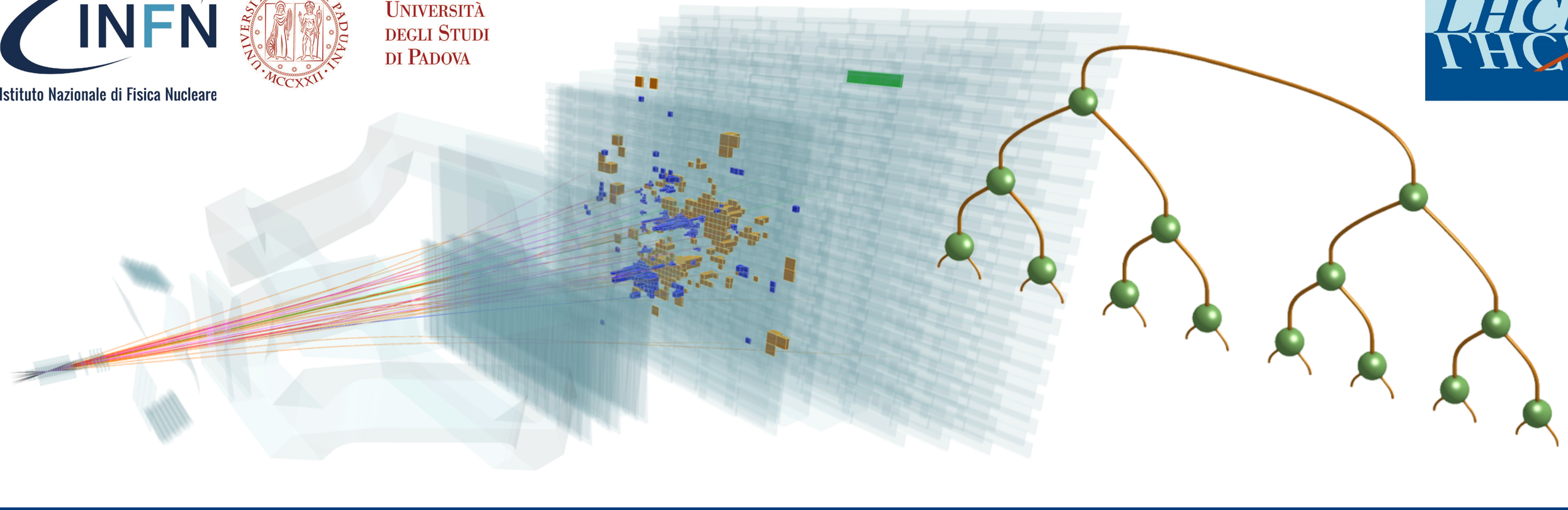


Quantum-inspired Machine Learning on high-energy physics data

Timo Felser[1,2], Alessio Gianelle[1], Donatella Lucchesi[1,2], Simone Montagero[1,2], Lorenzo Sestini[1], Marco Trenti[2], Davide Zuliani[1,2]

ICHEP 2020, Prague (online)
"Computing and Data Handling" Session IV — 31st July 2020

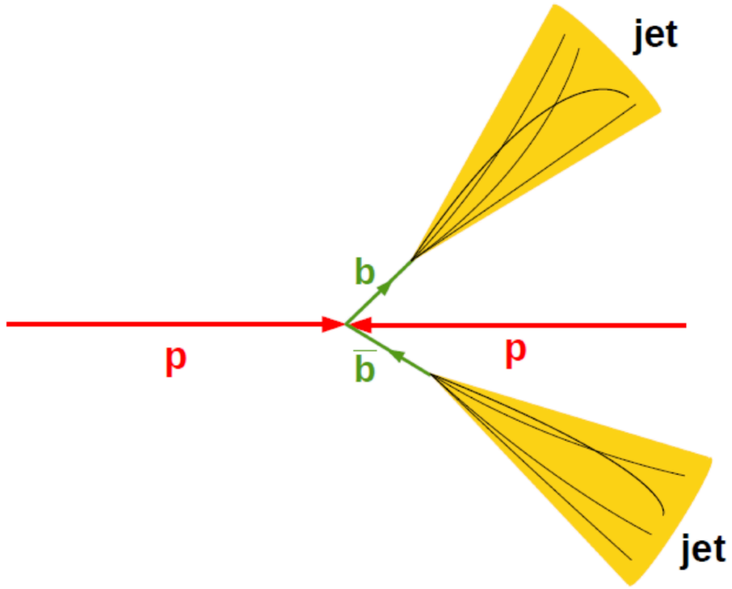
[1] INFN Padova, [2] University of Padova

Outline

- The Physics case
- What is a Tree Tensor Network (TTN)?
- Description of our work and results
- Conclusions

The Physics case

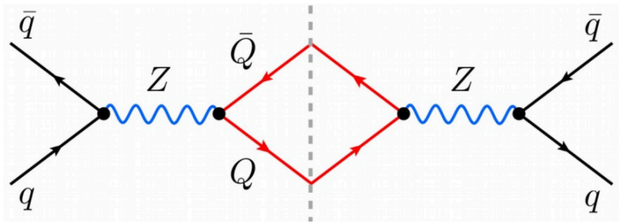
- At LHCb we are interested in identifying jets coming from heavy flavour quarks (b and c quarks)
  - This is achieved with **jet flavour tagging**
- In particular for our physics case we considered **b-jet flavour tagging**



- Try to distinguish jets coming from b and $\bar{b}$ quarks
- Fundamental technique to measure $b\bar{b}$ charge asymmetry

$$A_C^{b\bar{b}} = \frac{N(\Delta y > 0) - N(\Delta y < 0)}{N(\Delta y > 0) + N(\Delta y < 0)} \qquad \Delta y = y_b - y_{\bar{b}}$$

- The asymmetry is sensitive to New Physics



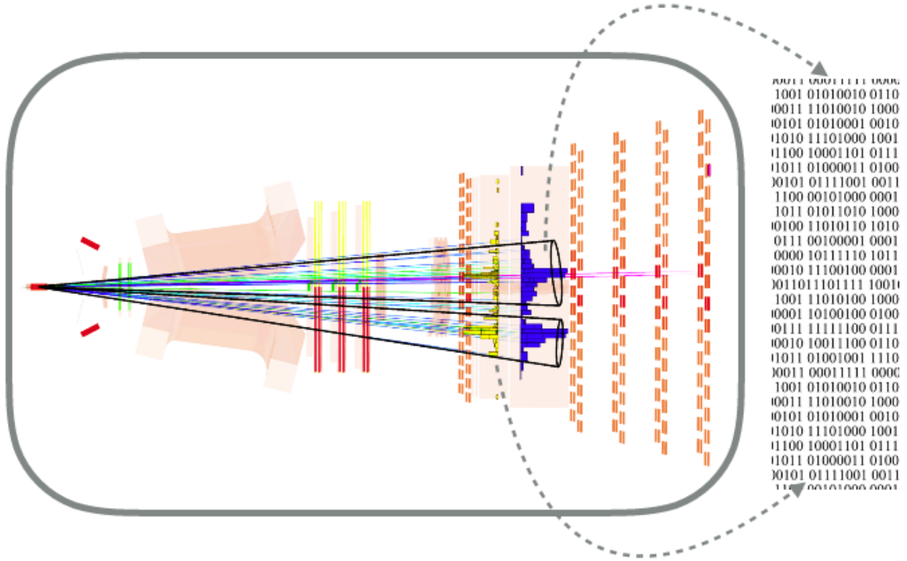
- In Run I the b-jet flavour tagging has been performed with the **muon tagging** approach: the charge of the highest p_T muon inside the jet is used to tag the quark flavour (semi-leptonic decay)

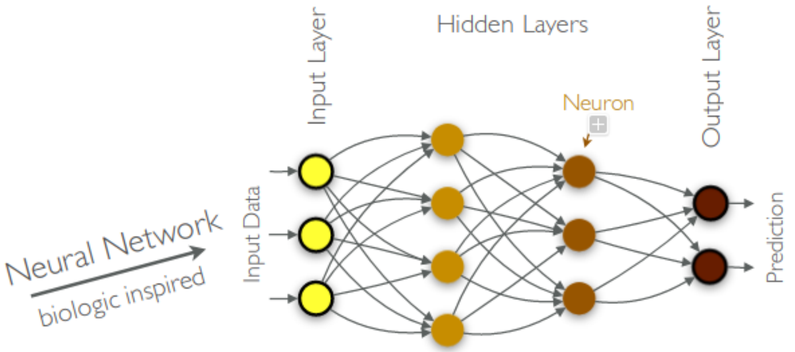
LHCb-PAPER-2014-023 Phys. Rev. Lett. 113 (2014) 082003

The Physics case

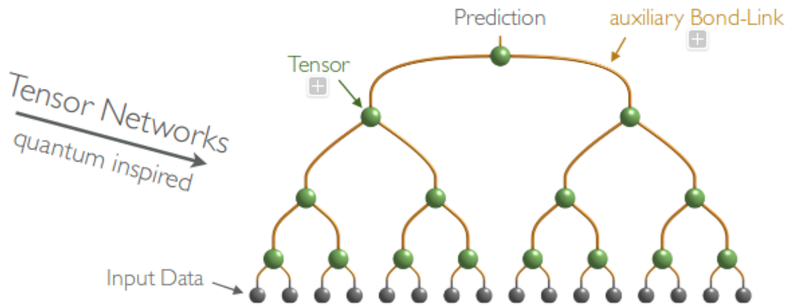
- Recently Machine Learning (ML) algorithms have been developed to solve HEP problems
- Jet sub-structure's variables are used as input

LHCb data





What is already been done

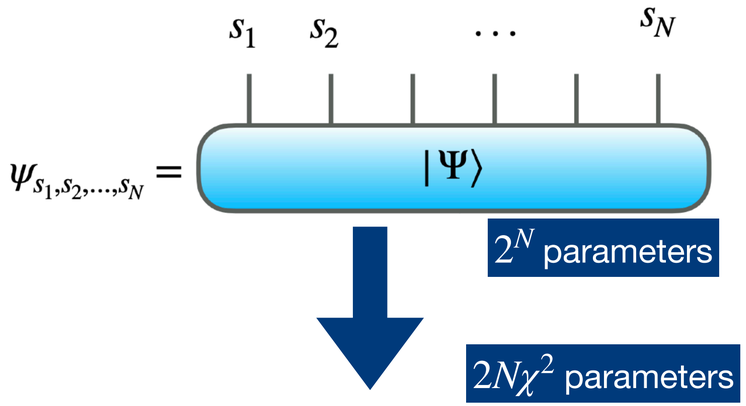


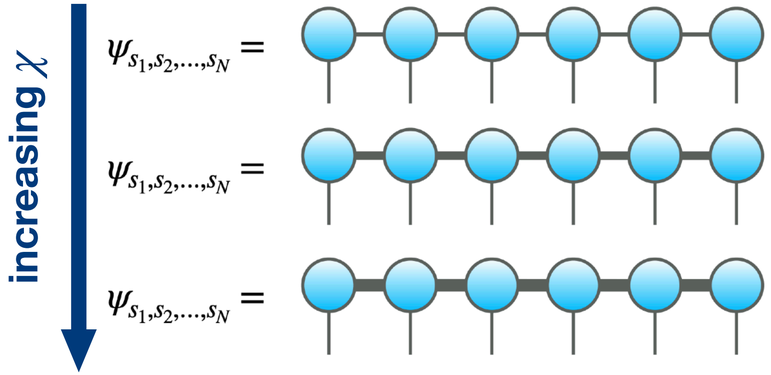
What we are going to do

- We would like to solve some "open problems" of ML:
  - Understand what the algorithm is doing (*e.g.* consider and measure correlations between features)
  - Real time application: prediction time ~ ns

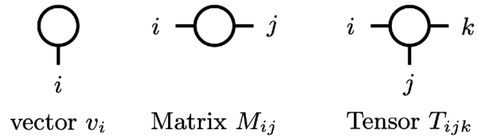
What is a TTN?

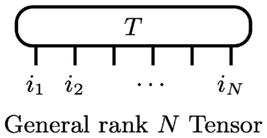
- Tensor Networks are a mathematical tool to investigate **quantum many-body systems** on **classical computers**
- **Efficient representation** of a quantum wave-function $|\psi\rangle$
- Approximation of a high-order tensor by a set of **low-order tensors** with a typical geometry



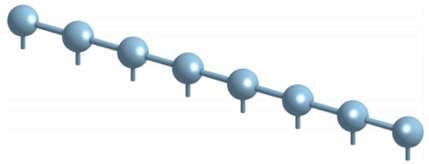


- Typical operations are:
  - Tensor contractions
  - Tensor reshaping
  - Tensor factorisation

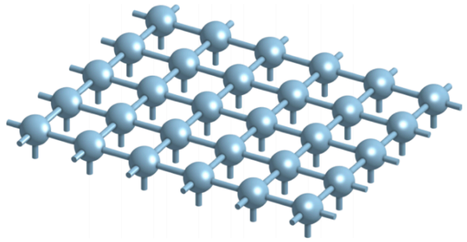




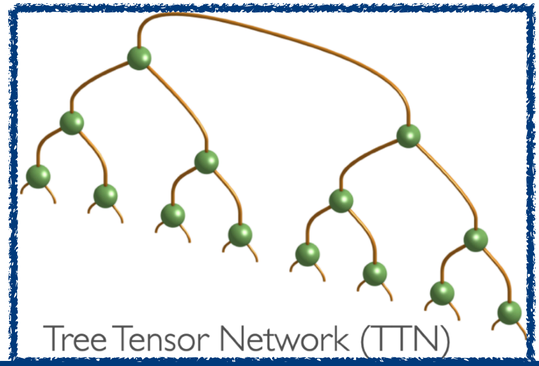
- The approximation is controlled by the **bond dimension** χ
- Several geometries are available:



Matrix Product State (MPS)



Projected Entangled Pair States (PEPS)



Tree Tensor Network (TTN)

What is a TTN?

- A TTN can be used as a **classifier** for **supervised ML problems**
- Data sample x are encoded in dimensional feature space and subsequently classified

$$f(x) = W \cdot \Phi(x)$$

$\Phi(x)$ = feature map
W = weight tensor
$f(x)$ = decision function

- A suitable $\Phi(x)$ for a TTN is a product of N *local feature maps* Φ^{s_i}

$$\Phi(x) = \Phi^{s_1}(x_1)\Phi^{s_2}(x_2)\dots\Phi^{s_N}(x_N)$$

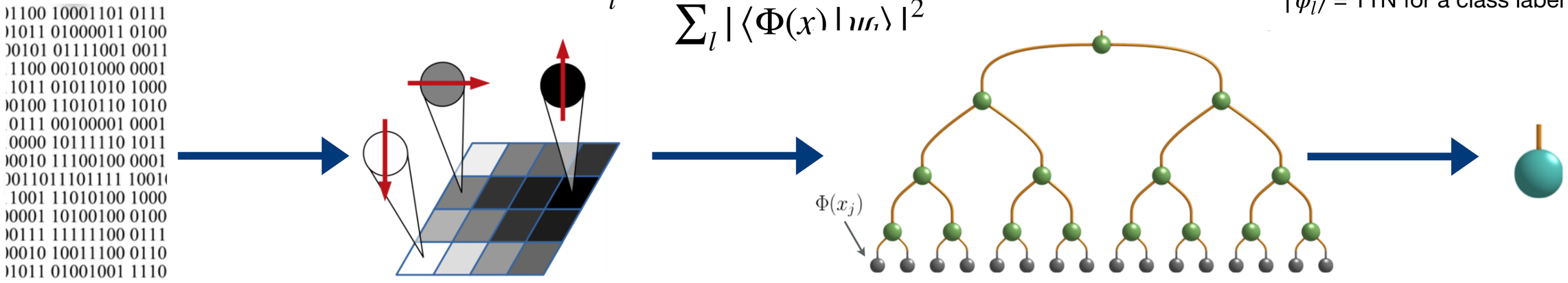
$$\Phi^{s_i}(x_j) = \left[\cos\left(\frac{\pi x_i'}{2}\right), \sin\left(\frac{\pi x_i'}{2}\right)\right]$$

each feature x_i is represented by a "quantum spin"

- Finally the prediction output is a probability

$$\mathscr{P}_l = \frac{|\langle\Phi(x)|\psi_l\rangle|^2}{\sum_l |\langle\Phi(x)|\psi_l\rangle|^2}$$

$|\psi_l\rangle$ = TTN for a class label l



Raw data | **Mapping to "spins"** | **Training and evaluating TTN** | **Prediction**

Description of our work and results

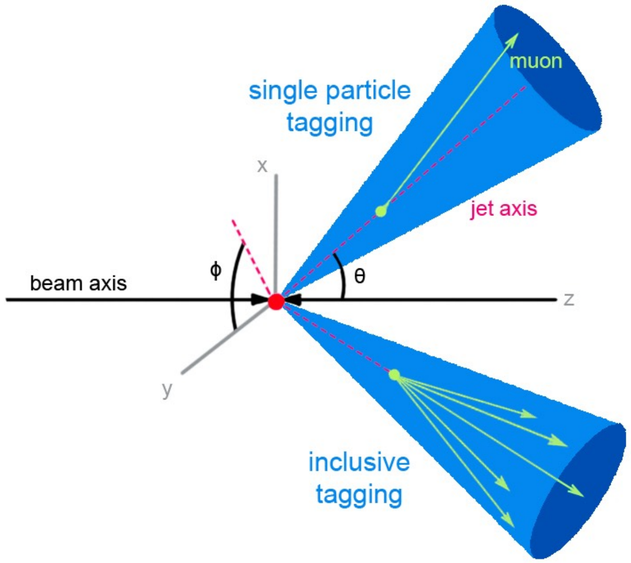
“Quantum-inspired Machine Learning on high-energy physics data”

Marco Trenti, Lorenzo Sestini, Alessio Gianelle, Davide Zuliani, Timo Felser, Donatella Lucchesi, Simone Montangero

arXiv:2004.13747

Data selection

- Monte Carlo samples from LHCb Open Data are used
- $b\bar{b}$ di-jets events at $13\ TeV$ are considered, with the following kinematic cuts:
  - $p_T > 20\ GeV$
  - $2.2 < \eta < 4.2$, where η is the pseudorapidity $\eta = -\log\left[\tan\left(\theta/2\right)\right]$
  - Both jets contain a Secondary Vertex (SV-tagging)



- Inside each jet $\mu^{\pm}$, $e^{\pm}$, $\pi^{\pm}$, $K^{\pm}$ and $p/\bar{p}$ with highest p_T are selected
- For each particle, three variables are considered: q, p_T^{rel} and ΔR where:
  - p_T^{rel} is the transverse momentum with respect to jet axis
  - ΔR is the distance between the particle and jet axis in the (η, ϕ) space
- Finally the jet charge $Q = \dfrac{\sum p_{T,i}^{rel} q_i}{\sum p_{T,i}^{rel}}$ is also considered

16 variables are used to describe jet substructure

“inclusive” jet tagging algorithm

Analysis main points

- A TTN and a Deep Neural Network (DNN) are used as classifiers
- The output of both methods is the probability $\mathscr{P}_b$ to classify a jet as generated by a b- or a $\bar{b}$-quark
  - For values of probability $\mathscr{P}_b > 0.5$ a jet is classified as generated by a b-quark
  - For values of probability $\mathscr{P}_b < 0.5$ a jet is classified as generated by a $\bar{b}$-quark

- The "figure of merit" for the tagging algorithm performance is the tagging power ε_{tag}

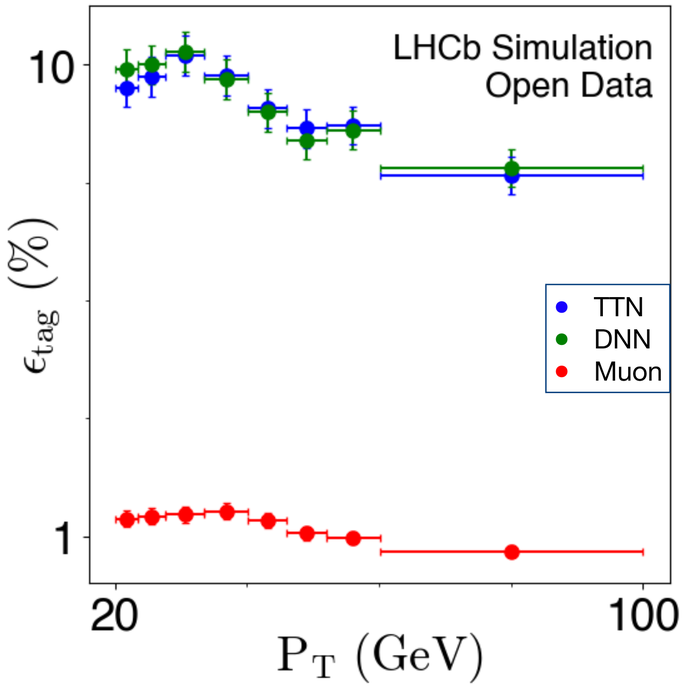
$$\varepsilon_{tag} = \varepsilon_{eff} \cdot (2a - 1)^2$$

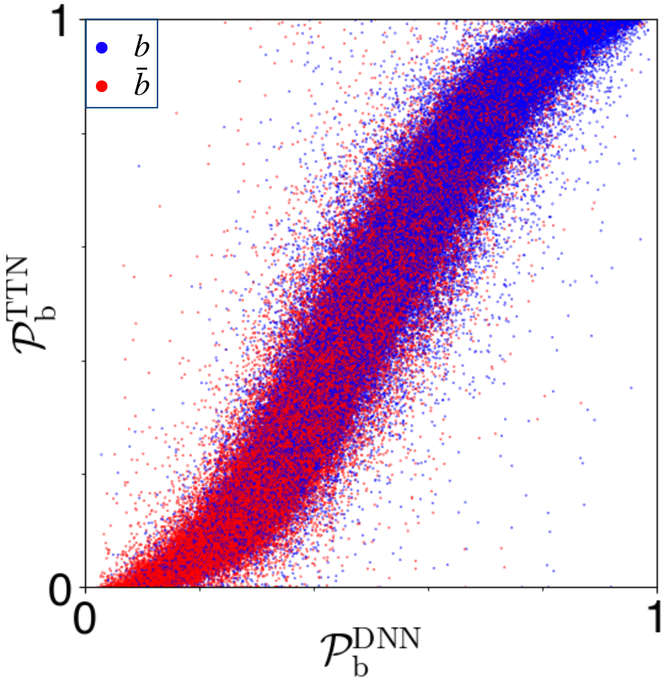
ε_{eff} = *efficiency*, fraction of jets where the classifier takes a decision

a = *accuracy*, fraction of jets where the classifier takes the **right** decision

- Both classifiers' performances are compared with the standard muon tagging approach
- Cuts are applied to the probability distribution to maximize the tagging power

Results (1)

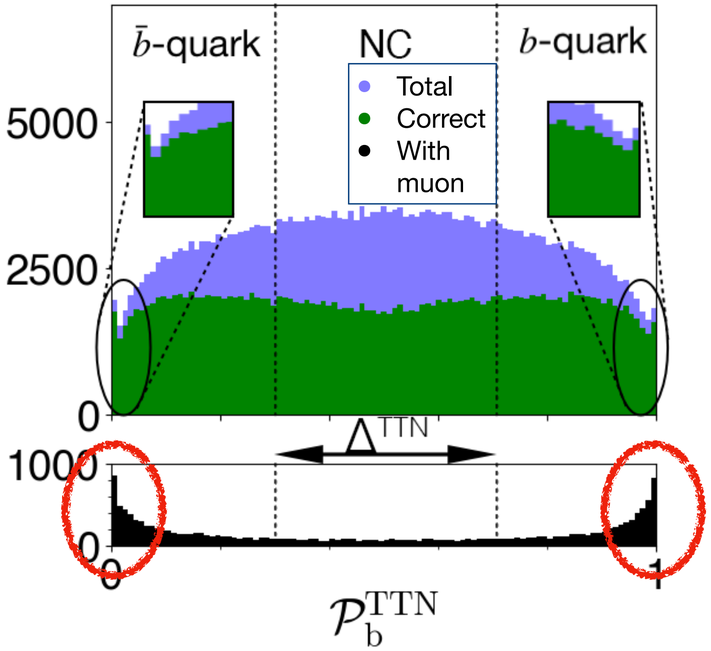


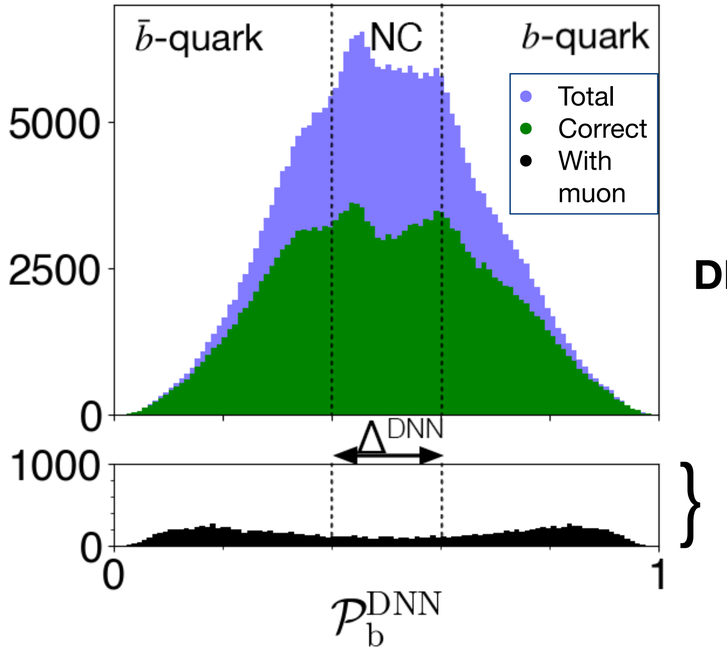


- Both ML approaches outperform the standard muon tagging approach by a factor ~ 10
- Better performances are obtained for lower jet p_T
- **Both TTN and DNN have similar performances as a function of jet p_T**
- The output of the classifiers are greatly correlated
- Test on physical variables (p_T and η distributions): no biases found

Results (2)

TTN probability distribution





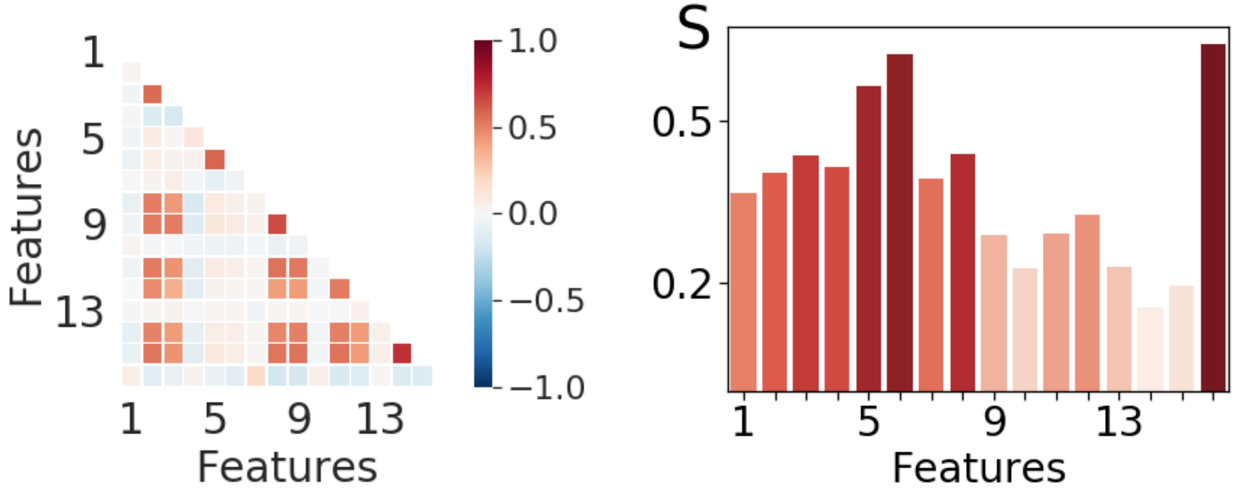
DNN probability distribution

} Probability distribution for jets with a muon inside

- Despite the similar performances, we obtain different distributions for the two classifiers
- Applied cuts (Δ^{TTN} and Δ^{DNN}) to maximize the tagging power are shown
- **The TTN is able to spot the presence of a muon inside the jet (peaks at $\mathcal{P}_b = 0,1$)**
- The DNN lacks these confident predictions

Insights on data with TTN

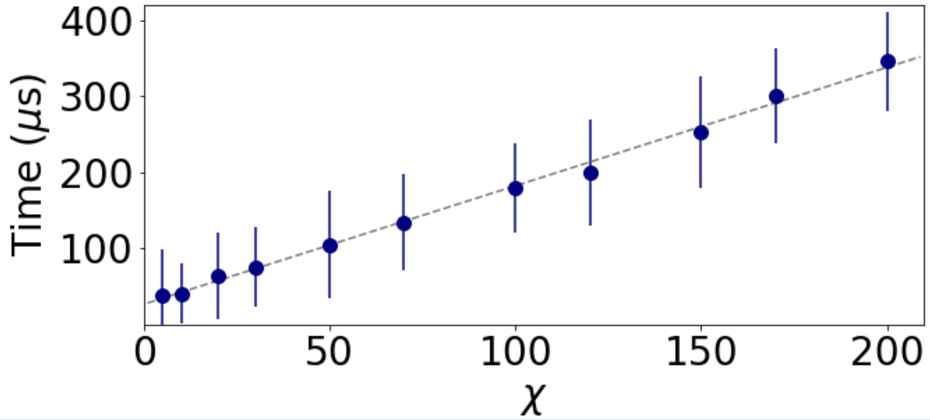
- TTN allows to measure **correlations** and **entanglement** within the classifier
- The most important features are selected for the classifications:
  - If two features are **highly correlated**, it is possible to **neglect** at least one of them
  - If the **entropy** of a set of features is **low**, all features from that set can be **discarded**



- This algorithm is called *Quantum Information Post-learning feature Selection* (QuIPS)
- The **8 most important variables** (“best 8”) are selected and compared with the full 16 variables model

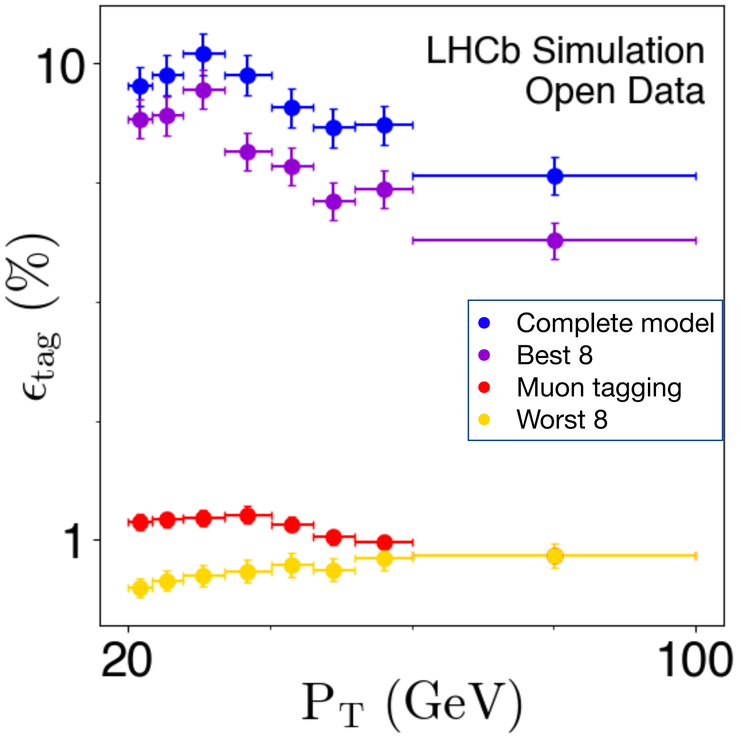
Insights on data with TTN

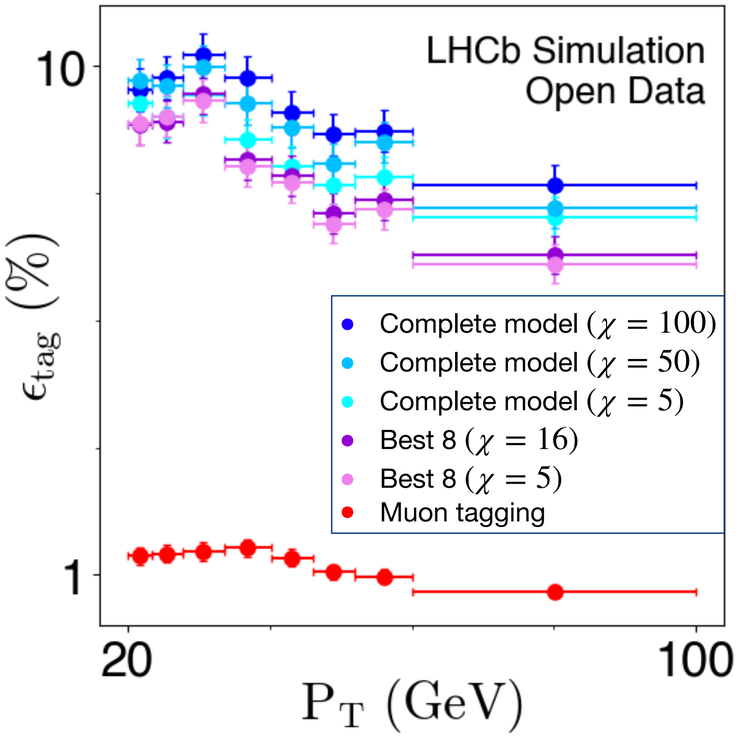
- Prediction time plays an important role in data classification
- In the TTN context it is possible to modify the bond dimension χ, as to target a specific prediction time
- **Key point: this can be done without retraining the TTN**
- This algorithm is called *Quantum Information Adaptive Network Optimization* (QIANO)
  - The TTN is trained with a maximum bond dimension χ_{max}
  - After the training the TTN is truncated to a specific $\chi < \chi_{max}$ via Singolar Value Decomposition
  - **The critical amount of information is kept**
  - The truncated TTN can classify data in **lower computational time**



Results

Just applying the QuIPS algorithm





Applying both the QuIPS and QIANO algorithms

- Both QuIPS and QIANO algorithms are applied to the 16-variables **complete model**
- By selecting the **best 8** variables via the QuIPS algorithm we lose only ~1% of accuracy
- Applying QIANO algorithm results in a reduction of average prediction time t_{pred} from $345\ \mu s$ to $37\ \mu s$
- Applying both QuIPS and QIANO: $t_{pred} \sim 19\ \mu s$ and still compatible accuracy

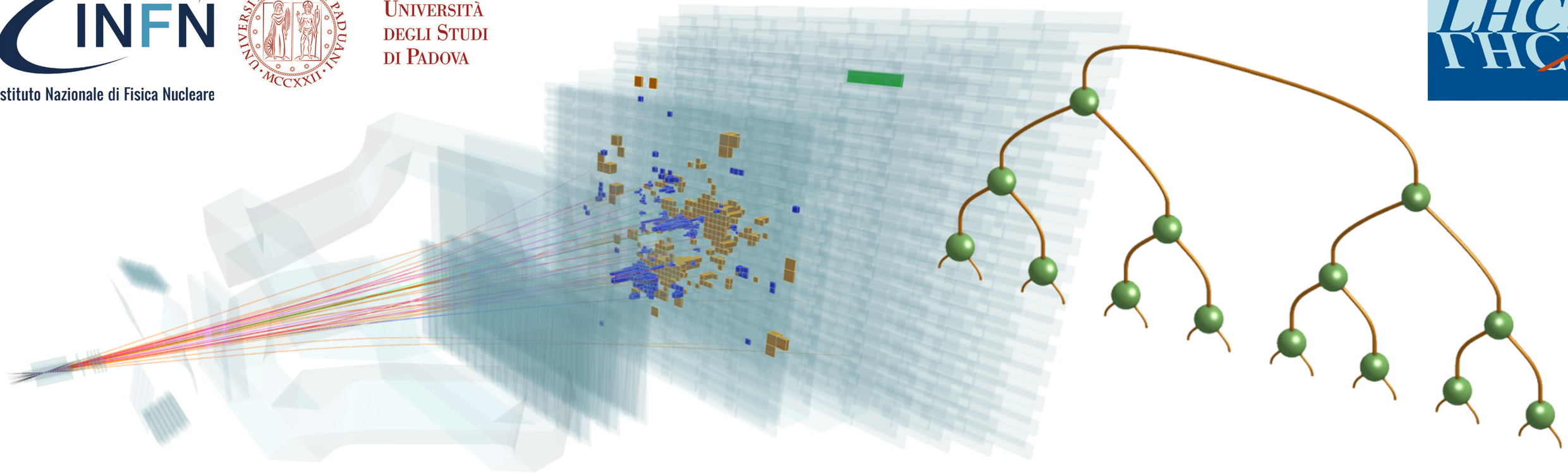
Planning to parallelize using GPUs: speed-up ~ 10x-100x

Conclusions

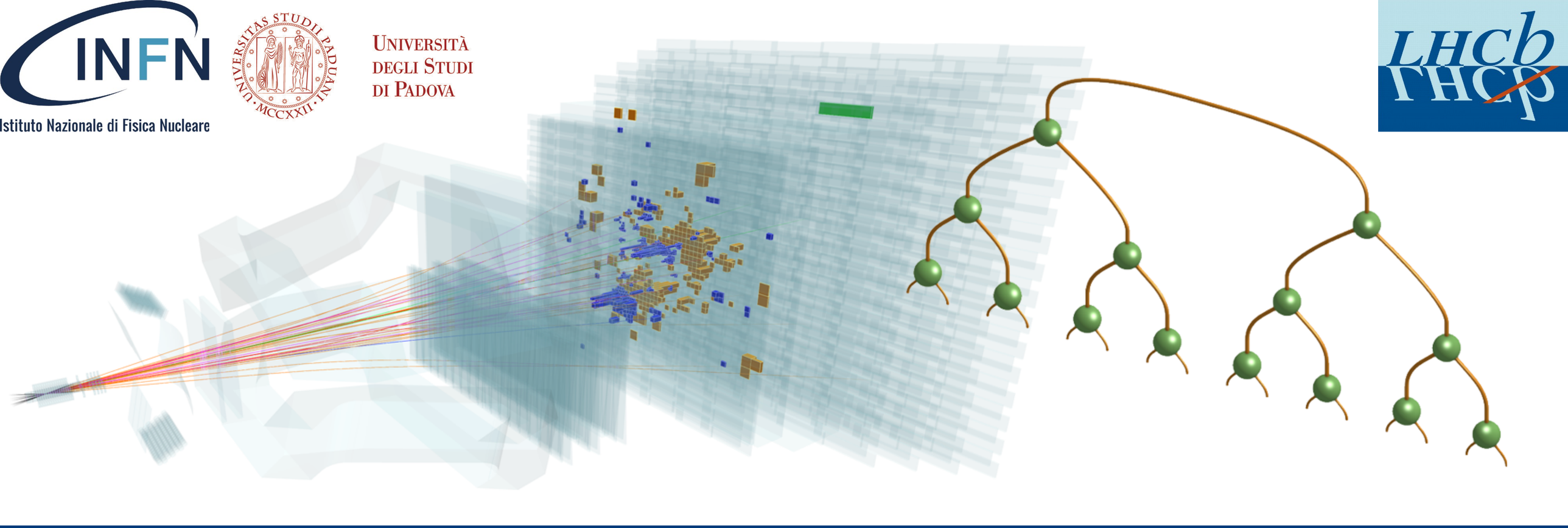
- New ML algorithms are required to analyze LHC data, particularly in future runs
- TTNs are a suited method for supervised ML problems, such as jet flavour tagging
  - **Comparable performances w.r.t. DNNs** and outperform standard jet tagging algorithms
  - Measure **correlations** and **entropy** between input variables
  - **Lower prediction times** due to truncation of TTN after the training
- For the future…
  - Next more complex task: b vs. c jet flavour tagging
  - Real time application

Istituto Nazionale di Fisica Nucleare

Università degli Studi di Padova



Thank you for your attention!



Backup slides