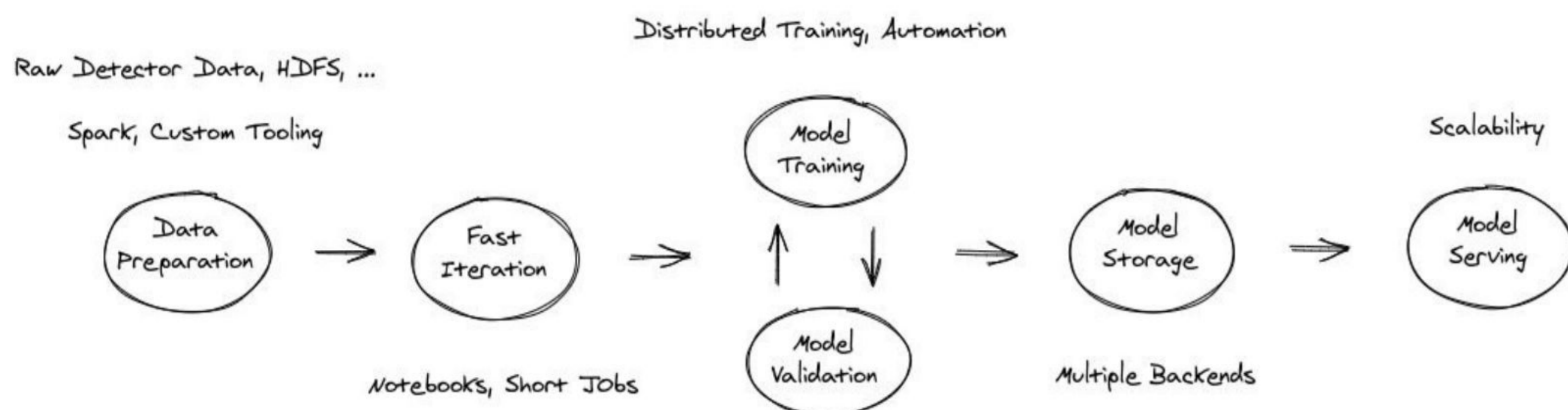# MLOps for High Energy Physics

Daniel Holmberg, *University of Helsinki*

Principles of MLOps

ML workflow organized as a directed acyclic graph

Experiments are reproducible

Versioning of data, code and model for traceability
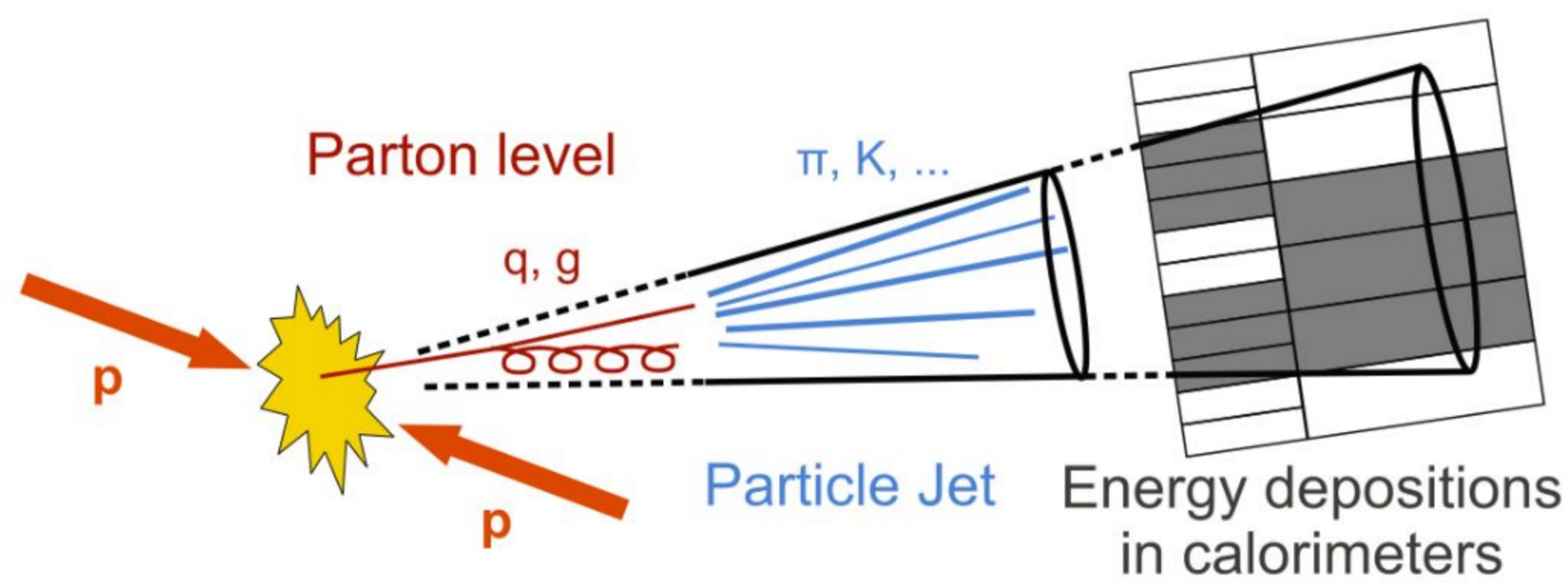
Continuously updateable model based on new features

Kubeflow at CERN

Centralized service for the whole ML lifecycle

Provide access to internal datacenter resources + potentially public cloud

Ensure efficient usage of on premises hardware accelerators
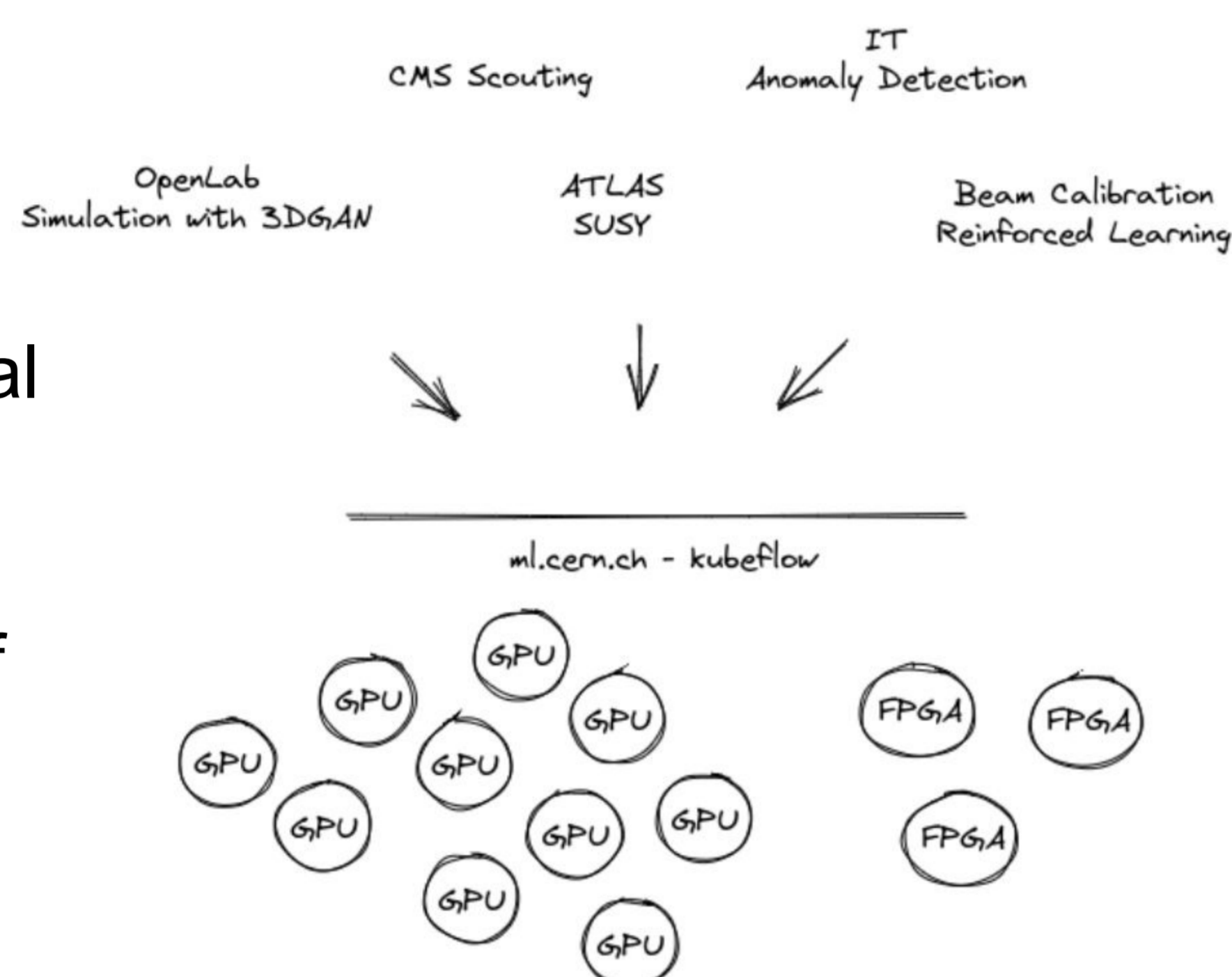
Example Usecase: Jet Energy Corrections

Use detector coordinates to represent jets as particle clouds

Analogous to point clouds in computer vision problems
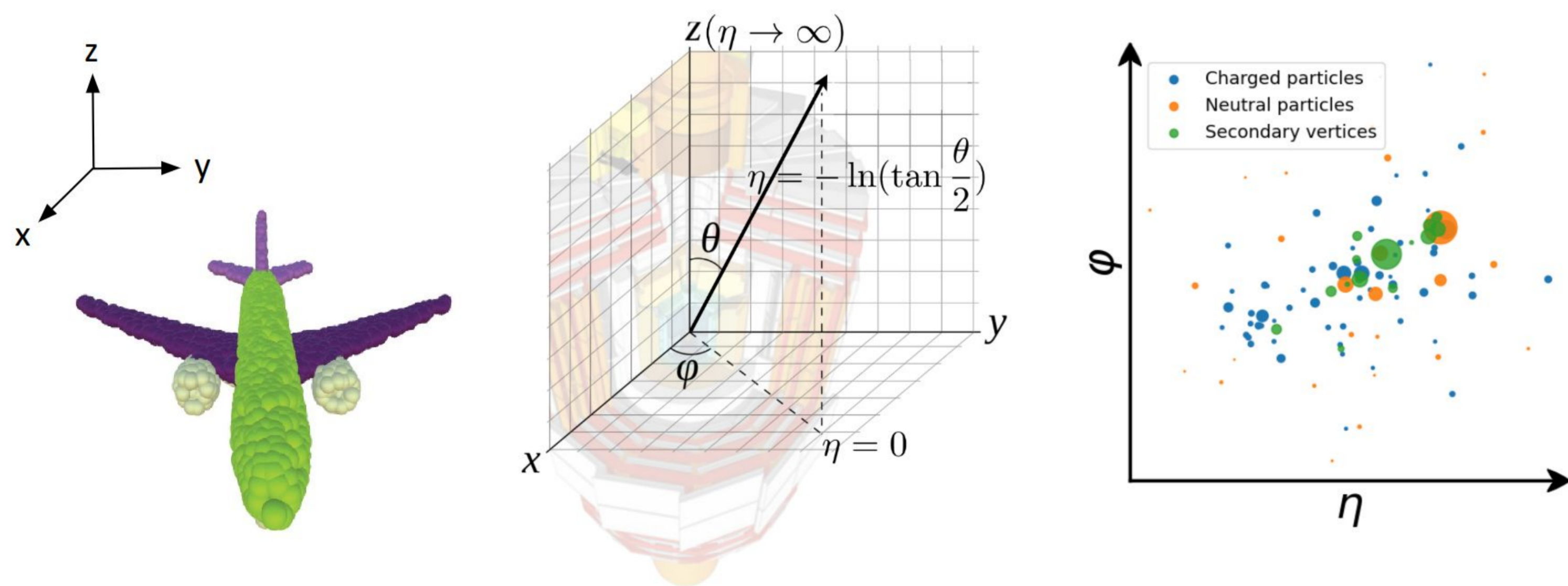
Map set of particle feature vectors x^i towards energy target y

$$f\left(\begin{array}{c} x_4\ x_5 \\ x_2\ x_3 \\ x_1 \end{array}\right) = y = f\left(\begin{array}{c} x_1 \\ x_5\ x_3 \\ x_2\ x_4 \end{array}\right)$$

The model must be invariant to the order of the particles

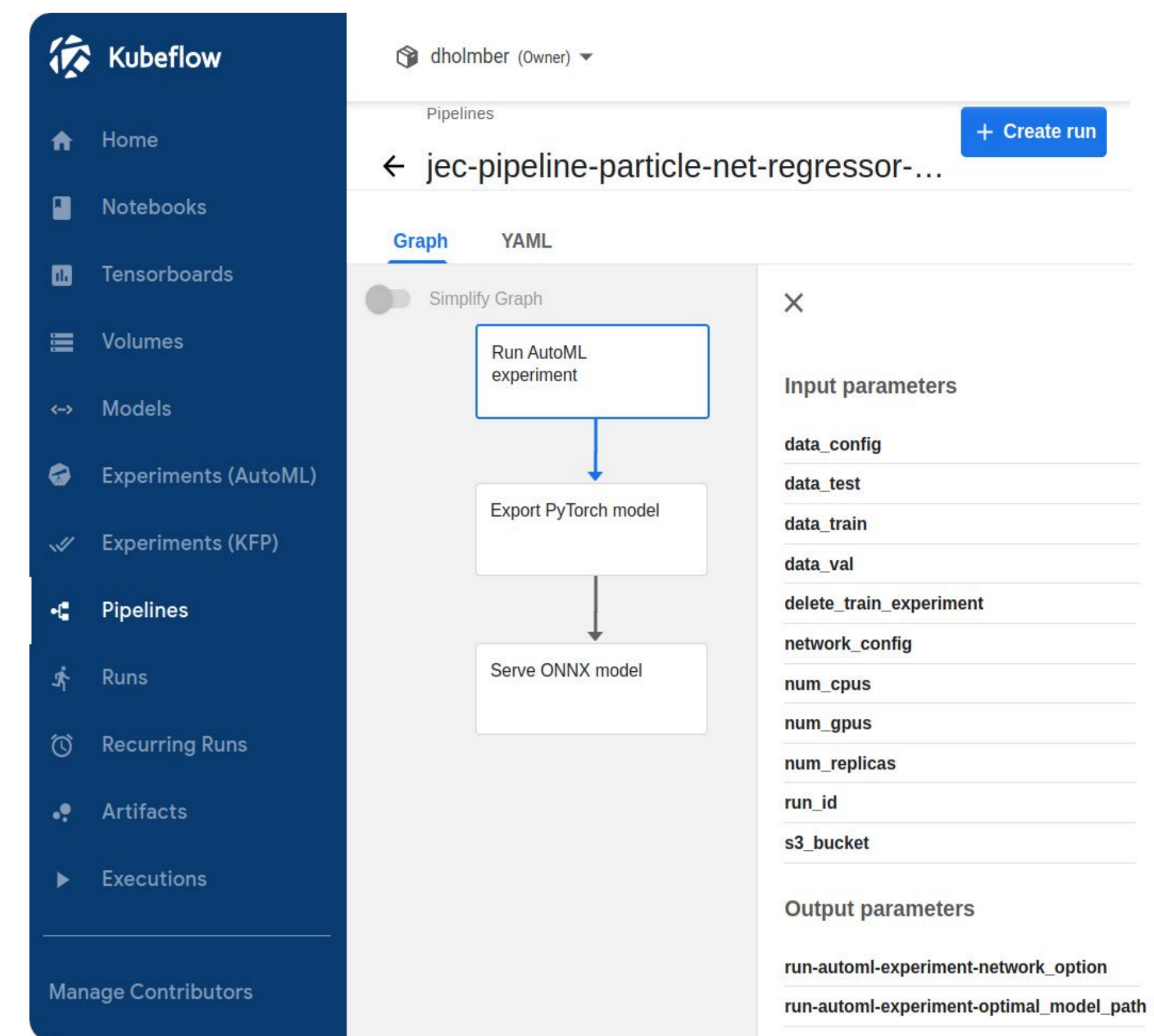Chose to try Particle Flow Network and ParticleNet

JEC Pipeline

Pack each pipeline step into containers stored on local registry

Each component specified using declarative yaml

Training with non-blocking GPU access

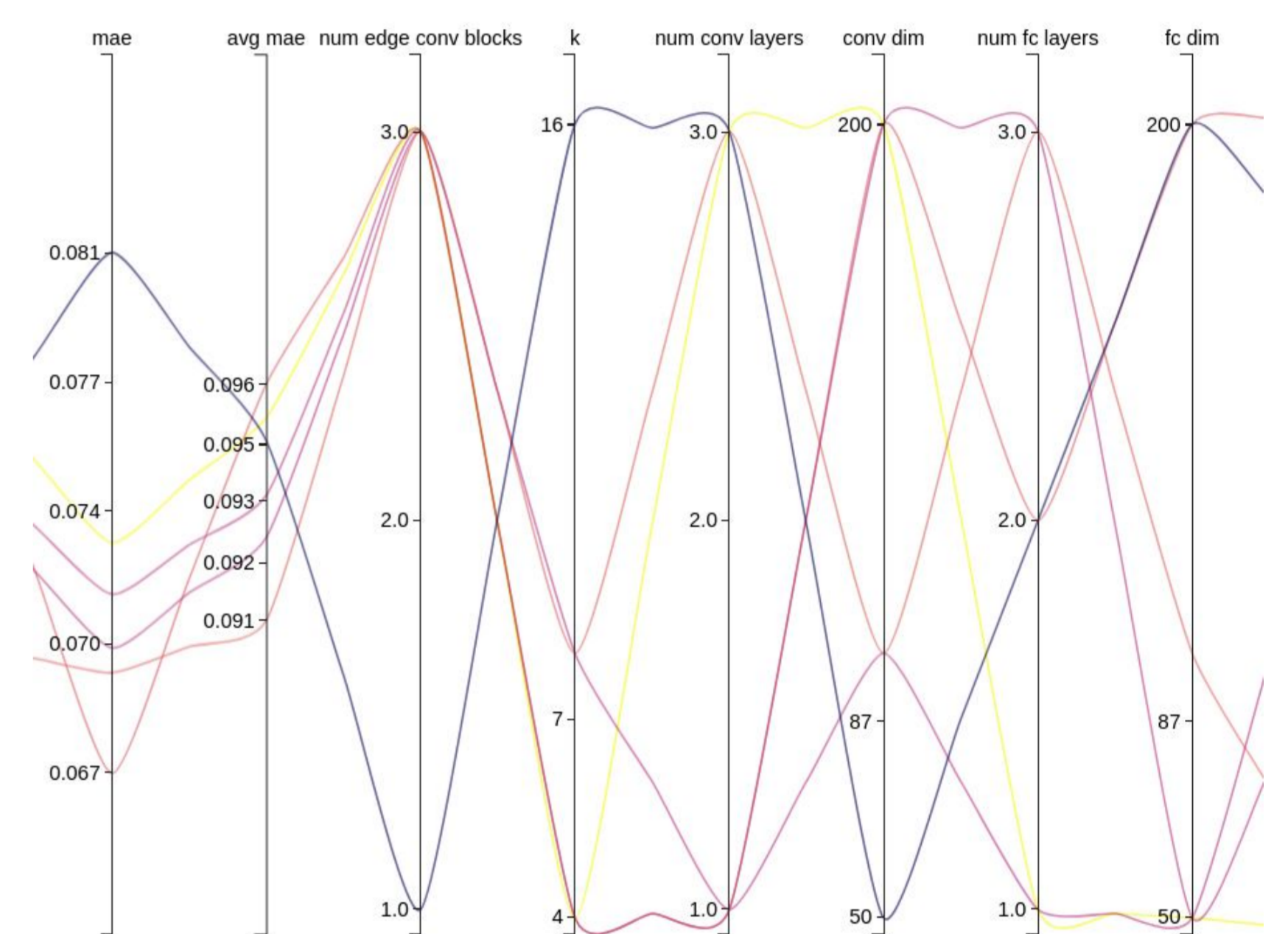Can be scheduled to run periodically

AutoML Experiment

Katib supports HP tuning, early stopping and neural architecture search

ML framework agnostic
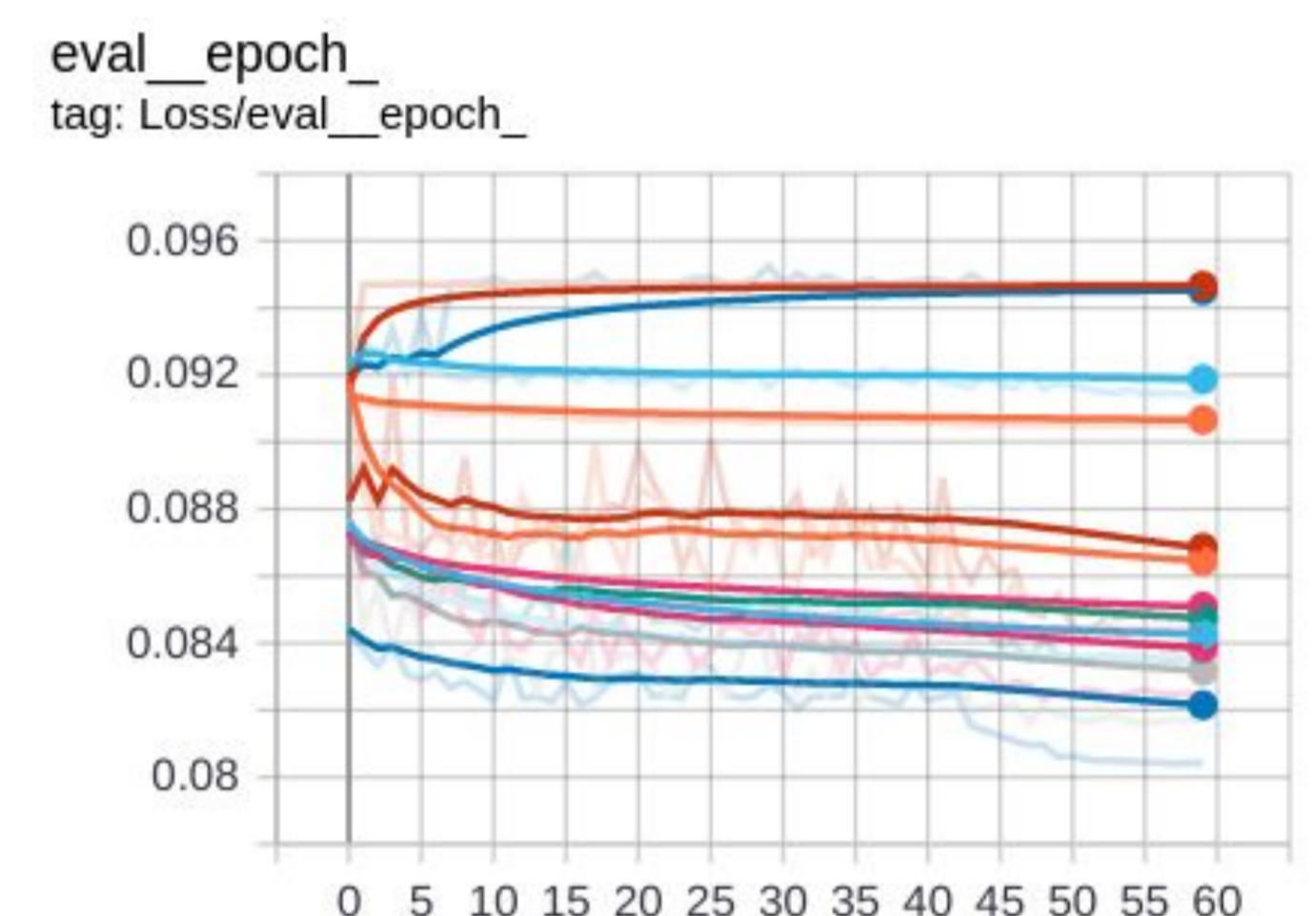
Parallelization of trials

Results visualized in UI

Monitor with Tensorboard

Live metrics logged to S3

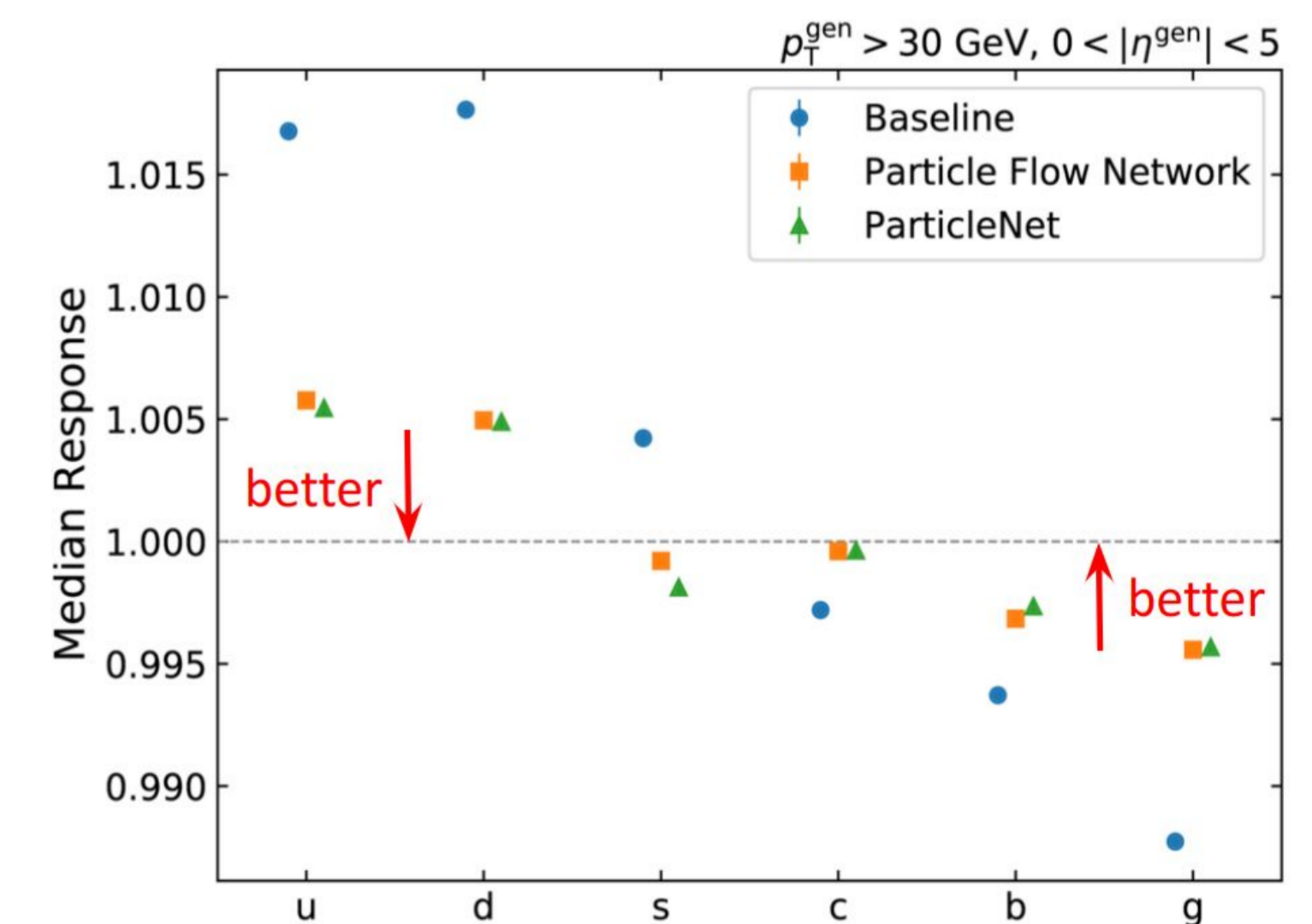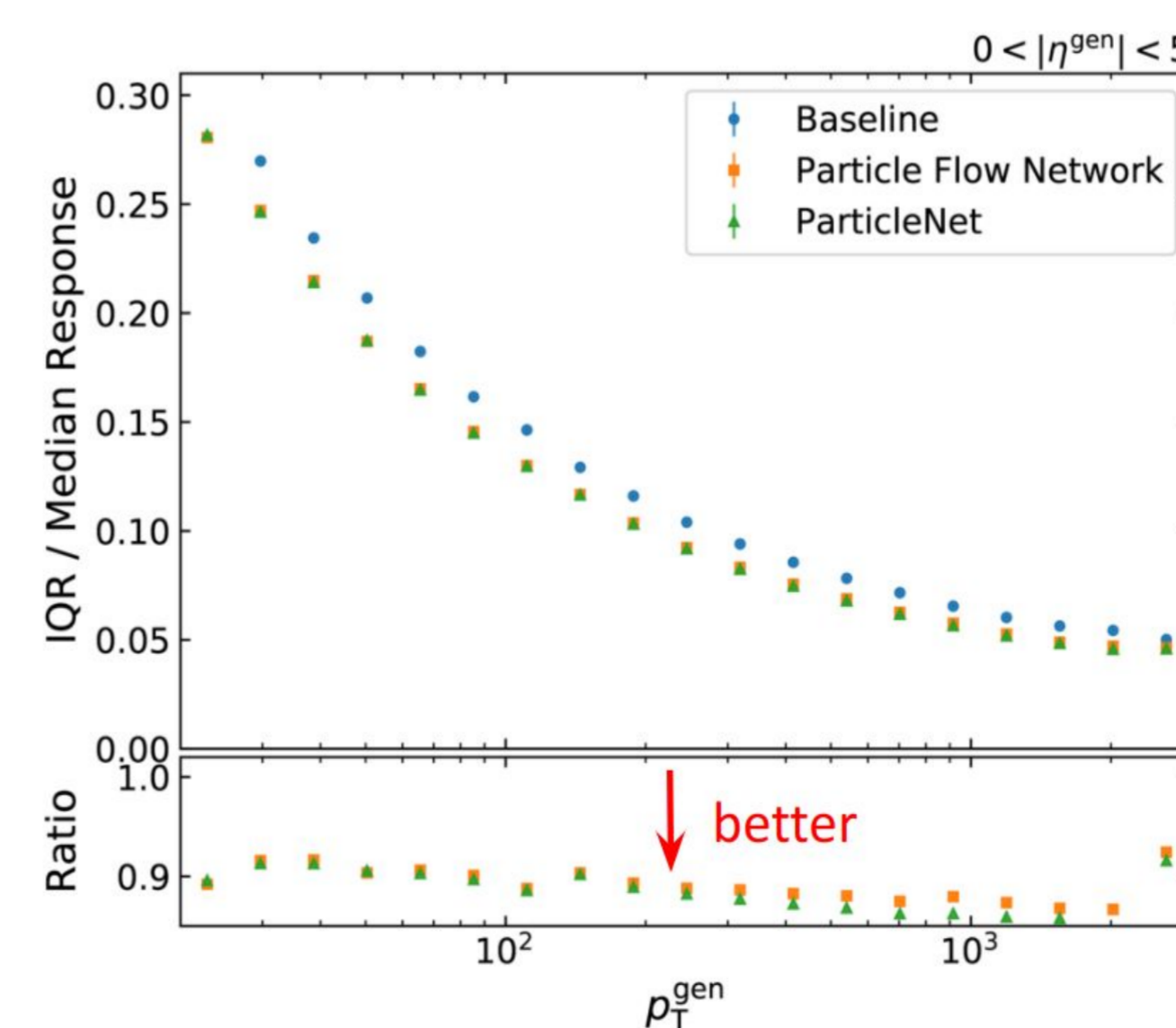Tensorboard pod can be pointed to object storage path and display training progress

Model Serving

Export PyTorch model to ONNX model from S3

Serve over http with Nvidia Triton

Pods autoscale to incoming traffic

Improvements in energy resolution and flavor dependence!

References

ML platform: https://ml.docs.cern.ch

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

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