ML in ALICE + ideas for HMPID

Fabio Catalano

25/01/2023

Ongoing ML activities in Run 3

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

HF-hadron trigger

- BDTs to trigger on displaced decay-vertex topologies

... not a comprehensive list!

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

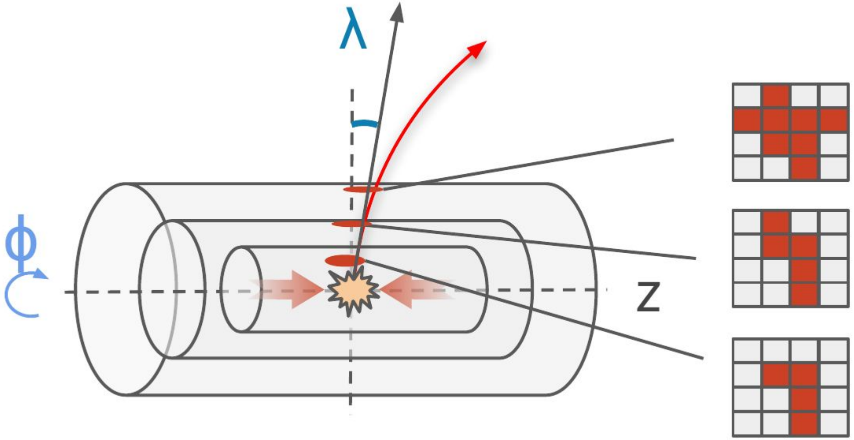
- ZDC calorimeter simulation with Generative Adversarial Networks and Variational Autoencoders

General framework developments

- common tools and procedures

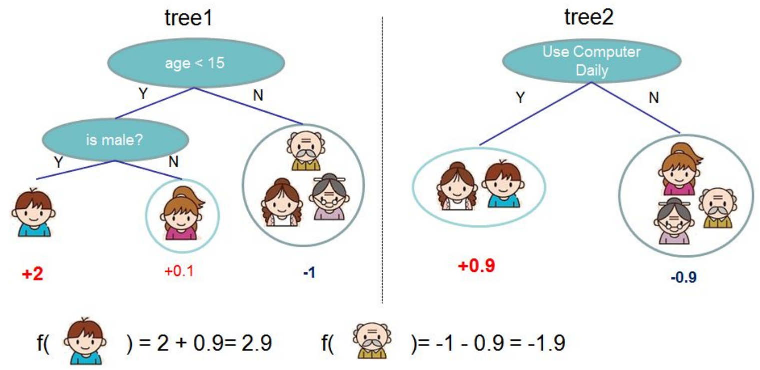
PID with the ITS2

- 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



PID with the ITS2 – Model and inputs

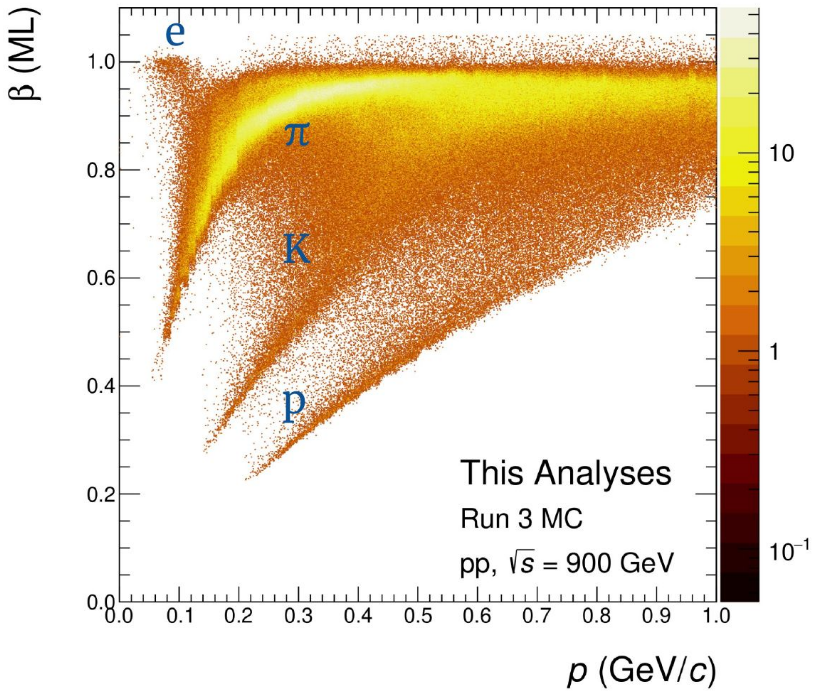
- XGBoost BDT regressor to estimate the particle β
  - collection of Decision Trees (DT)
  - BDT result is the sum of the DT outputs

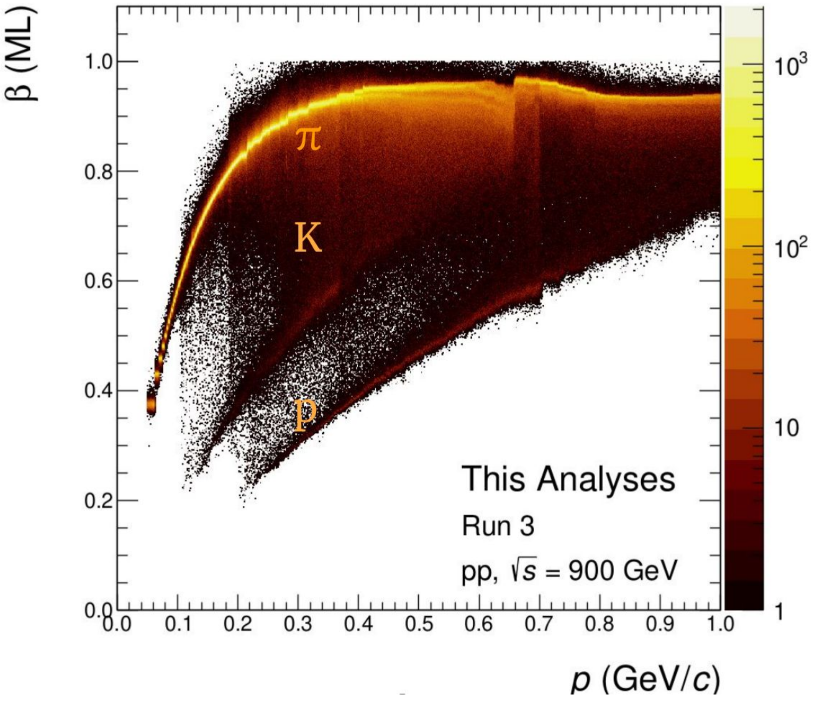


- "High-level" variables as input
  - track information (p, tan(λ)) and properties of clusters (size, shape, …) in the ITS2 layers

p	ITS-TPC track momentum
tgL	ITS-TPC track tangent of polar angle
ClSizeL*n*	number of activated pixel in the n^{th} layer
meanClsize	average number of activated pixel x Cos(λ)
mean_patt_ID	average ID number associated with cluster shape

PID with the ITS2 — Results



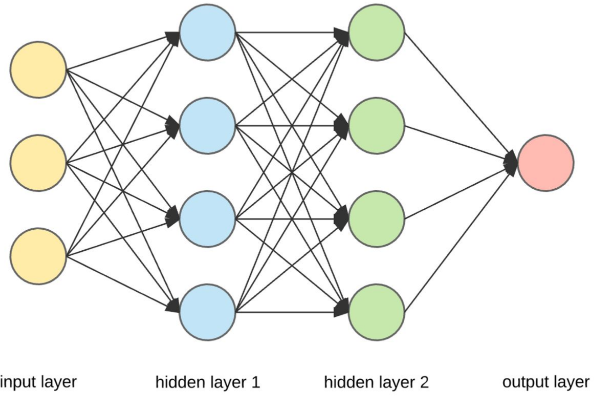


- Method validated on Run 3 MC
  - good separation between π, K, p
- Encouraging results on Run 3 data
  - training using particles tagged in TPC
  - waiting for well calibrated data

PID with the ITS2 – Python software used

- hipe4ml used for the BDT training and optimisation (available on PyPI)
    - wrapper around popular ML libraries (scikit-learn, XGBoost, optuna, …)
    - ease typical steps of the analysis workflow
    - common software developed by ALICE members

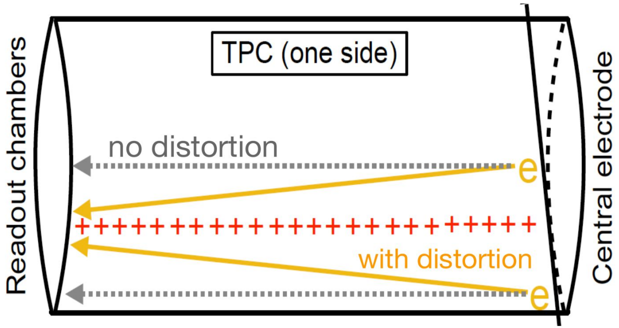
- Same task can be performed using Shallow Neural Networks
    - same inputs and outputs, performance should be similar to BDT
    - need to develop a script for model training and optimisation using TensorFlow or PyTorch
    - possible contribution to hipe4ml package (some work already started by Bachelor student)

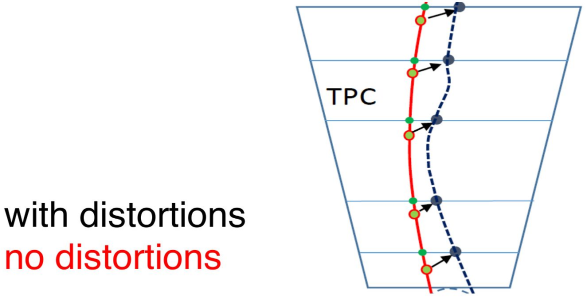


Corrections for TPC space-charge distortions

- ALICE Time Projection Chamber (TPC)
  - charged particle tracking and identification
  - continuous data readout up to 50 kHz in Pb-Pb collisions

- Positive ions (space charge) from the readout chambers pile up inside the drift volume
  - cause large space-charge distortions of measured space points O(mm - cm)
  - the space-charge density and distortions fluctuate with timescale O(ms)

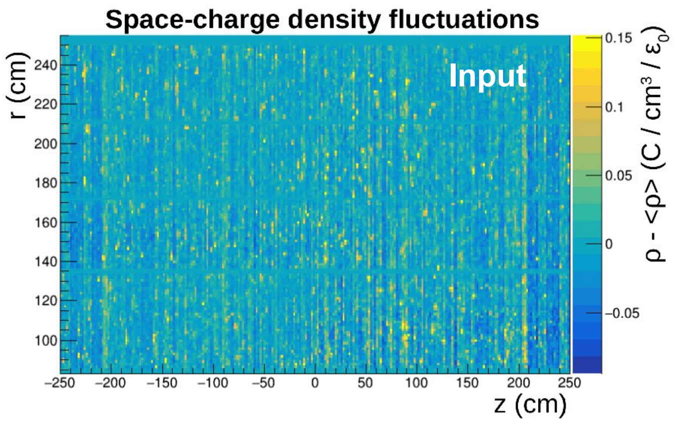


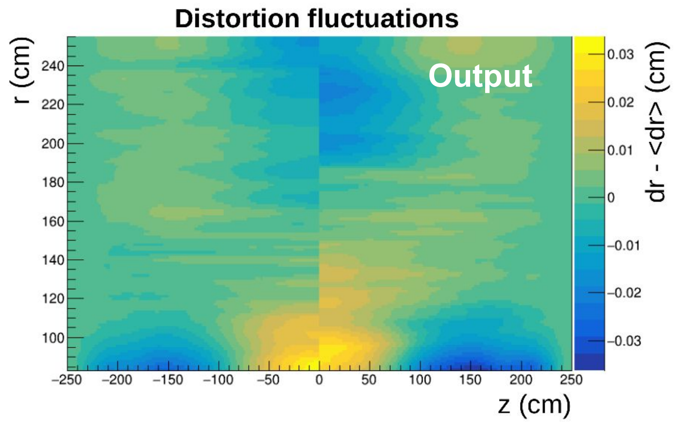


- ML algorithms (BDTs, NNs) and Convolutional Neural Networks to correct the distortions
  - fast data-driven prediction of the distortions from the measured space-charge density

CNN for distortion corrections — Model and inputs

- ➢ CNN (U-Net) implemented in [TensorFlow](https://www.tensorflow.org)
- ➢ Inputs:
  - ○ space-charge density fluctuations in 3D (r, φ, z)
  - ○ mean space-charge density in 3D (r, φ, z)
- ➢ Output:
  - ○ predicted distortion fluctuations in the radial direction

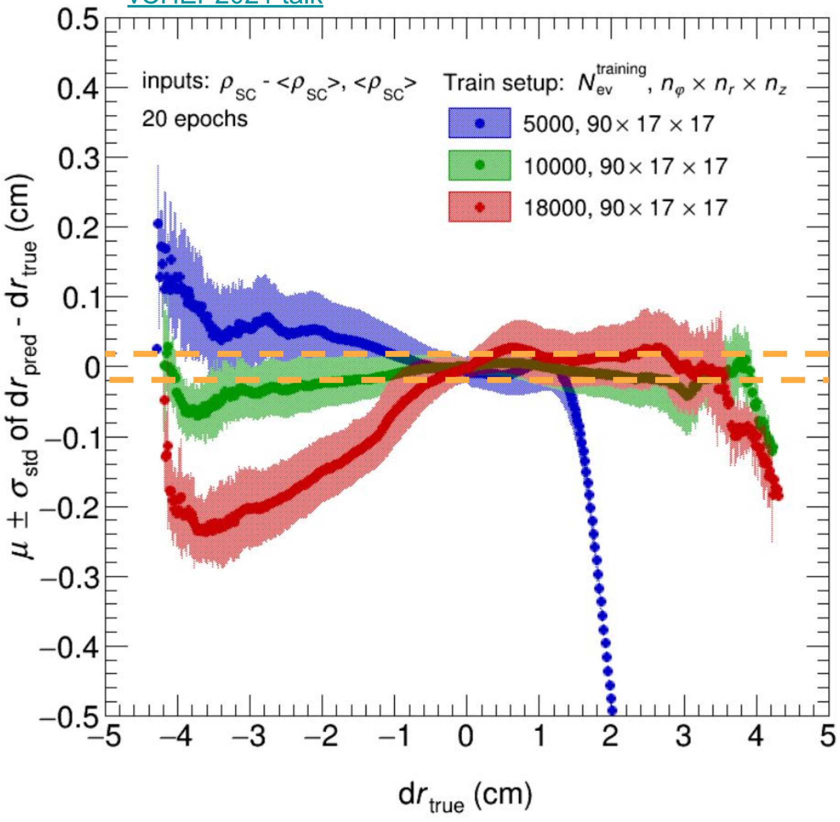




CNN for distortion corrections — Results

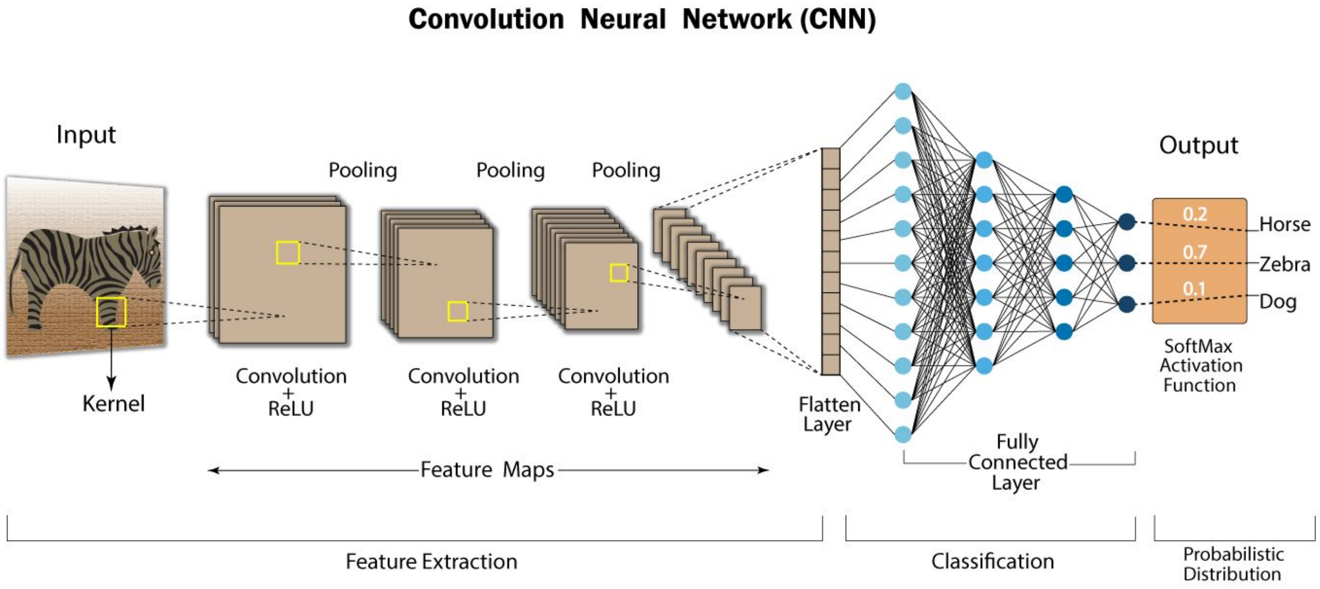
- Goal
  - difference between predicted and true distortion fluctuations within the intrinsic TPC resolution (about 200 μm)
- NN difficulties in learning both global and local effects and dealing with the intrinsic asymmetries of the problem

vCHEP2021 talk



DNN tools for HMPID — Parameter extraction

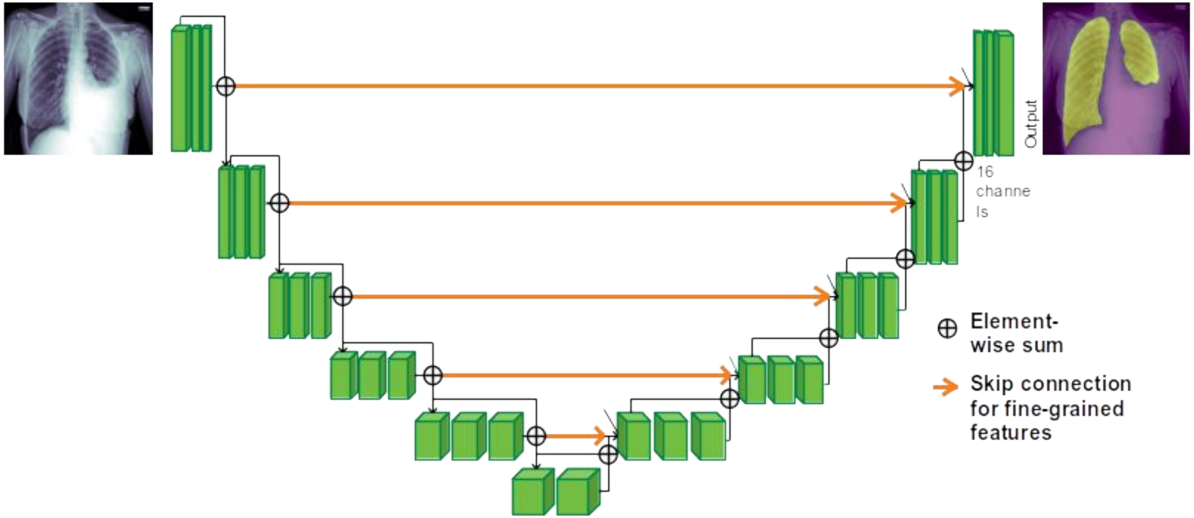
Thinking about replacing parts of the reconstruction workflow



- CNN to extract parameters of interest from image-like input
  - no need to provide high-level quantities
  - features are learnt by the model during the training

DNN tools for HMPID — Segmentation

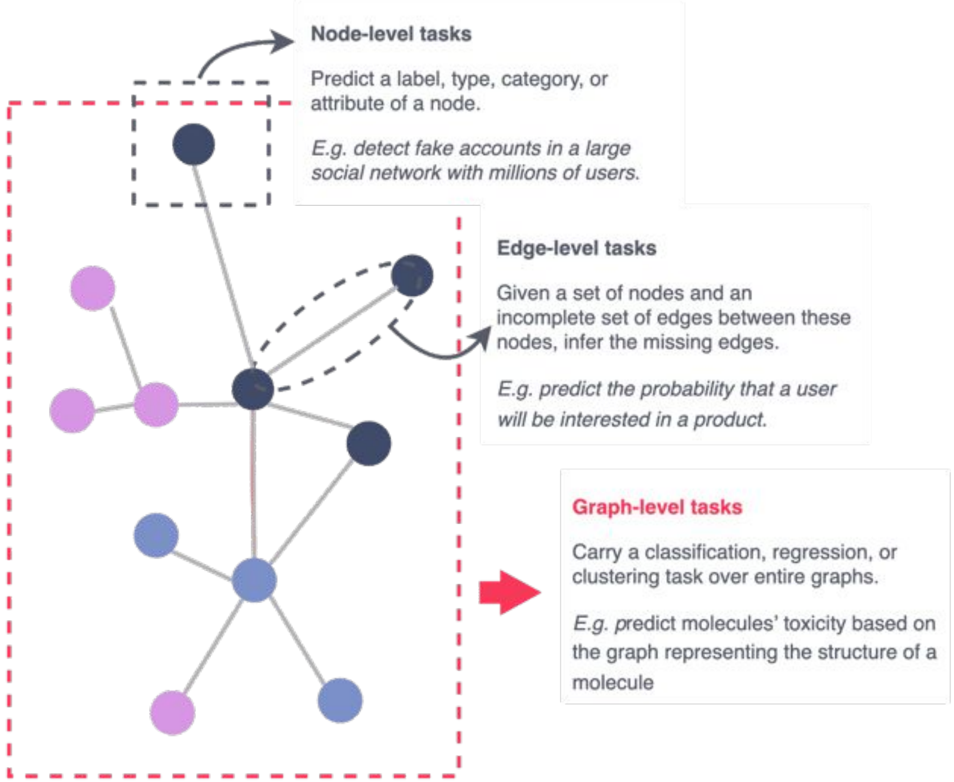
Thinking about replacing parts of the reconstruction workflow



- ➢ U-Net for image segmentation
  - used to isolate elements of interest on a image
  - could be applied as a pre-processing step to separate signals from different particles falling in the same readout frame

DNN tools for HMPID – GNN

Thinking about replacing ~~parts of~~ the reconstruction workflow



- Graph Neural Networks
- Input embedded in a graph
  - usually more friendly (and powerful) than images for physics problems
- Could be used, e.g., to retrieve the number of tracks and their parameters of interest starting from the signals in one readout frame
  - potentially can replace all the reconstruction workflow

ML inference on the GRID/EPN/FLP

- Models exported to ONNX format
  - supports almost any ML model (BDT, NN, …) and library (XGBoost, PyTorch, TensorFlow, ...)
  - industry standard
- Inference of models in ONNX format performed by ONNXRuntime
  - C++ API available
  - integrated in O2/O2Physics
  - ML models stored in CCDB

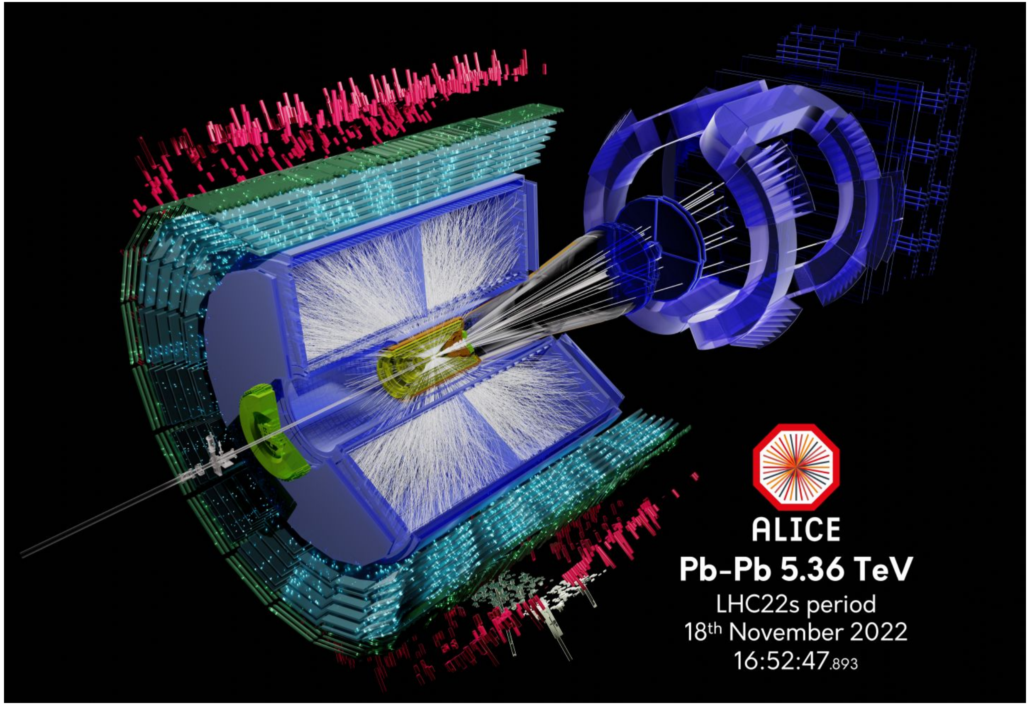
- Other possibilities under investigation
  - ROOT TMVA
    - only for models trained in ROOT
  - TMVA SOFIE
    - experimental tool in ROOT to read and perform inference for ONNX models
    - limited number of ONNX operators supported

Backup

Machine learning in ALICE

- Fundamental tool to
  - maximise the physics potential of the measurements
  - cope with the huge amount of data produced at LHC

- ALICE ML activities involve
  - physics analyses
  - detector calibrations
  - data quality control (QC)
  - Monte Carlo simulations



Run 2 ML applications at a glance

Established

Signal-vs-background classification

- Boosted Decision Trees (BDTs) and Neural Networks (NN) replacing “traditional” linear selections

Jet p_T reconstruction

- correction for the background from the underlying event
- regression task using shallow NN

Heavy flavor jet tagging

- BDTs and Deep Neural Networks (DNN) to tag heavy-flavour jet topologies

Monte Carlo (MC) reweighting

- improve agreement between data and MC simulations

Data quality assurance (QA)

- K-nearest neighbors and Autoencoders to detect outliers

RootInteractive

- tool for multidimensional statistical analysis
- wrappers for tree-based models and NNs

... not a comprehensive list!

Run 2 ML applications at a glance

Exploratory

Signal-vs-background classification

- Boosted Decision Trees (BDTs) and Neural Networks (NN) replacing “traditional” linear selections

Jet p_T reconstruction

- correction for the background from the underlying event
- regression task using shallow NN

Heavy flavor jet tagging

- BDTs and Deep Neural Networks (DNN) to tag heavy-flavour jet topologies

Monte Carlo (MC) reweighting

- improve agreement between data and MC simulations

Data quality assurance (QA)

- K-nearest neighbors and Autoencoders to detect outliers

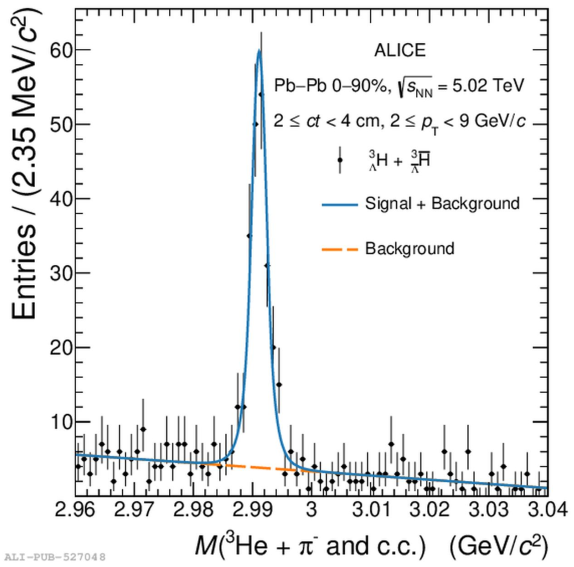
RootInteractive

- tool for multidimensional statistical analysis
- wrappers for tree-based models and NNs

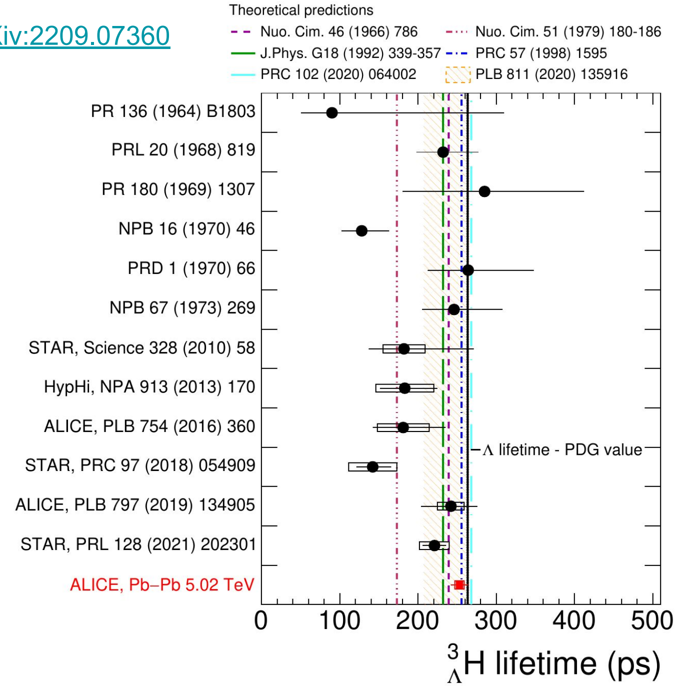
... not a comprehensive list!

ML to improve measurements — Hypertriton

- ➢ XGBoost BDTs for binary classification (signal vs combinatorial background)
- ➢ Models trained employing high-level physical variables (decay length, PID, ...)



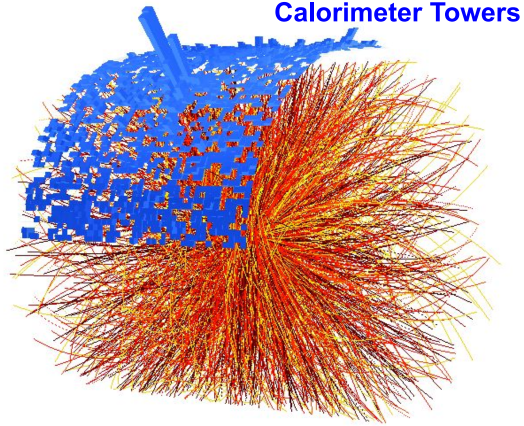
arXiv:2209.07360



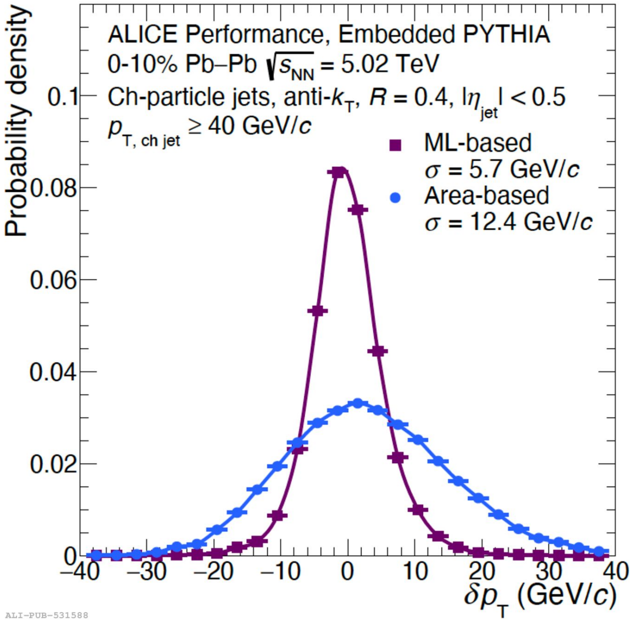
- ➢ Most precise measurements of the hypertriton lifetime and Λ separation energy

ML to improve measurements — Jet p_T reconstruction

- Reconstruction of inclusive jet p_T in heavy-ion collisions
  - difficult due to large fluctuating background from the underlying event



[Phys. Rev. C 99, 064904](#) [arXiv:2211.04384](#)



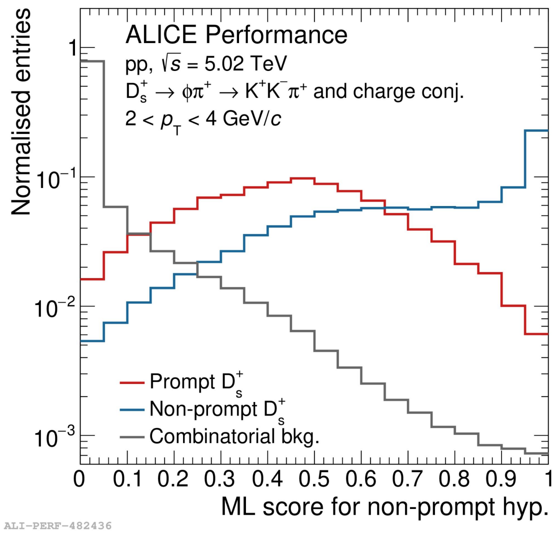
$\delta p_T = p_{T,\text{rec}} - p_{T,\text{true}}$

- Shallow NN from scikit-learn to correct the jet p_T
  - jet and constituent (p_T of leading tracks) properties as input to the model

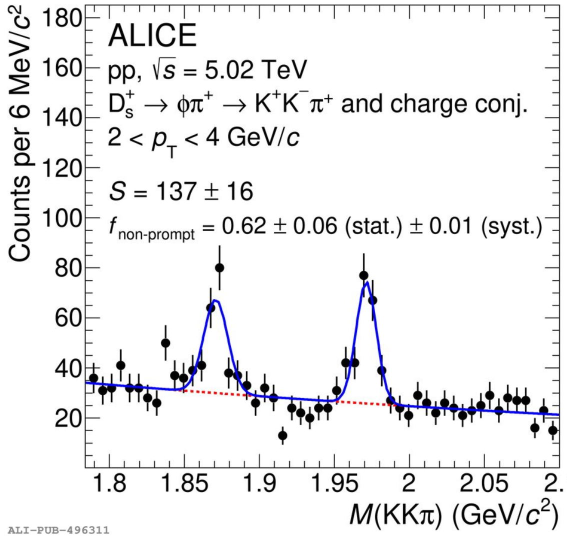
- Improved performance w.r.t. “standard” area based approach
  - narrower δp_T → reduced residual fluctuations

ML to enable new measurements — Non-prompt D mesons

- XGBoost BDTs for multiclass-classification, to disentangle
  - two kinds of signal (prompt and non-prompt D mesons)
  - combinatorial background

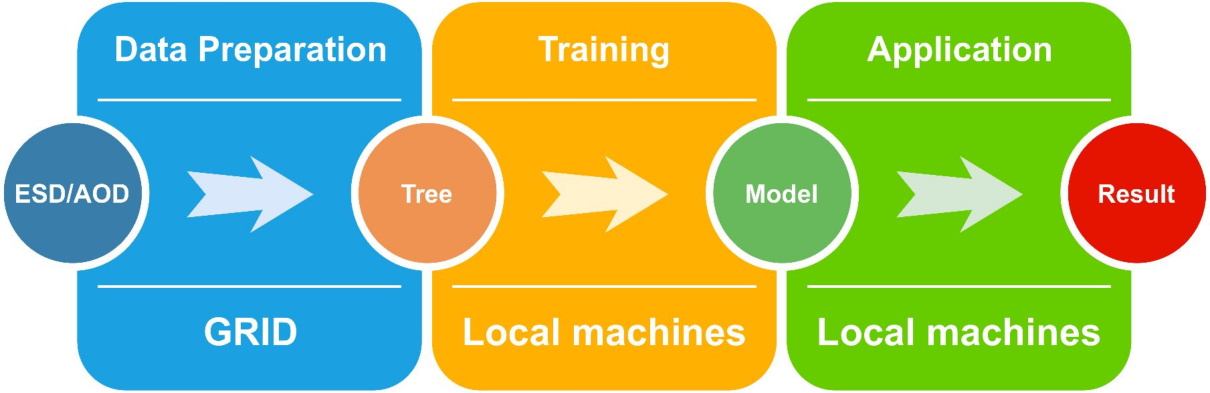


JHEP 05 (2021) 220



- Possible to measure prompt and non-prompt D^0, D^+, and D^+_s mesons separately
  - non-prompt D measurements not possible without ML

Analysis workflow — Local inference



Data preparation

- Information written from ESD/AOD to ROOT TTree
- Full data and MC samples downloaded locally
  - O(10) GBs for pp and p-Pb analyses, a few TBs for Pb-Pb

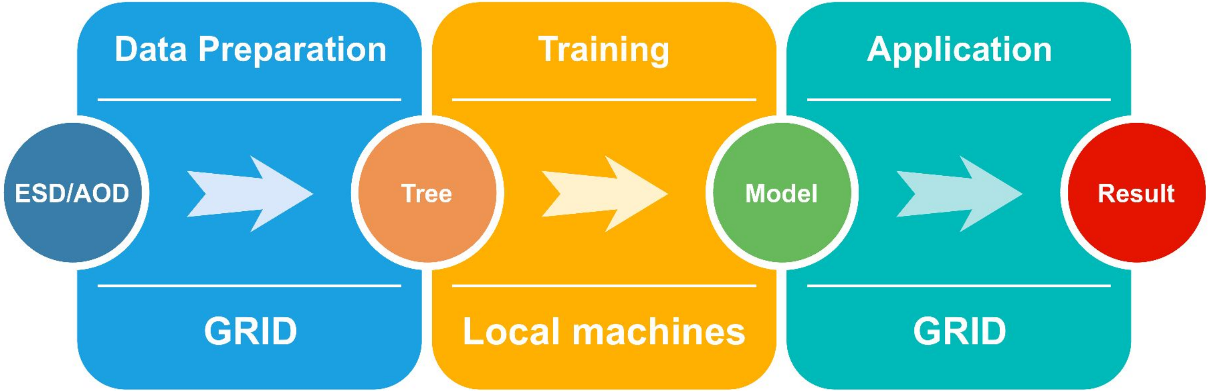
Training and optimisation

- Small fraction of real data and all MC simulations used to train/optimise the model
  - a few minutes/hours on laptops or desktops

Inference on full data sample

- A few minutes/hours depending on the use case
  - in pp and p-Pb on a laptop or desktop
  - in Pb-Pb high-end machine needed to store and process the large amount of data

Analysis workflow — Grid inference



Data preparation

- ➢ Information written from ESD/AOD to ROOT TTree
- ➢ Only data needed for training downloaded locally
  - ○ a few GBs independently from the collision system

Training and optimisation

- ➢ Small fraction of real data and all MC simulations used to train/optimise the model
  - ○ a few minutes/hours on laptops or desktops

Inference on full data sample

- ➢ From about 1 to 3 days on the GRID
  - ○ usual time for a train run from the user point of view
  - ○ the ML inference can be added to standard analysis tasks

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)

Software for ML — Python

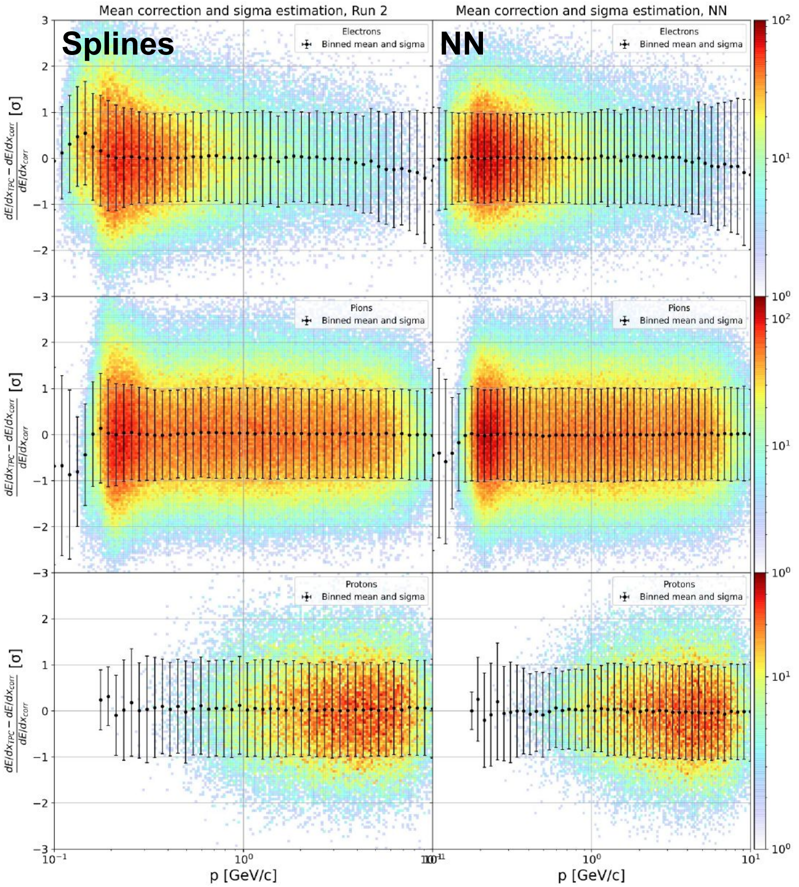
- ML applications in ALICE based either on
  - ROOT TMVA
  - python software stack (scikit-learn, XGBoost, TensorFlow, PyTorch, …)

- Common software developed by ALICE members
  - automatise analysis workflows and/or ease typical steps
  - MachineLearningHEP
  - hipe4ml (available on PyPI)

- Application of BDTs on the GRID enabled by treelite
  - allow to use a model trained in python in the ALICE C++ software
  - integrated in AliPhysics
  - support for XGBoost, LightGBM, scikit-learn

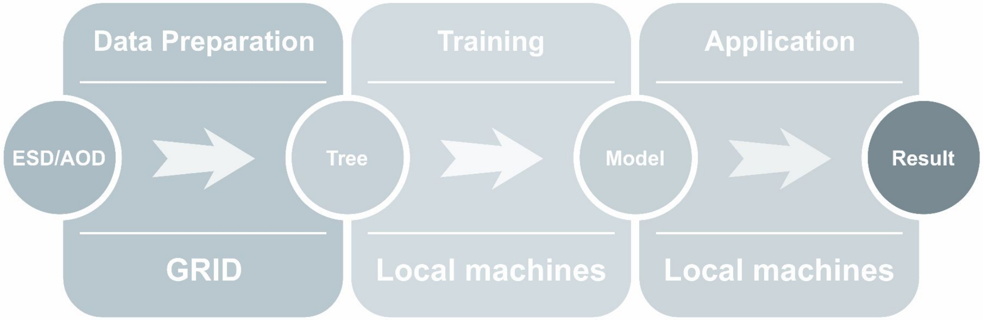
TPC PID calibration

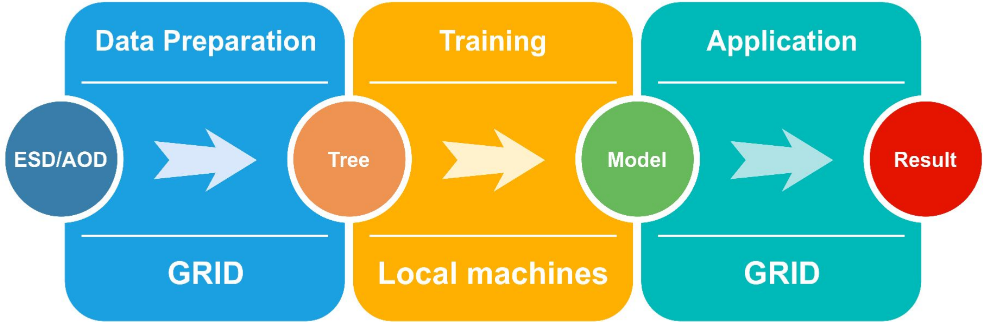
- NN corrections to the Bethe-Bloch parameterization of particle energy loss (dE/dx)
  - track information as input (p, tan(λ), N_{CLS}, …)
  - n-dimensional corrections → correlations kept into account
  - only one iteration needed
- To replace the Spline corrections of Run 2
  - per-dimension splines assuming factorisation
  - multiple iterations to produce
- Performance comparable or better than Splines on Run 2 data
- Need clean V0s data for NN training to have Run 3 results



σ = detector resolution on dE/dx

Analysis workflows in Run 3



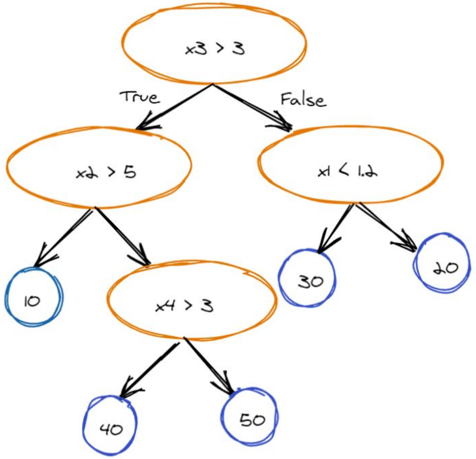


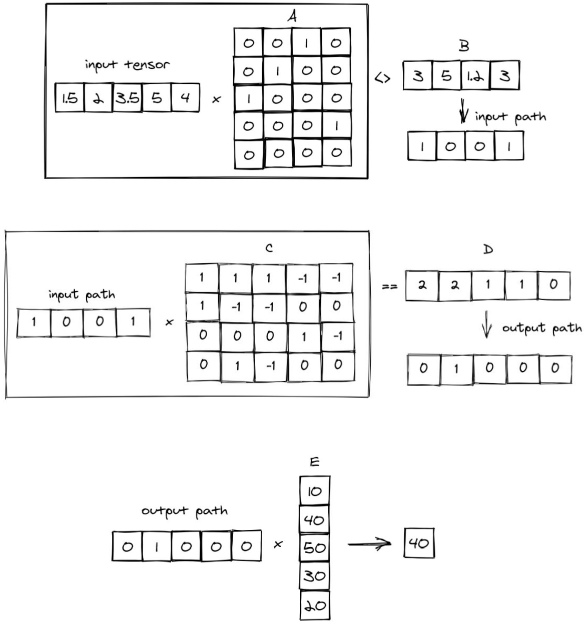
- ALICE will collect a lot of data in Run 3
  - ~14 pb^{-1} of pp collected this year
  - 5-7 nb^{-1} of Pb-Pb expected in full Run 3
- ML model inference on full data samples challenging on local machines
  - even with server-grade machines
- Need a efficient way to perform ML inference on the GRID/EPN/FLP, to support
  - analyses
  - “core” tasks (trigger, calibration, PID, …)

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

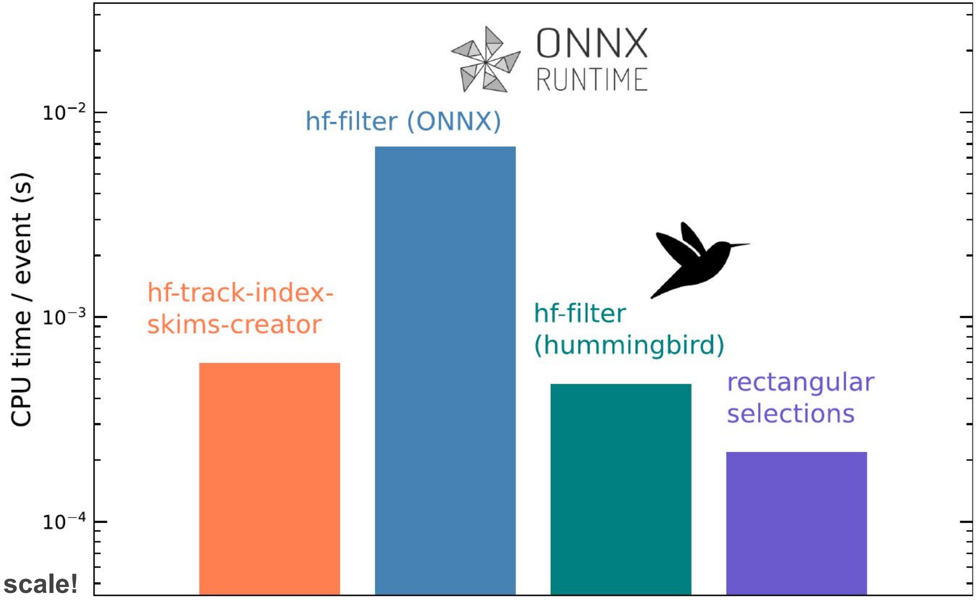
- hummingbird (python library)
  - converts trained ML models into tensor computation for faster inference





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!

Summary and outlook

- ALICE ML activities expanded considerably in the last years

2018

- one published paper used ML
- ML used mainly in analysis

2022

- about 15 papers employing ML
- ML relevant also for PID, detector calibration, trigger, …

- We should continue to strengthen the adoption of ML tools in ALICE activities
- Many developments ongoing
  - create common tools, documentation, and define best practices
  - provide data from Arrow tables to inference libraries efficiently and with flexibility
  - common computing resources for ML training and optimisation