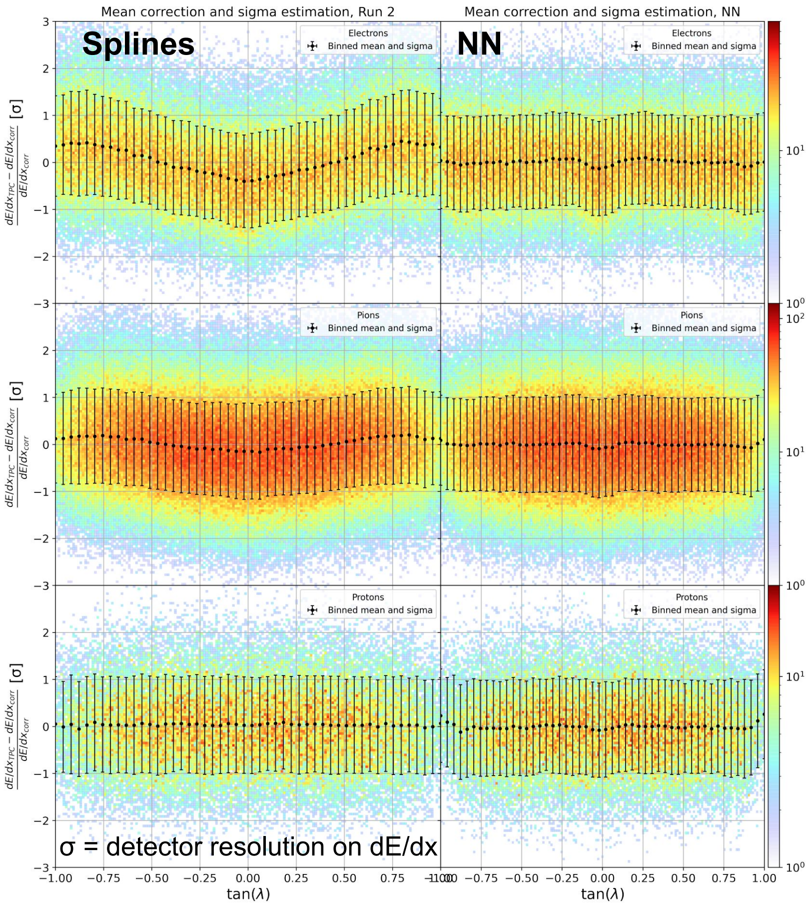
ML activities in ALICE

Fabio Catalano, Francesco Mazzaschi

2nd CERN IT Machine Learning Infrastructure Workshop
11/10/2023

TPC PID calibration with neural networks

- NN corrections to the Bethe-Bloch parameterization of particle energy loss (dE/dx)
  - track information as input (p, tan(λ), N_{CLS}, …)
  - n-dimensional (6D) corrections ⟶ correlations kept into account
  - only one iteration needed
- Replaced the Spline corrections of Run 2
  - per-dimension splines assuming factorisation
  - multiple iterations to produce
- Performance comparable or better than Splines on Run 2 data
- Fully data-driven NN corrections now available for all Run 3 pp data

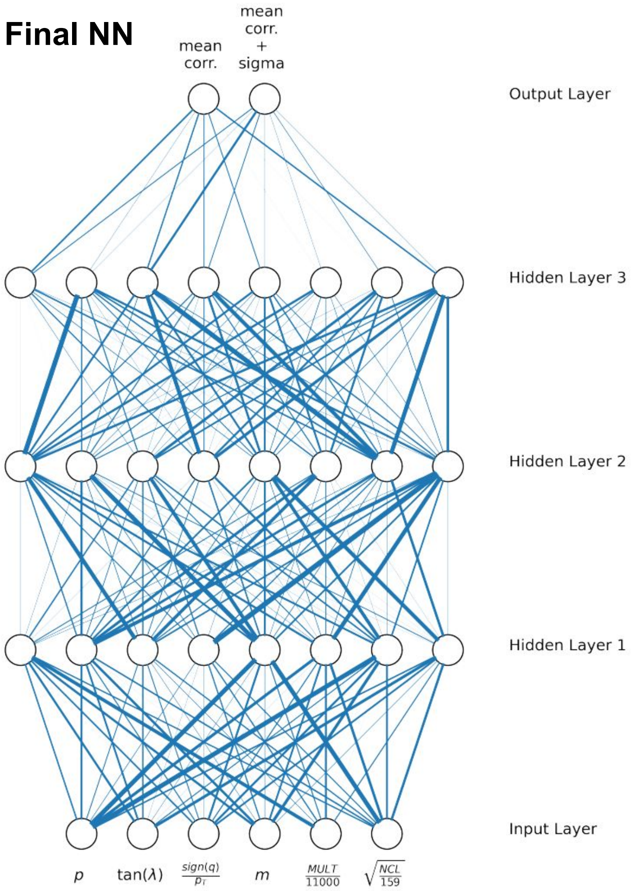


Further details: CERN-THESIS-2022-342

TPC PID calibration with neural networks

- Fully connected NNs performing a regression
  - PyTorch library used
  - final NN trained on the output of two larger models (12 nodes x 10 layers) for performance reasons at inference time
- Training performed for each data-taking period
  - starting from analysis-object data (AO2D)
  - on GSI SLURM cluster
  - ~7-8 hours of GPU time on Nvidia V100 or AMD MI100
- ~300 hours of training time per data-taking year
- Trained models uploaded to CCDB and accessible on the WLCG GRID
  - model inference based on ONNXRuntime

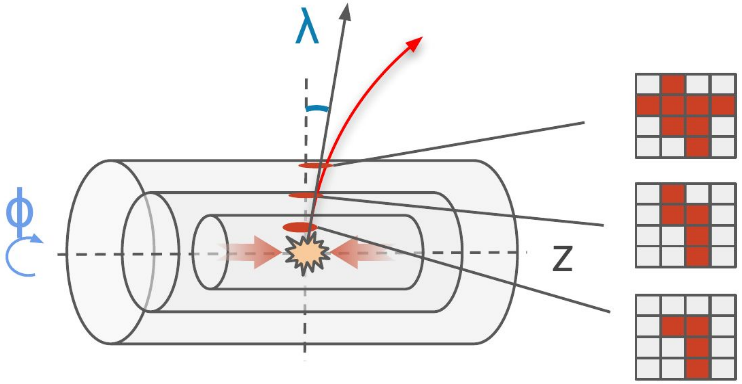
Final NN



Further details: CERN-THESIS-2022-342

Particle identification with the ITS2

- The new ALICE Inner Tracking System (ITS2) has a binary pixel readout
  - no dE/dx information as present in Run 1 and Run 2
- Topology of the produced signal (cluster) can be used as a proxy for the energy loss of the particle



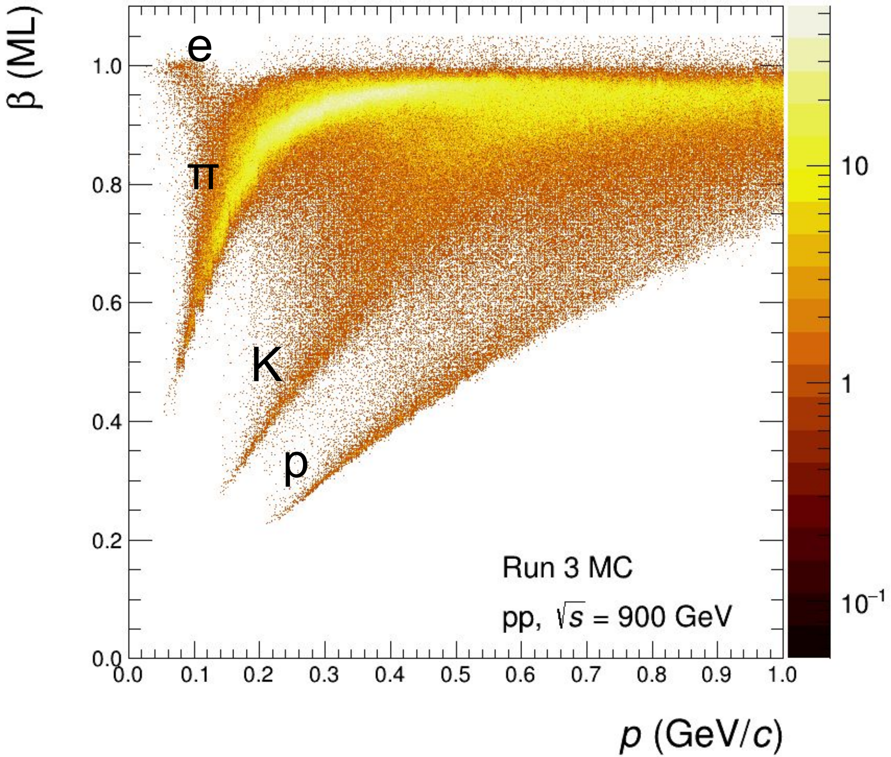
- XGBoost BDT regressor to estimate the particle β
  - track information (p, tan(λ)) and properties of clusters (size, shape, …) in the ITS2 layers as inputs to the model

Particle identification with the ITS2

- Training using particles tagged in TPC
  - starting from reconstruction output
  - not dependent on data taking period
  - ~ 30 min on Turin INFN cluster
    - Nvidia RTX A6000 GPU 48 GB

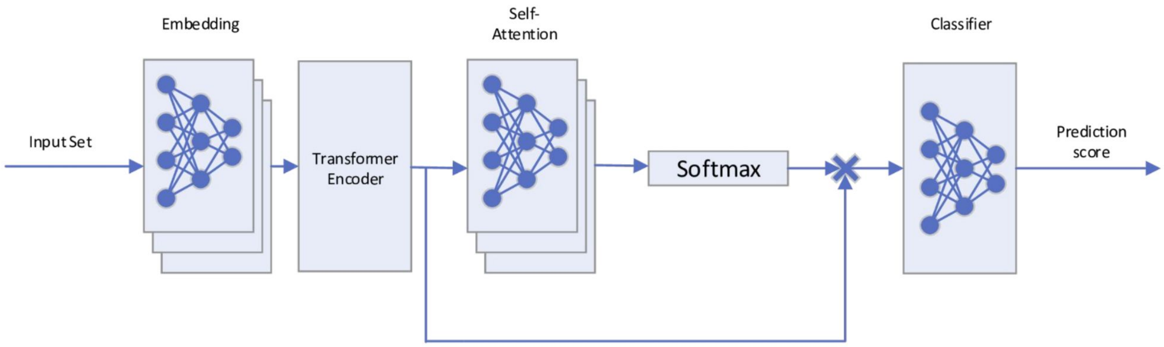
- Method validated on Run 3 MC
  - good separation between e, π, K, p at low momentum

- Encouraging results on Run 3 data
  - further studies using tagging performed with K_s^0, Λ, Ω decays ongoing
  - training on large data samples foresee



Combination of detector PID information

- Combine the particle-identification information of different detectors to provide global PID
  - replace hand-crafted combinations and selections
  - provide high purity samples of particles of a given species
- Different NN models trained for each particle species and data-taking period
  - PyTorch library used, ~1h training time per model on Nvidia GTX 1660Ti
  - starting from analysis-object data (AO2D)
  - track information and detector signals related to PID as input



- Information from one or more detector could be missing
  - typical for low p_T particles
  - solution: model based on feature set embedding (FSE) with multi-head self-attention mechanism

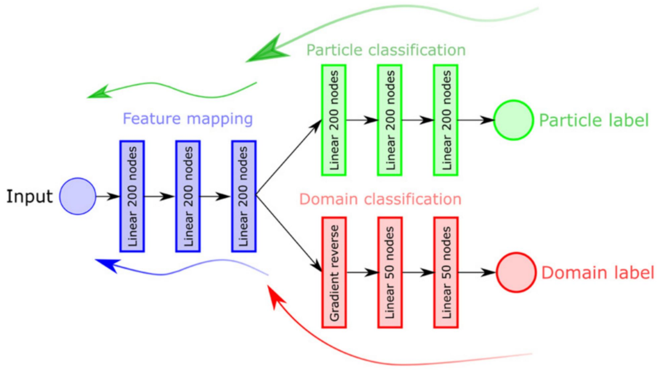
Further details: M. Kabus talk at CHEP 2023

Combination of detector PID information

- On Run 2 pp MC, NN with self-attention + FSE shows better performance than other approaches for incomplete data
  - data imputation
    - mean
    - linear regression
  - NN ensemble

	π			p			K		
model	purity	efficiency	F_1	purity	efficiency	F_1	purity	efficiency	F_1
mean	0.9718	0.9934	0.9825	0.9559	0.8927	0.9232	0.8858	0.8081	0.8452
regression	0.9723	0.9931	0.9826	0.9520	0.8973	0.9238	0.8795	0.8168	0.8470
case deletion	–	–	–	–	–	–	–	–	–
NN ensemble	0.9745	0.9914	0.9829	0.9607	0.8895	0.9237	0.8751	0.8207	0.8470
attention + FSE	0.9734	0.9937	0.9835	0.9648	0.9009	0.9318	0.8841	0.8337	0.8581

best model **2nd best model**



- Further developments
  - address data-to-MC discrepancies using domain adversarial neural networks
  - define approach to systematic uncertainty estimation

Further details: [M. Kabus talk at CHEP 2023]

Resources for large scale ML trainings

- How to provide resources to support the use cases shown before?
  - systematically train and validate the output of ML models
  - relatively light trainings, but many models and/or frequent re-training needed

- Possibility to use ALICE resources under consideration
  - opportunistic use of ALICE EPN farm
    - composed by ~350 servers, each one equipped with 8 AMD MI50 or MI100 GPUs (plus 2x 32-core AMD CPUs and 512 GB of RAM)
    - used for synchronous and asynchronous reconstruction (w.r.t. data taking)
  - HPC Perlmutter in Berkeley
    - on nodes equipped with Nvidia A100 GPUs
  - resources accessible via batch jobs (similar to GRID analysis model)

- Any resource available/foreseen from CERN?

Common resources for ML developments

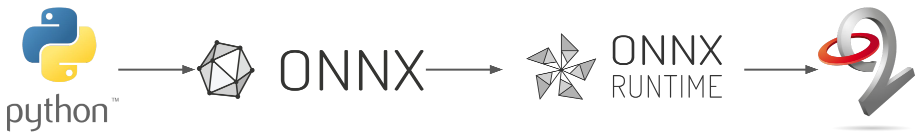
- ALICE is potentially able to provide in-house resources for large(-ish) scale ML projects

- What about hardware resources for analysers and early/cutting-edge developments?
  - dedicated resources for ML usually provided to developer/analyser by his/her institution

- Central resources from CERN available/foreseen?
  - interested in easy access $\longrightarrow$ shell and/or Jupyter notebook
  - SWAN very nice but limited in term of resources

Integrate ML model inference in ALICE software

- Inference of ML models in ALICE software implemented via ONNX + ONNXRuntime
  - positive experience so far



- Models, usually trained with Python software, exported to ONNX format
  - supports most ML models (BDT, NN, …) and libraries (XGBoost, PyTorch, TensorFlow, ...)
  - stable format ⟶ good for model preservation
  - industry standard

- Inference of models in ONNX format performed by ONNXRuntime library
  - integrated in ALICE software stack
  - C++ API available, some custom classes developed to simplify usage
  - mainly used on the GRID at the moment
  - ML models stored in CCDB and retrieved at runtime, possible also to use CVMFS

Integrate ML model inference in ALICE software

- Under investigation
  - provide data from Apache Arrow tables (ALICE Run 3 data format) to ONNXRuntime efficiently and with flexibility
  - TMVA SOFIE as inference provider
    - experimental tool in ROOT to read and perform inference for ONNX models
    - pros: easy integration, possibly better support for ALICE data format through RDataFrame Arrow backend
    - cons: limited number of ONNX operators supported

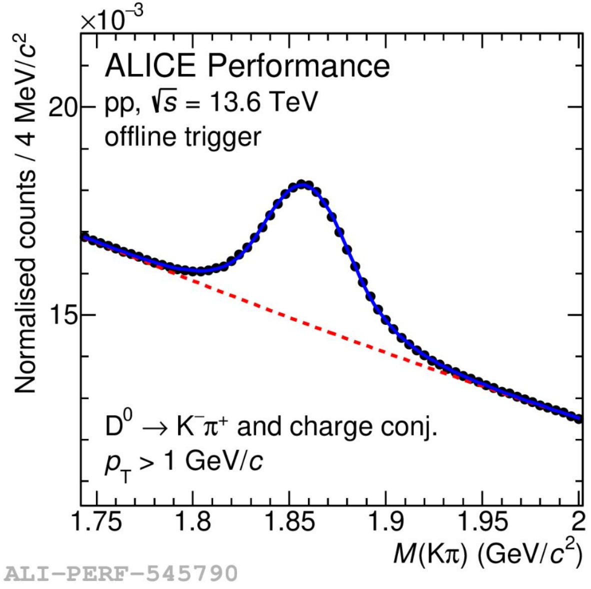
- Problem of inference of ML models on large amount of data probably common to all LHC experiments
  - expertise transfer and/or common developments would be beneficial

Heavy-flavour hadron trigger for pp collisions

- ➢ Software trigger for ALICE high-energy pp program includes selection of interesting events for heavy-flavour (HF) hadron studies [CERN-LHCC-2020-018]
  - running at the analysis level on AO2Ds
  - intervals of raw-data frames selected and subsequently re-reconstructed

- ➢ HF selections based on XGBoost multi-class BDTs
  - training time negligible
  - inference on all collected data using ONNXRuntime
    - 10x speedup if BDTs converted into tensor format with [hummingbird]

- ➢ Currently being used in the skimming of all 2022 and 2023 pp data collected by ALICE



Backup

Ongoing ML activities in Run 3

Signal-vs-background classification
- BDTs and NN replacing “traditional” linear selections

Particle identification (PID)
- exploit complex relationship between track properties and PID
  - NNs to combine info from different detectors
  - PID with ITS2 using BDT regression

TPC response calibration
- ML to compute corrections of spatial charge distortions
- NN for energy-loss (dE/dx) calibration

General framework developments
- common tools and procedures

HF-hadron trigger
- BDTs to trigger on displaced decay-vertex topologies

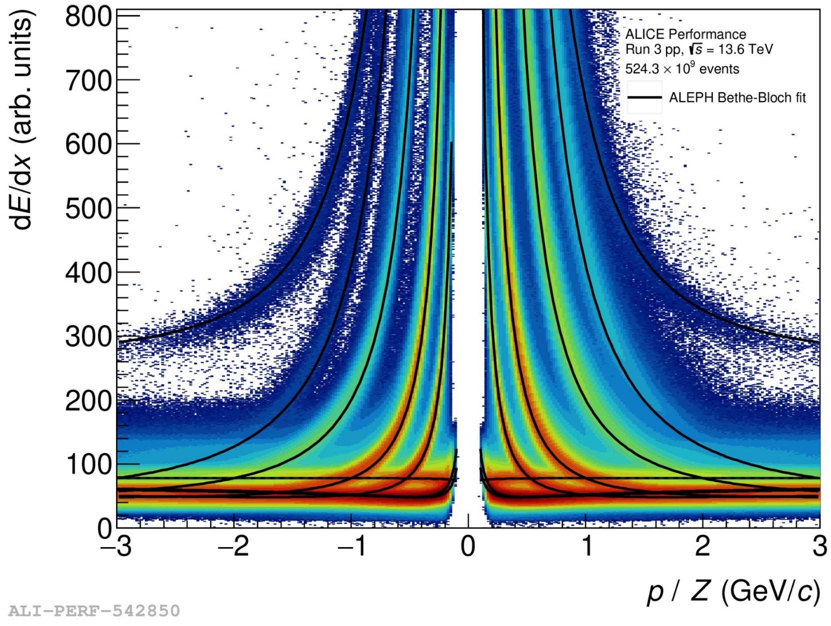
MFT-MCH track matching
- NN classification giving the score for a correct match

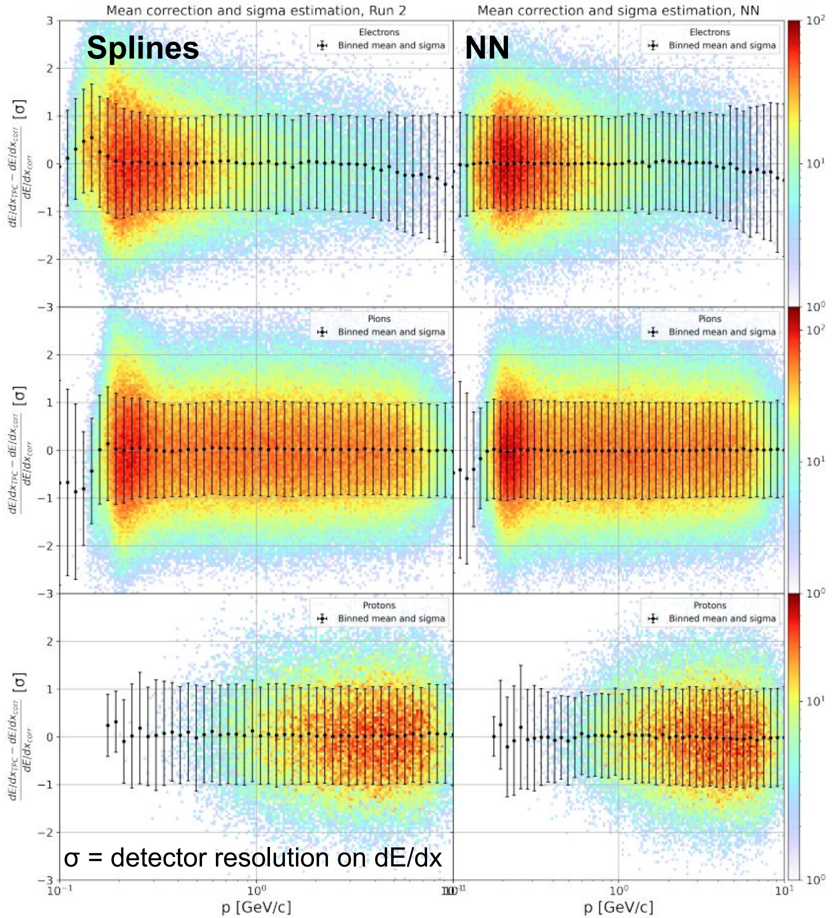
ML for EMCal QC/calibration
- alert experts quickly and accurately about issues in data-taking, flag bad towers

Fast simulation
- ZDC calorimeter simulation with Generative Adversarial Networks and Variational Autoencoders

... not a comprehensive list!

TPC PID calibration with neural networks





Further details: [CERN-THESIS-2022-342]

Combination of detector PID information

- Tests of domain adversarial neural networks on Run 2 pp MC

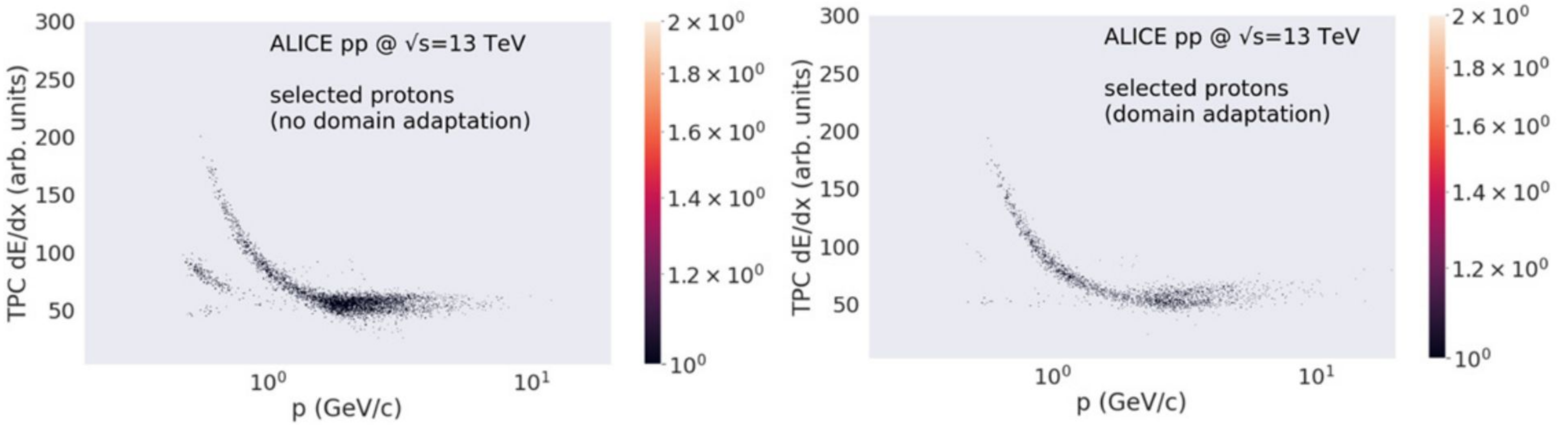


Figure 3. Preliminary result of DANN PID for the TPC detector signal (dE/dx) as a function of particle momentum for particles identified as protons without domain adaptation (left) and with domain adaptation (right).

Further details: [JINST 17 C07016 (2022)]

Software for ML

- ➢ ML applications in ALICE based either on
  - ROOT TMVA
  - python software stack (scikit-learn, XGBoost, TensorFlow, PyTorch, …)

ROOT Data Analysis Framework TMVA

- ✅ Well integrated in ALICE analysis software and on the GRID
- ❌ Limited selection of ML models and tools
- ❌ Limited documentation

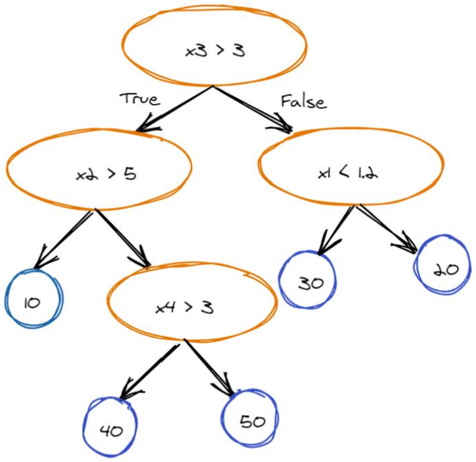
python™ dmlc XGBoost scikit learn

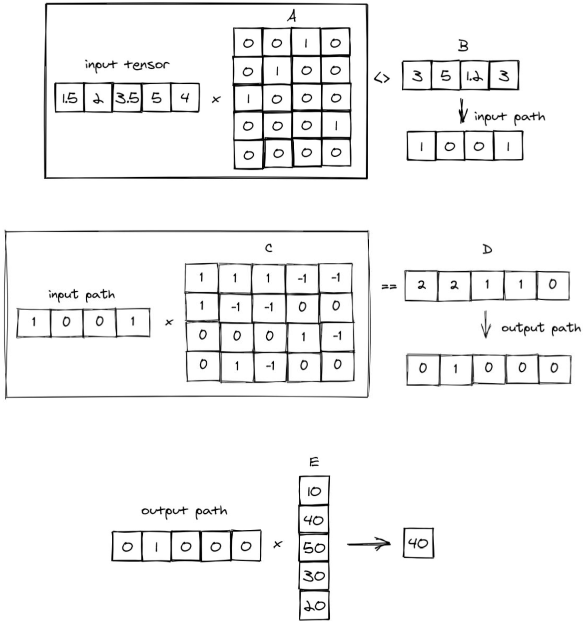
- ✅ Widely used outside HEP
- ✅ Many ML models and techniques available
- ❌ Need interfaces with the ALICE environment (uproot, treelite, ONNXRuntime)

BDT inference optimisations

- ONNXRuntime is optimised for the tensor computations typical of NNs
  - not so efficient for the inference of BDTs (used in many ALICE analyses) and classical ML algorithms

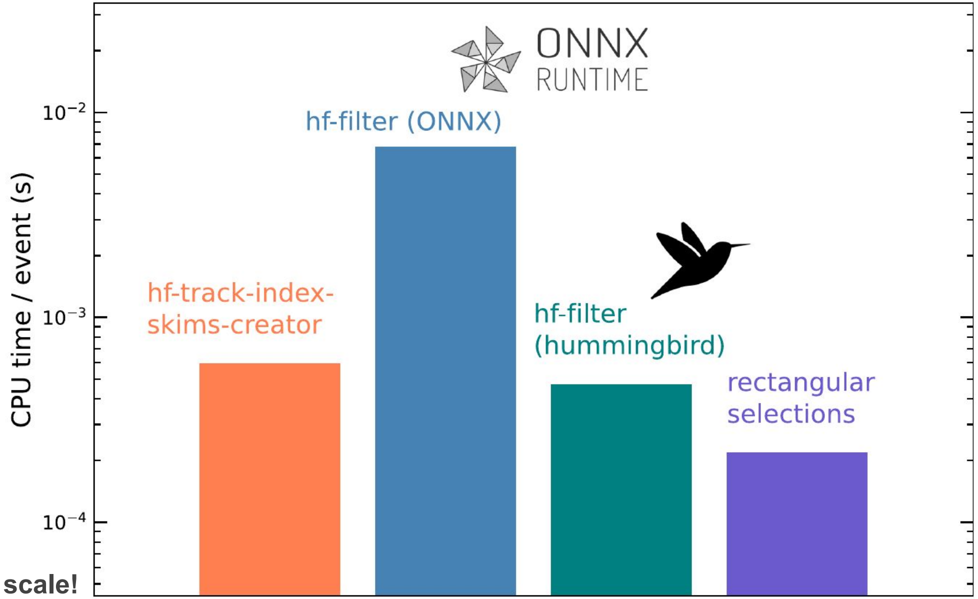
- hummingbird (python library)
  - converts trained ML models into tensor computation for faster inference





BDT inference optimisations

- Performance improvement given by humminbird tested in the context of heavy-flavour hadron trigger studies
  - multi-class BDTs used as software trigger for pp events
  - about 10x speedup compared to non-converted models
  - CPU time / event comparable to rectangular selections



Mind the log scale!