AI and HPC
–
Distributed Hyperparameter Optimization using HPC systems

Eric Wulff [1,2], Maria Girone[1,2],
Juan Pablo García Amboage[1,2], Joosep Pata[3]
[1]CERN, [2]CoE RAISE WP4
[3]NICPB

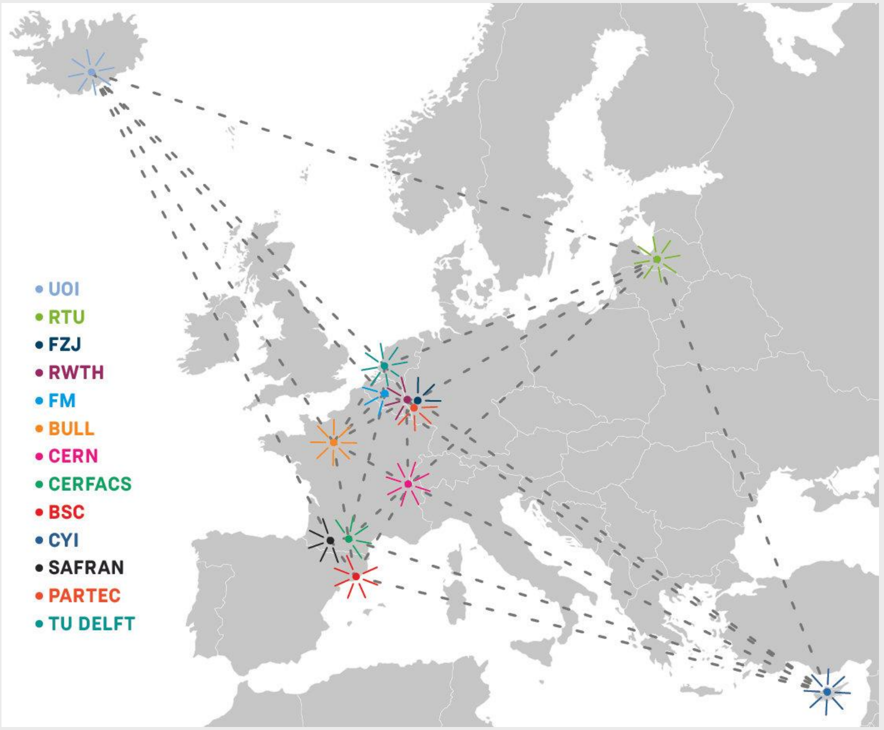
With material from the CMS Collaboration

Introduction

CoE RAISE

- CoE RAISE [1]: Center of Excellence for Research on AI- and Simulation-based Engineering at Exascale
  - Develop novel, scalable Artificial Intelligence technologies
  - Connect
    - hardware infrastructure
    - software infrastructure
    - compute-driven use cases
    - and data-driven use cases
- CERN (Dr. M. Girone) leads WP4: ***Data-Driven Use-Cases towards Exascale*** [2]
  - Including Task 4.1 (E. Wulff): *Event reconstruction and classification at the CERN HL-LHC*, which we'll see more details on later
- UOI (Prof. M. Riedel) leads WP2: ***AI- and HPC-Cross Methods at Exascale*** [3]
  - Provides expert support on HPC and AI methods to use cases in WP4

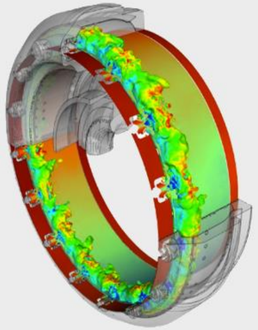
CoE RAISE Partners



[1] https://www.coe-raise.eu [2] https://www.coe-raise.eu/wp4 [3] https://www.coe-raise.eu/wp2

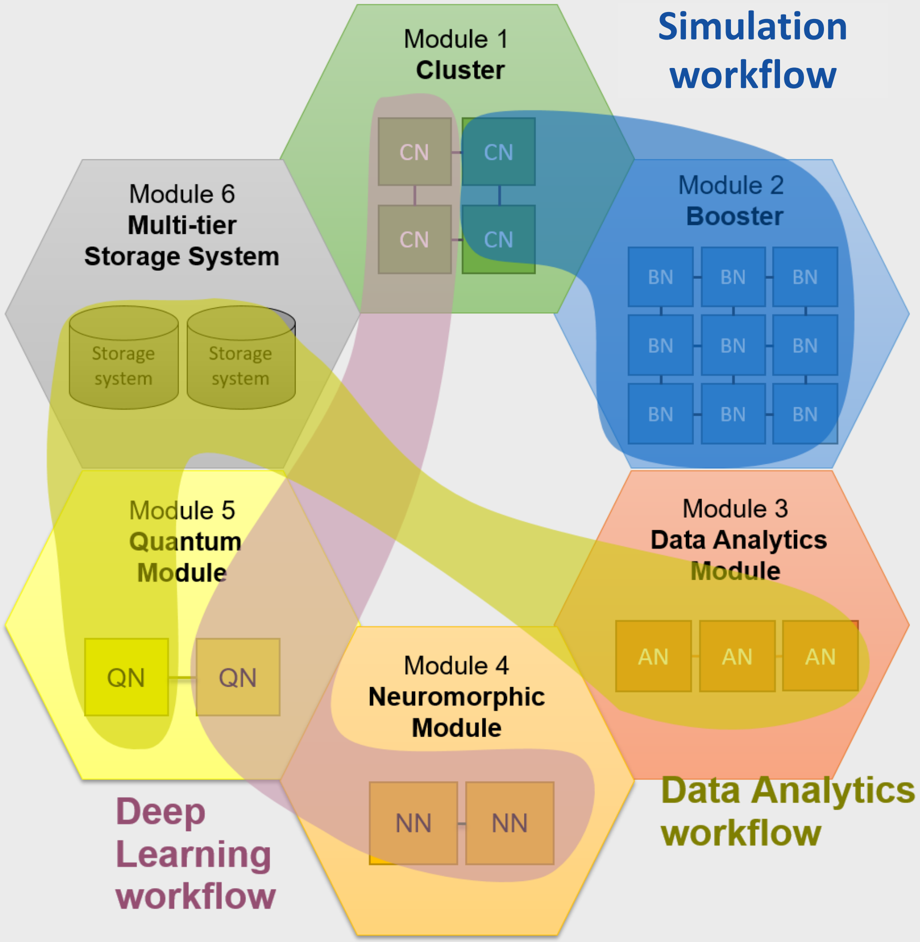
CoE RAISE: Modularity of Next-Generation HPC Systems

Complex hardware



Complex task

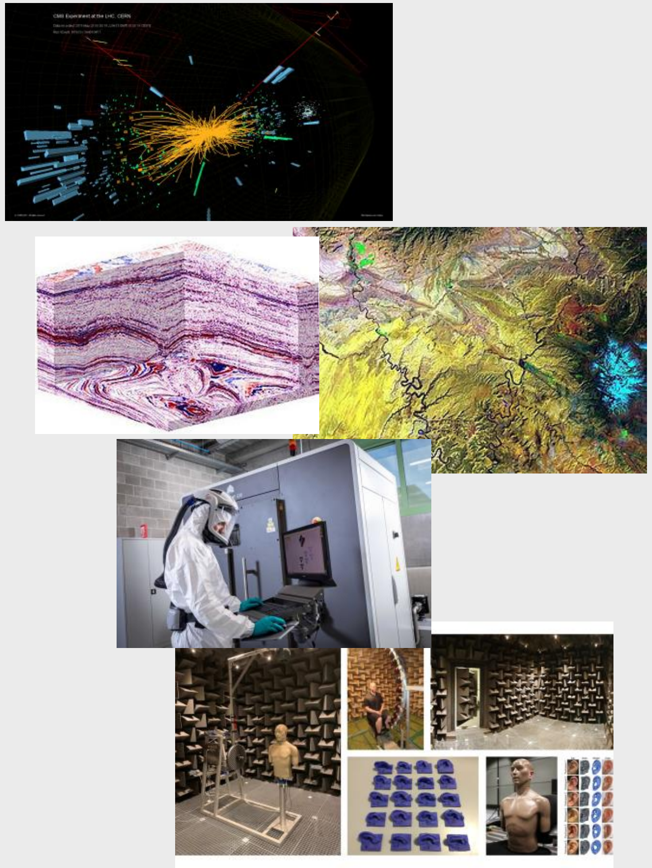
- Find the most suitable hardware for a specific task
- Enable intertwined AI- and HPC-workflows
- EuroHPC's roadmap includes integrating Quantum Computers and Quantum simulators in already existing supercomputer centers [1]



[1] https://eurohpc-ju.europa.eu/selection-six-sites-host-first-european-quantum-computers-2022-10-04_en

WP4 use-cases

- Representative use-cases from research and industry/SMEs, which have a strong focus on ***data-driven*** technologies, i.e., analyzing data-rich descriptions of physical phenomena
  - *Event reconstruction and classification at the CERN HL-LHC (CERN, RTU)*
    - develop novel approaches for HL-LHC collision event reconstruction replacing traditional algorithms with AI-driven techniques towards HPC-to-Exascale
  - *Seismic imaging with remote sensing for energy applications (FZJ, UOI, CYI)*
    - optimize seismic imaging and remote sensing, enabling AI approaches, combining satellite and airborne data with seismic imaging
  - *Defect-free metal additive manufacturing (UOI, FM)*
    - develop prediction models that detect porosity inside metal parts such that the information is exploited to improve the product quality in additive manufacturing
  - *Sound engineering (FZJ, UOI)*
    - develop a deep-learning-based algorithm that associates individual anatomy to a head-related transfer function (HRTF), for use in spatial audio systems

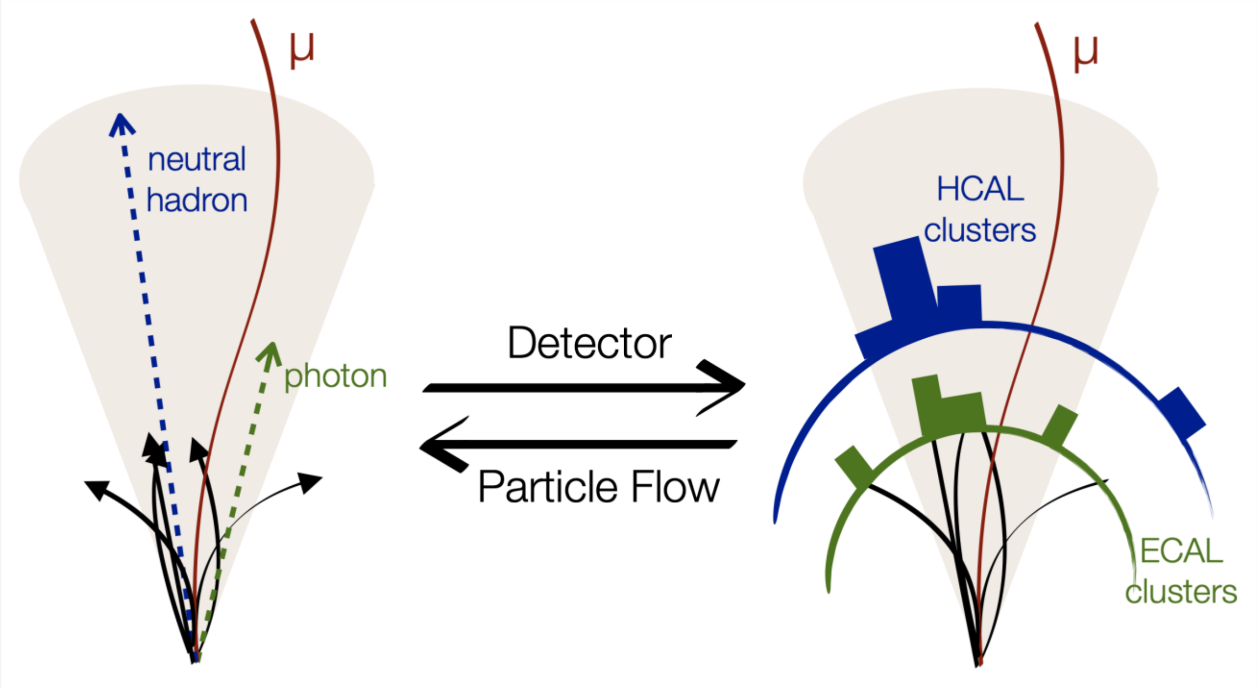


RAISE example use-case:

Event reconstruction and classification at the CERN HL-LHC

Event reconstruction at the LHC

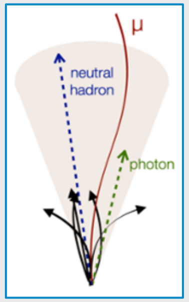
- Event reconstruction attempts to solve the inverse problem of particle-detector interactions, i.e., going from detector signals back to the particles that gave rise to them
- Particle-flow (PF) reconstruction takes tracks and clusters of energy deposits as input and gives particle types and momenta as output



AI-based particle flow reconstruction workflow

RAISE Center of Excellence

Physics simulation

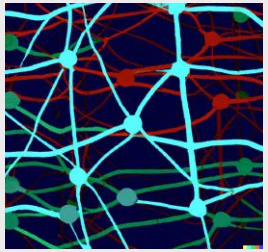


Data selection

Dataset creation

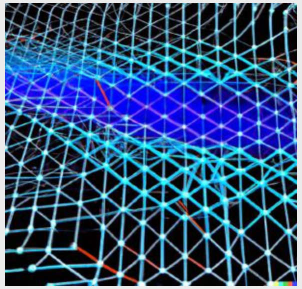
Data pre-processing

GNN training

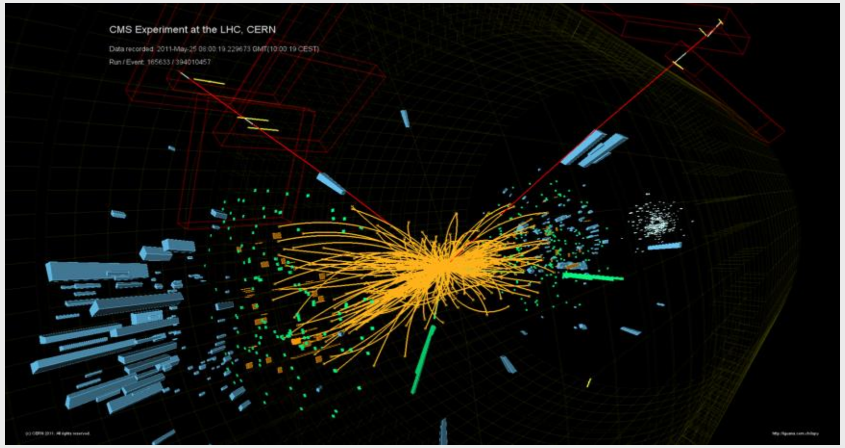


Model export

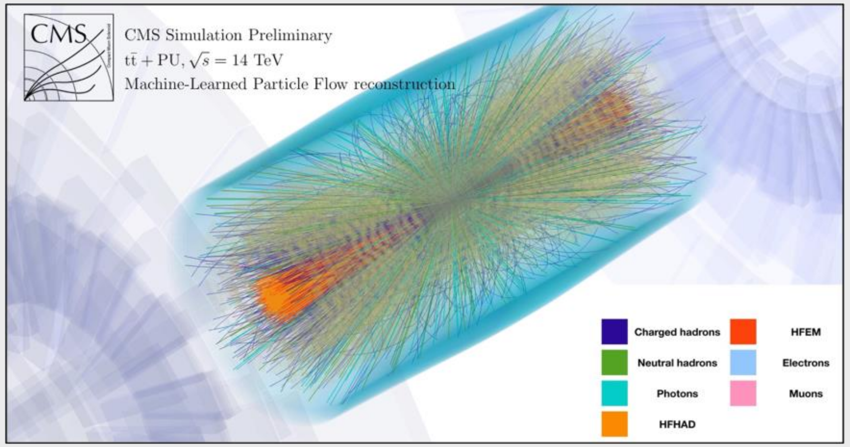
Trained model



Event reconstruction



CMS Collision event



MLPF event reconstruction [1]

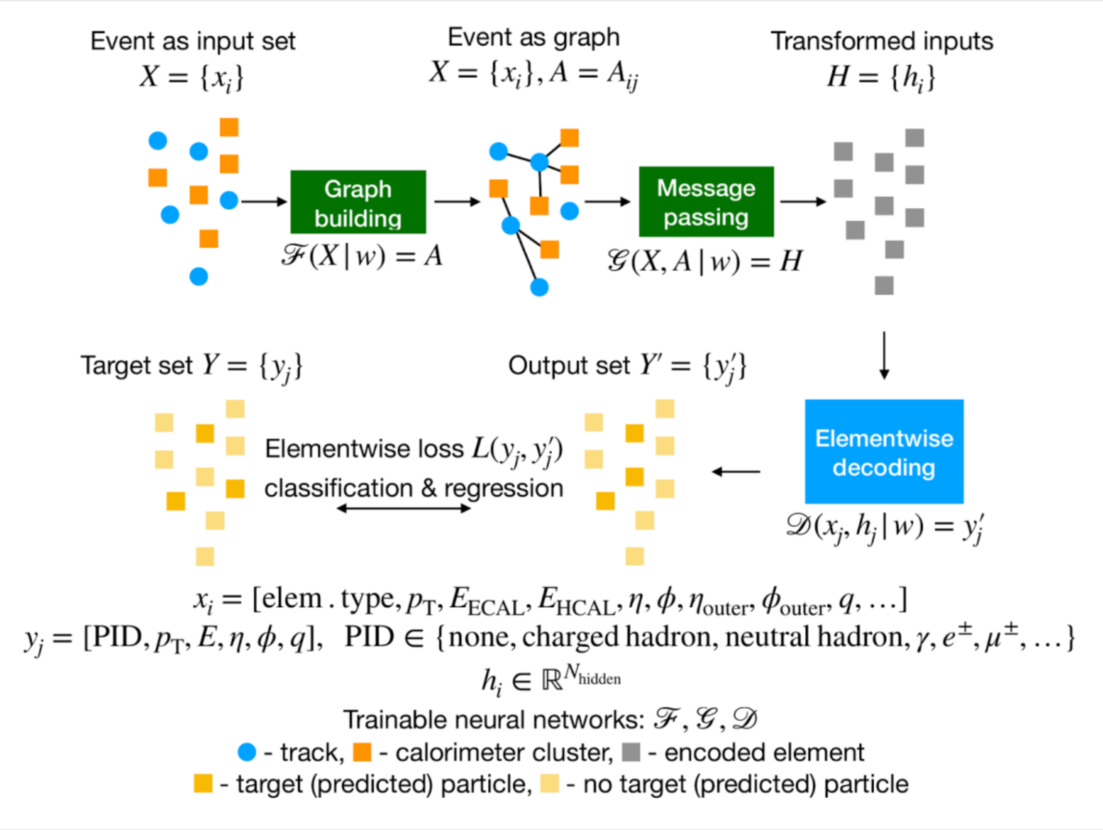
[1] Pata, J., Duarte, J., Mokhtar, F., Wulff, E., Yoo, J., Vlimant, J.-R., Pierini, M., Girone, M. (2022). *Machine Learning for Particle Flow Reconstruction at CMS*. Retrieved from http://arxiv.org/abs/2203.00330

Machine-Learned Particle-Flow (MLPF)

- The Particle Flow (PF) Algorithm [1]
  - Tries to identify and reconstruct all stable individual particles from collision events by combining information from different subdetectors (tracks, calorimeter clusters)

- Machine-Learned Particle-Flow (MLPF) [2]
  - GPU accelerated, GNN-based algorithm for PF
  - Code available on GitHub
  - ACAT2021 talk by J. Pata (and proceedings)
  - ACAT 2021 talk by E. Wulff (and proceedings)
  - ACAT2022 poster – latest results

The MLPF model



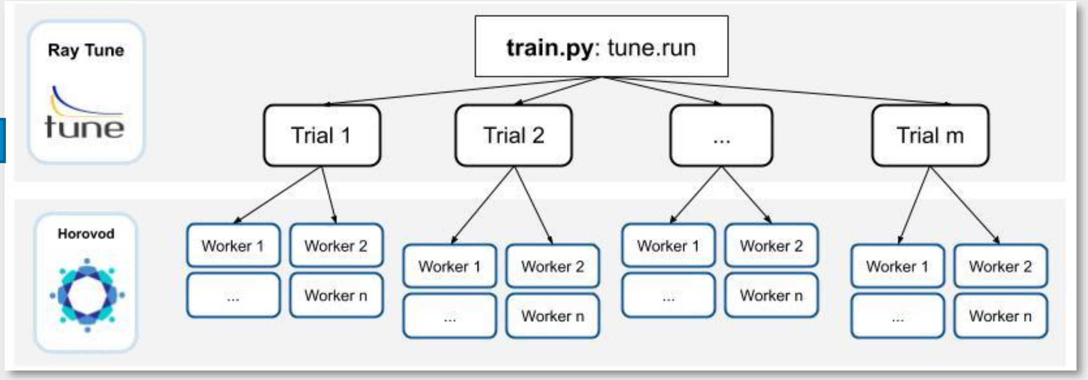
Based on Eur. Phys. J. C 81, 381 (2021)
https://arxiv.org/abs/2101.08578

[1] CMS Collaboration https://cds.cern.ch/record/1194487?ln=en

[2] Pata, J., Duarte, J., Vlimant, JR. *et al.* MLPF: efficient machine-learned particle-flow reconstruction using graph neural networks. *Eur. Phys. J. C* **81,** 381 (2021). https://doi.org/10.1140/epjc/s10052-021-09158-w

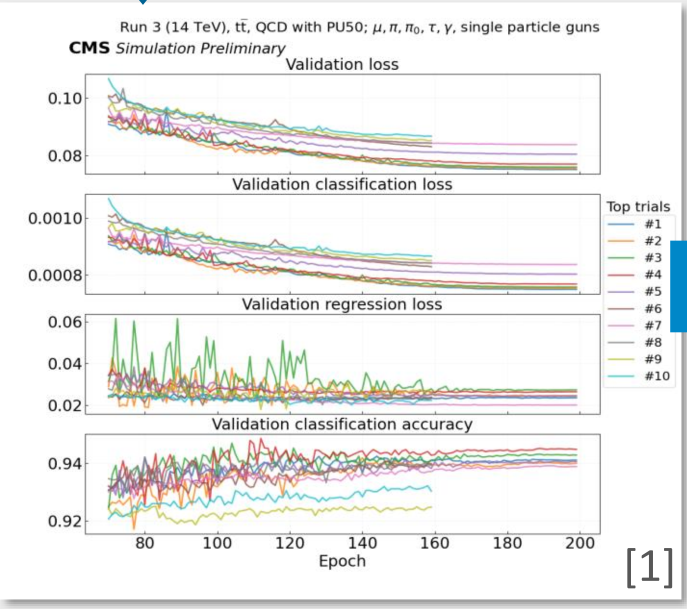
Large-scale distributed hyperparameter optimization (HPO)

Distributed HPO

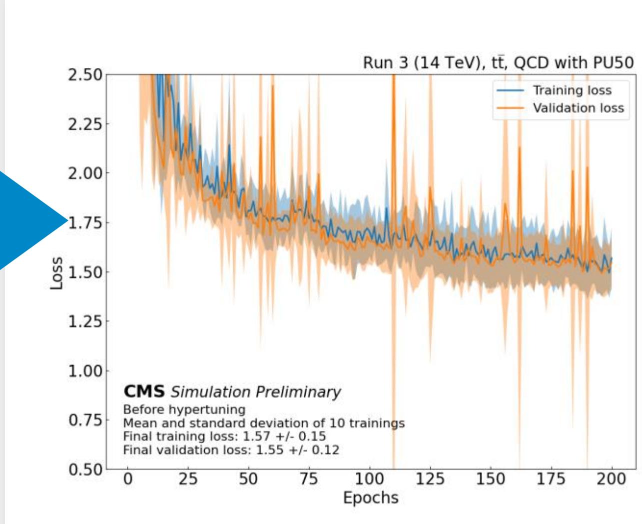


- 96 GPUs in parallel
- Using ASHA + Bayesian Optimization
- Scalable up to hundreds of GPUs
- Mean validation loss decreased by **~44%** giving a significant performance improvement

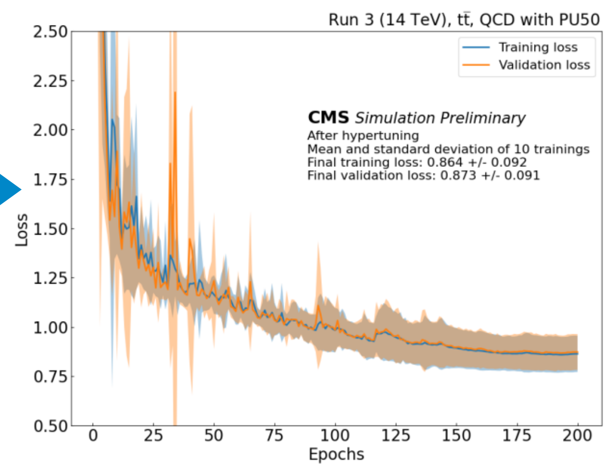
Model selection



Assess learning variability



Better learning

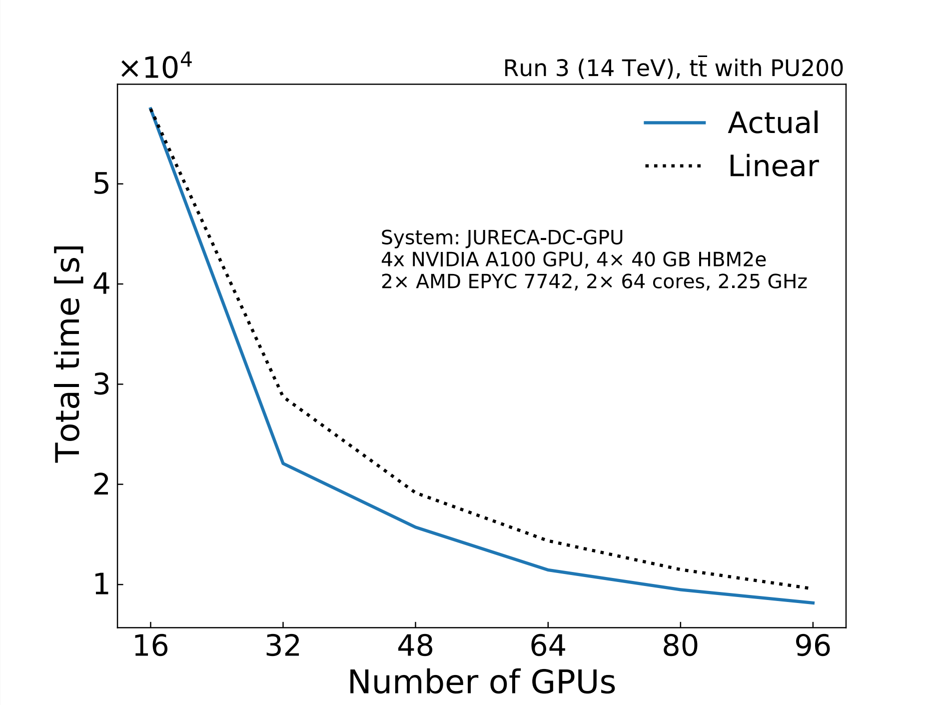


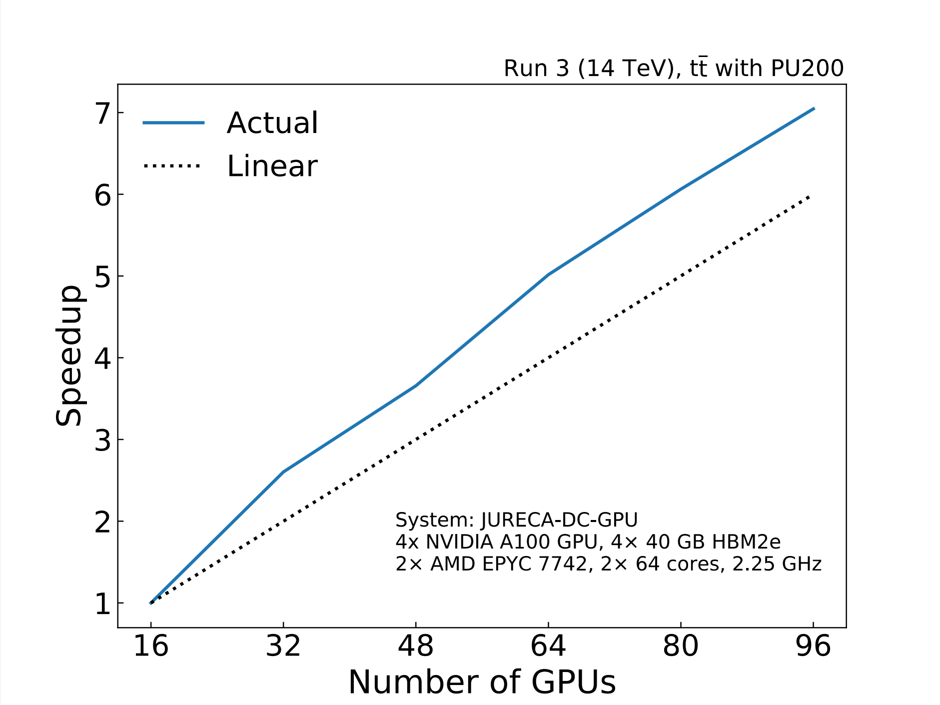
[1]

[1] E.Wulff, M. Girone, J. Pata https://doi.org/10.1088/1742-6596/2438/1/012092

Scaling of HPO of MLPF on multiple compute nodes

- Scaling of a HPO run of MLPF on the JURECA-DC-GPU system at the Jülich Supercomputer Centre (JSC), 4 NVIDIA A100 and 2× 64 cores AMD EPYC 7742 per node
- **Superlinear scaling** due to re-loading of models when using fewer nodes
- Using the ASHA algorithm to schedule and terminate trials, in combination with Bayesian optimization



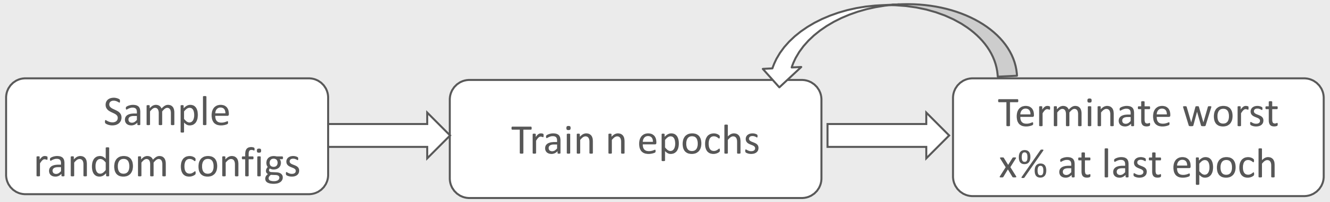


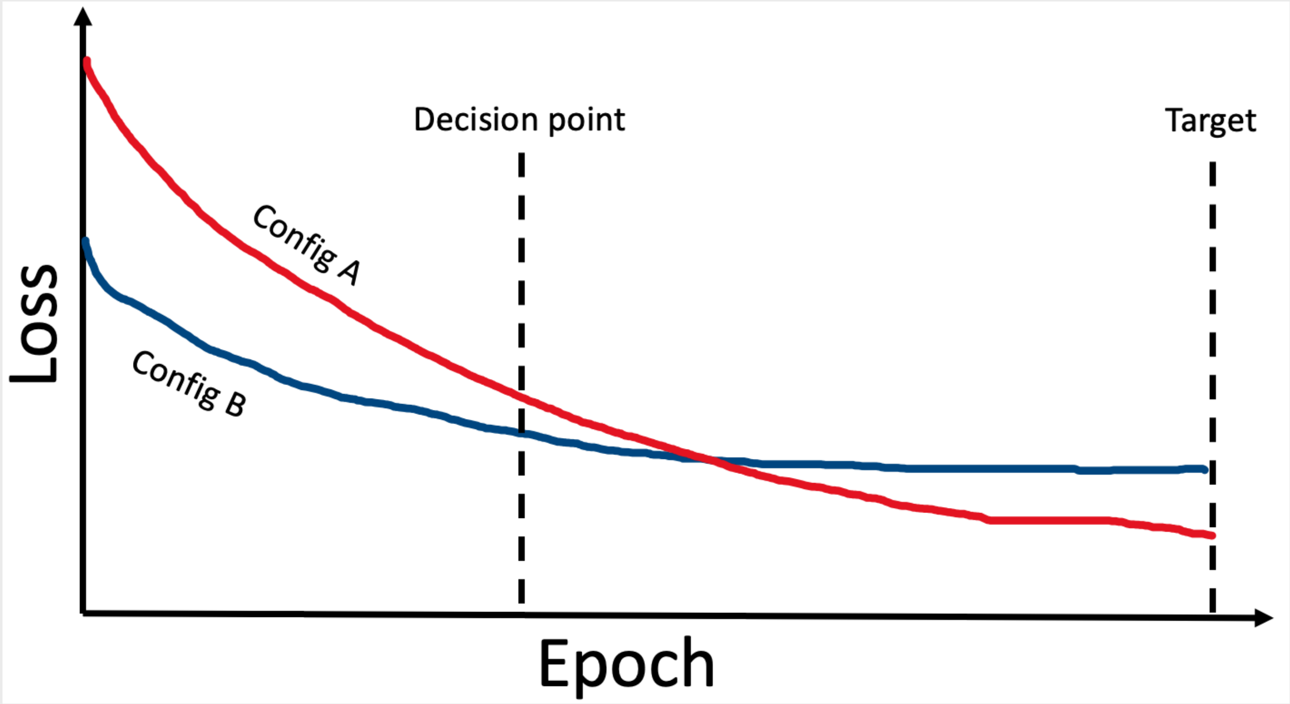
Data used: *Simulated particle-level events of ttbar and QCD with PU200 using Pythia8+Delphes3 for machine learned particle flow (MLPF),* https://doi.org/10.5281/zenodo.4559324

Quantum-SVR for model performance prediction in HPO

Model performance prediction using QSVR

- Current STOTA hypertuning algorithms rely on early stopping
- Stopping criterion: ranking according to a single metric (e.g., validation loss)
- Potential problem: loss curves are not linear
- Idea 1: Use a non-linear stopping criterion
  - For instance, an SVR model, inspired by [1]
- Idea 2: Use quantum computing to fit a Quantum-SVR (QSVR)

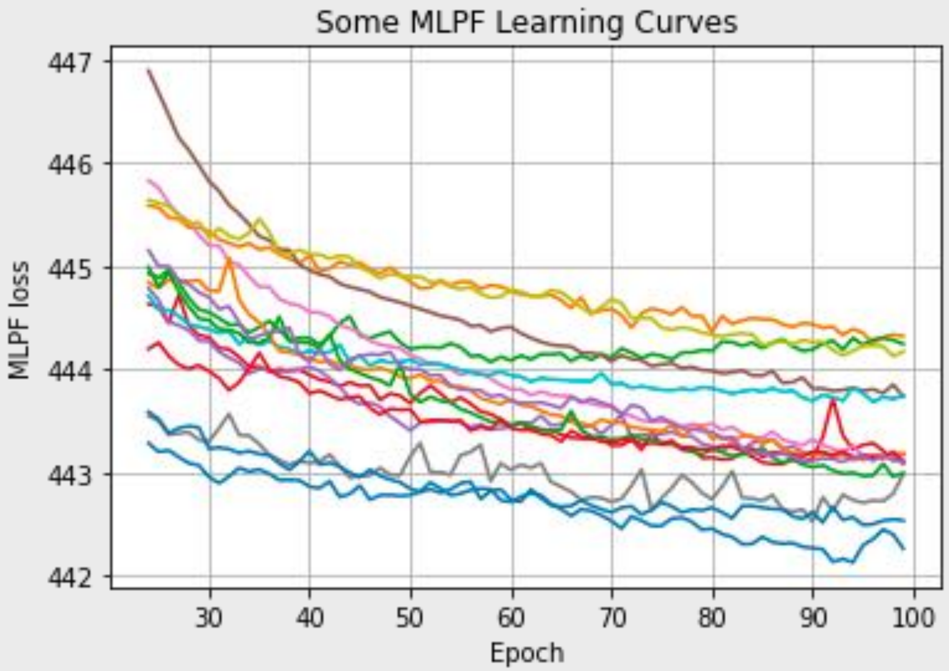




Dataset creation

RAISE Center of Excellence

- Generated dataset consisting of learning curves and HP configs
  - Run 300 MLPF trainings
  - For each training, sample HPs from a 7-dimensional search space
  - Train for 100 epochs on the publicly available Delphes dataset (https://doi.org/10.5281/zenodo.4559324)
- Inputs:
  - HP configuration
  - Partial learning curve
  - 1st and 2nd order differences of the partial learning curve
- Targets
  - Final validation loss



Accessing D-Wave Quantum Annealer in CoE RAISE

- A quantum annealer is a particular kind of quantum computer
  - Solves QUBO problems (Quadradic Unconstrained Binary Optimization)
- SVR can be formulated as a QUBO problem [1]
- The annealer returns multiple solutions
  - Quantum annealing is a stochastic process
- Challenges
  - We can only fit 20 training samples
  - Unstable results, quantum noise

Minimize: $$f(x) = \sum_{i<j}^{N} Q_{i,j} x_i x_j + \sum_{i}^{N} Q_{i,i} x_i$$

$x_i \in \{0,1\}$ and Q is a $N \times N$ symmetric matrix

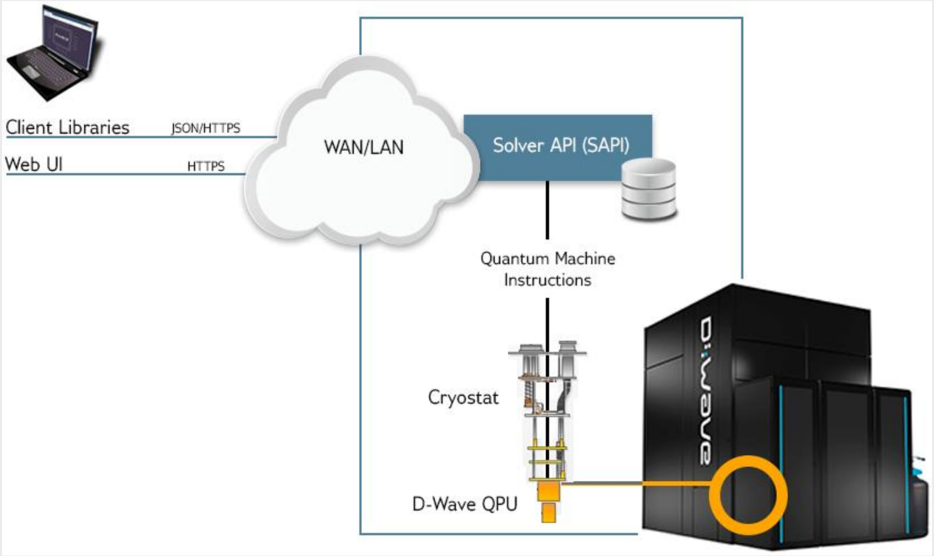


Image from D-Wave documentation

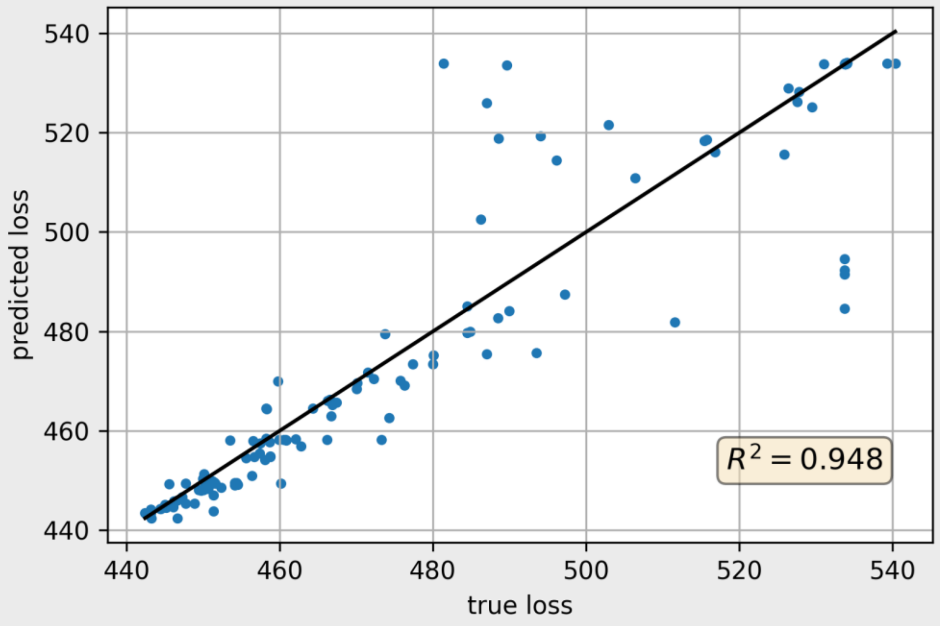
QSVR results

- Predicting final loss from fraction of loss curve (25%)
- QSVR results comparable to classical SVR and to simulated quantum annealing

R^2 scores

	Best	Worst	Mean	Std	Number of trainings
SVR	0.959	0.318	0.889	0.050	1000
Sim-QSVR	0.949	0.383	0.901	0.045	100
QSVR	0.948	0.742	0.880	0.056	10

Best QSVR results



Summary

Summary

- CoE RAISE develops novel, scalable AI methods towards Exascale
  - Use-cases from a wide range of sciences and industry
- Hyperparameter optimization could benefit any data-driven AI-based algorithm
- Large-scale distributed HPO significantly increased model performance in the example use-case of Machine-Learned Particle Flow (MLPF)
  - Would not have been possible without access to HPC resources
- The disruptive technology of Quantum Computing is already here and can be integrated in hybrid Quantum-HPC workflows
  - The technology is still very early-stage and is likely to improve greatly in the future

drive. enable. innovate.

RAISE Center of Excellence

The CoE RAISE project have received funding from
the European Union's Horizon 2020 –
Research and Innovation Framework Programme
H2020-INFRAEDI-2019-1 under grant agreement no. 951733

Follow us:

Backup

Hyperparameter Optimization

- Tune hyperparameters (HPs) to improve model

- HPs are not learned by gradient descent
  - Often stay constant during the learning process
  - Defines the model architecture (e.g., #layers, #nodes per layer, type of activation function, etc.)
  - Defines the learning algorithm (e.g., optimizer, learning rate, batch size, momentum, weight decay, dropout etc.)

- Can be automated using HPO algorithms
  - E.g., Hyperband, Bayesian Optimization

- HPO on complex models and large datasets is compute-resource intensive
  - Benefits greatly from HPC resources
  - In need of smart, efficient search algorithms

Hypertuning tool of choice: Ray Tune

- Open-source tool for multi-node distributed hyperparameter optimization
- Many built-in SOTA search algorithms
  - **ASHA/Hyperband**
  - **Bayesian Optimization**
  - Population Based Training
- Supports TensorFlow, PyTorch and others
- Supports integration of many other hypertuning tools such as **Scikit-Optimize**, **HyperOpt**, Optuna, SigOpt, etc.

Using Ray Tune on SLURM clusters

- Can be unintuitive when first setting up
- Ray expects a head-worker architecture with a single point of entry
  - We must start a head node and multiple worker nodes before running the Ray Tune training script on the head node
- Once properly set-up, works great

```
#!/bin/sh

#SBATCH ...
#SBATCH ...


# Get the node names
nodes=$(scontrol show hostnames $SLURM_JOB_NODELIST)
nodes_array=( $nodes )

# Get the head node
node_1=${nodes_array[0]}
ip=$(srun --nodes=1 --ntasks=1 -w $node_1 host ${node_1}i | awk '{ print $4 }') port=6379
ip_head=$ip:$port
export ip_head
echo "IP Head: $ip_head"

echo "STARTING HEAD at $node_1"
srun --nodes=1 --ntasks=1 -w $node_1 mlpf/raytune/start-head.sh $ip &
sleep 30

worker_num=$(($SLURM_JOB_NUM_NODES - 1)) #number of nodes other than the head node
for (( i=1; i<=$worker_num; i++ ))
do
  node_i=${nodes_array[$i]}
  echo "STARTING WORKER $i at $node_i"
  srun --nodes=1 --ntasks=1 -w ${node_i} mlpf/raytune/start-worker.sh $ip_head &
  sleep 5
done

# Run the Ray Tune script
python3 tune_script.py --cpus "${SLURM_CPUS_PER_TASK}" --gpus "${SLURM_GPUS_PER_TASK}"
exit
```

Hypertuning MLPF on HPC systems

RAISE Center of Excellence

- Thanks to Forschungszentrum Jülich (FZJ), San Diego Supercomputing Center (SDSC), Flatiron Institute (collaboration with CMS and CERN openlab)

- Using multiple compute nodes with 4 GPUs per node
  - Both systems: 4 NVIDIA A100 40GB per node
  - @CoreSite: 64 core Intel Icelake Platinum 8358
  - @JUWELS: 2x 24 core AMD EPYC Rome 7402

- We did 2 stages of hypertuning:
  - Using Ray Tune
  - BOHB [1] - Bayesian Optimization combined with Hyperband – using JUWELS Booster
  - ASHA [2] + Bayesian Optimization [3] – using CoreSite
  - ~76000 core-hours in total

- Back of the envelope calculation shows that it would have taken ~6 months on a single GPU instead of ~83 hours using HPC systems

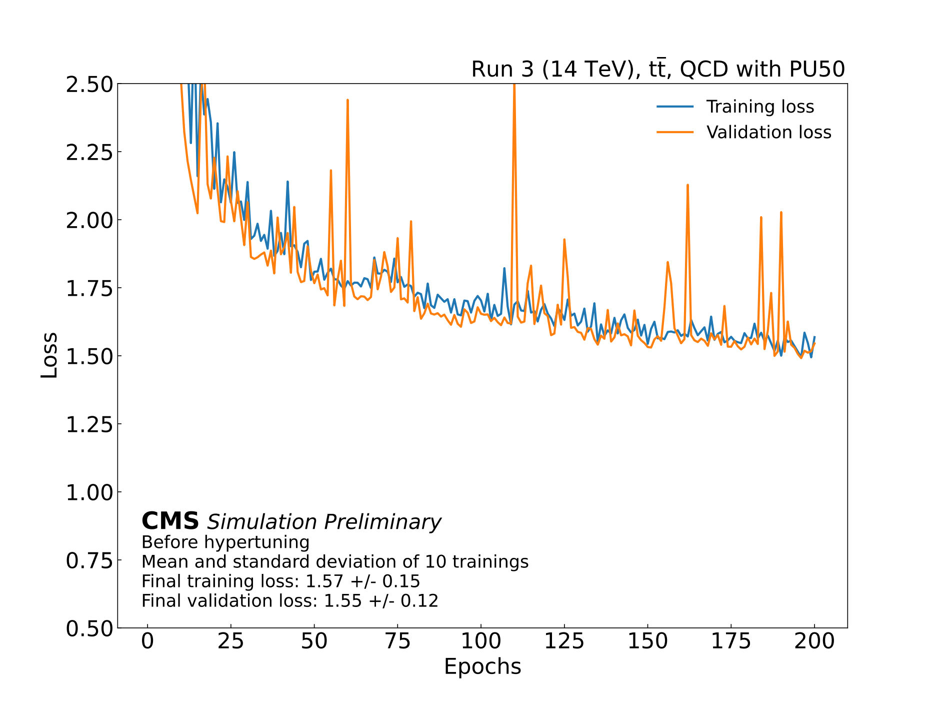
Run in part on the JUWELS Booster [2] Image: JÜLICH Forschungszentrum

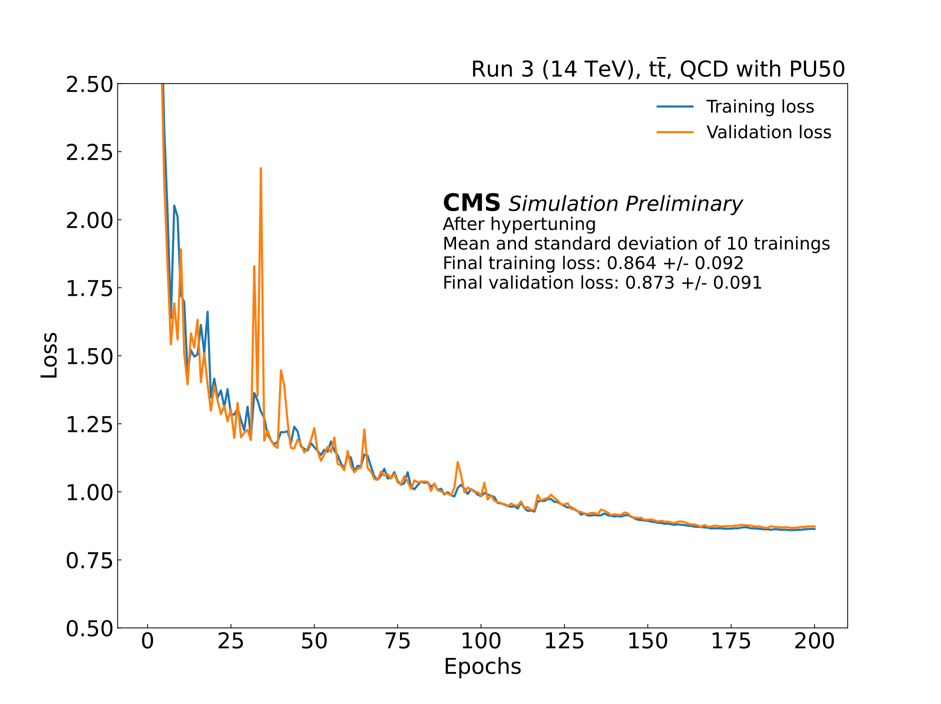
```
search_space = {
   "lr": loguniform(1e-4, 3e-2),
   "expdecay_decay_steps": quniform(10, 2000, 10),
   "dropout": uniform(0.0, 0.5),
   "clip_value_low": uniform(0.0, 0.2),
   "dist_mult": uniform(0.01, 0.2),
}
```

```
search_space = {
       "layernorm": samp([False, True]),
       "ffn_dist_hidden_dim": samp([32, 64, 128, 256]),
       "ffn_dist_num_layers": samp([1, 2, 3, 4]),
       "distance_dim": samp([32, 64, 128, 256]),
       "num_node_messages": samp([1, 2, 3, 4]),
       "num_graph_layers_common": samp([1, 2, 3, 4]),
       "num_graph_layers_energy": samp([1, 2, 3, 4]),
       "bin_size": samp([16, 32, 40, 64, 80]),
       "normalize_degrees": samp([True, False]),
       "output_dim": samp([32, 64, 128, 256]),
}
```

Improvements from hypertuning

- Loss curves before (left) and after (right) hypertuning
- Only the physical datasets, no single particle gun samples
- Mean and standard deviation of 10 trainings with identical hyperparameters
- Mean validation loss decreased by ~44%





[1] E.Wulff, M. Girone, J. Pata https://doi.org/10.1088/1742-6596/2438/1/012092