python client to request predictions and get usage statistics

Analyze inference time and plot physics result in a notebook server on Kubeflow


Model Servers

+ NEW MODEL SERVER

| Status | Name | Age | Predictor | Runtime | Protocol | Storage URI |
|---|---|---|---|---|---|---|
| | particle-net-regressor-... | 1 month ago | Triton | 21.09-py3 | | s3://jec-data/particle-net-regressor-25a03c |
| | pfn-regressor-ea37f4 | 2 months ago | Triton | 21.09-py3 | | s3://jec-data/pfn-regressor-ea37f4 |



# Demo

# Conclusions

ML can provide **significant improvements** in high energy physics use cases

Jet energy regression example

- Energy resolution improved by 10%
- Flavor dependence improved by factor of 3

Kubeflow greatly **facilitates the scalability** of large-scale workloads

- Excellent mutual integration of components (Pipelines, AutoML, operators, KServe)
- Customizable and reproducible environments
- Predictor pods scale up to support concurrent inference requests

# Material

Kubeflow Pipeline

https://gitlab.cern.ch/dholmber/jec-pipeline

Kubeflow inference

https://gitlab.cern.ch/dholmber/jec-inference

CERN Kubeflow docs

https://ml.docs.cern.ch

KubeCon Recording

https://youtu.be/iqbsbXZDjs8

# Material

JEC with GNN thesis

https://helda.helsinki.fi/handle/10138/344118

Thesis code

https://gitlab.cern.ch/dholmber/jec-gnn

https://gitlab.cern.ch/dholmber/ml-jec-vars

Learning to Discover

https://indico.ijclab.in2p3.fr/event/5999/timetable/#32-jet-energy-corrections-with

**Thank you for the attention!**

**Questions?**

# Particle Flow Network [arXiv:1810.05165]

An MLP $\phi$ is applied to every particle $x_i$

$$\mathbf{h}_i = \phi(\mathbf{x}_i)$$

Aggregate latent features $h_i$ using sum pooling (order invariant operation)

Feed into another MLP $\rho$ mapping to the regression target

$$f(\mathbf{X}) = \rho\left(\sum_{i \in \mathcal{V}} \phi(\mathbf{x}_i)\right)$$

# ParticleNet [arXiv:1902.08570]

Initial graph in ($\eta$, $\varphi$) space — updated after each edge convolution

Local patch for every particle using *k*-nearest neighbors

Define edge features for each center-neighbor pair

$$\mathbf{e}_{ij} = \psi(\mathbf{x}_i,\ \mathbf{x}_j)$$

Aggregate using average pooling and concatenate with skip connection

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \frac{1}{k}\sum_{j\in\mathcal{N}_i^k}\psi(\mathbf{x}_i,\ \mathbf{x}_j)\right)$$

Pool outputs and feed into another MLP mapping to the target

$$f(\mathbf{X},\mathbf{A}) = \rho\left(\frac{1}{n}\sum_{i\in\mathcal{V}_i^n}\phi(\mathbf{x}_i,\mathbf{X}_{\mathcal{N}_i^k})\right)$$