npj | Quantum Information www.nature.com/npjqi

ARTICLE OPEN

Quantum-inspired machine learning on high-energy physics data

Timo Felser[1,2,3,4✉], Marco Trenti[1,2], Lorenzo Sestini[3], Alessio Gianelle[3], Davide Zuliani[2,3], Donatella Lucchesi[2,3] and Simone Montangero[2,3,5]

Tensor Networks, a numerical tool originally designed for simulating quantum many-body systems, have recently been applied to solve Machine Learning problems. Exploiting a tree tensor network, we apply a quantum-inspired machine learning technique to a very important and challenging big data problem in high-energy physics: the analysis and classification of data produced by the Large Hadron Collider at CERN. In particular, we present how to effectively classify so-called b-jets, jets originating from b-quarks from proton–proton collisions in the LHCb experiment, and how to interpret the classification results. We exploit the Tensor Network approach to select important features and adapt the network geometry based on information acquired in the learning process. Finally, we show how to adapt the tree tensor network to achieve optimal precision or fast response in time without the need of repeating the learning process. These results pave the way to the implementation of high-frequency real-time applications, a key ingredient needed among others for current and future LHCb event classification able to trigger events at the tens of MHz scale.

npj Quantum Information (2021)7:111 ; https://doi.org/10.1038/s41534-021-00443-w

INTRODUCTION

Artificial Neural Networks (NN) are a well-established tool for applications in Machine Learning and they are of increasing interest in both research and industry[1–6]. Inspired by biological NN, they are able to recognise patterns while processing a huge amount of data. In a nutshell, a NNs describes a functional mapping containing many variational parameters, which are optimised during the training procedure. Recently, deep connections between Machine Learning and quantum physics have been identified and continue to be uncovered[7]. On one hand, NNs have been applied to describe the behaviour of complex quantum many-body systems[8–10] while, on the other hand, quantum-inspired technologies and algorithms are taken into account to solve Machine Learning tasks[11–13].

One particular numerical method originated from quantum physics which has been increasingly compared to NNs are Tensor Networks (TNs)[14–16]. TNs have been developed to investigate quantum many-body systems on classical computers by efficiently representing the exponentially large quantum wavefunction $|\psi\rangle$ in a compact form and they have proven to be an essential tool for a broad range of applications[17–26]. The accuracy of the TN approximation can be controlled with the so-called *bond-dimension* χ, an auxiliary dimension for the indices of the connected local tensors. Recently, it has been shown that TN methods can also be applied to solve Machine Learning (ML) tasks very effectively[13,16,27–31]. Indeed, even though NNs have been highly developed in recent decades by industry and research, the first approaches of ML with TN yield already comparable results when applied to standard datasets[13,27,32]. Due to their original development focusing on quantum systems, TNs allow to easily compute quantities such as quantum correlations or entanglement entropy and thereby they grant access to insights on the learned data from a distinct point of view for the application in ML[16,30]. Hereafter, we demonstrate the effectiveness of the approach and, more importantly, that it allows introducing algorithms to simplify and explain the learning process, unveiling a pathway to an explainable Artificial Intelligence. As a potential application of this approach, we present a TN supervised learning of identifying the charge of *b*-quarks (i.e. *b* or $\bar{b}$) produced in high-energy proton–proton collisions at the Large Hadron Collider (LHC) accelerator at CERN.

In what follows, we first describe the quantum-inspired Tree Tensor Network (TTN) and introduce different quantities that can be extracted from the TTN classifier which are not easily accessible for the biological-inspired Deep NN (DNN), such as correlation functions and entanglement entropy which can be used to explain the learning process and subsequent classifications, paving the way to an efficient and transparent ML tool. In this regard, we introduce the Quantum-Information Post-learning feature Selection (QuIPS), a protocol that reduces the complexity of the ML model based on the information the single features provide for the classification problem. We then briefly describe the LHCb experiment and its simulation framework, the main observables related to *b*-jets physics, and the relevant quantities for this analysis together with the underlying LHCb data[33,34]. We further compare the performance obtained by the DNN and the TTN, before presenting the analytical insights into the TTN which, among others, can be exploited to improve future data analysis of high-energy problems for a deeper physical understanding of the LHCb data. Moreover, we introduce the Quantum-Information Adaptive Network Optimisation (QIANO), which adapts the TN representation by reducing the number of free parameters based on the captured information within the TN while aiming to maintain the highest accuracy possible. Therewith, we can optimise the trained TN classifier for a targeted prediction speed without the necessity to relearn a new model from scratch.

[1]Tensor Solutions, Institute for Complex Quantum Systems, University of Ulm, Ulm, Germany. [2]Dipartimento di Fisica e Astronomia G. Galilei, Università di Padova, Padova, Italy. [3]INFN, Sezione di Padova, Padova, Italy. [4]Theoretische Physik, Universität des Saarlandes, Saarbrücken, Germany. [5]Padua Quantum Technologies Research Center, Università degli Studi di Padova, Padova, Italy. ✉email: timo.felser@tensor-solutions.com

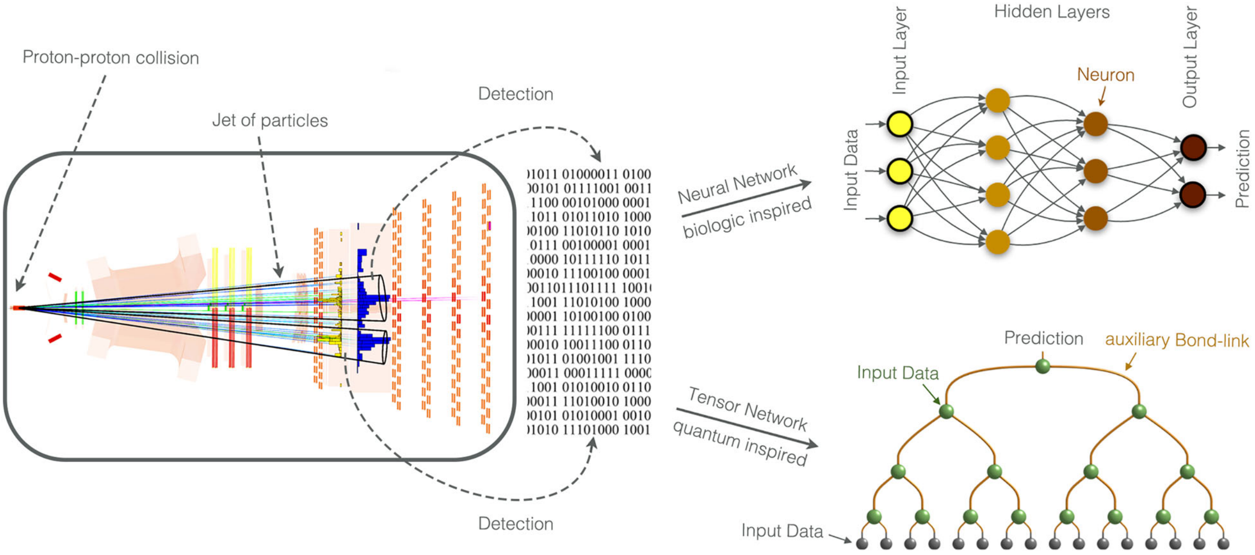

Fig. 1 Data flow of the Machine Learning analysis for the *b*-jet classification of the LHCb experiment at CERN. After proton–proton collisions, b- and $\bar{b}$-quarks are created, which subsequently fragment into particle jets (left). The different particles within the jets are tracked by the LHCb detector. Selected features of the detected particle data are used as input for the Machine Learning analysis by NNs and TNs in order to determine the charge of the initial quark (right).

TNs are not only a well-established way to represent a quantum wavefunction $|\psi\rangle$, but more general an efficient representation of information as such. In the mathematical context, a TN approximates a high-order tensor by a set of low-order tensors that are contracted in a particular underlying geometry and have common roots with other decompositions, such as the Singular Value Decomposition (SVD) or Tucker decomposition[35]. Among others, some of the most successful TN representations are the Matrix Product State—or Tensor Trains[18,27,36,37], the TTN—or Hierarchical Tucker decomposition[30,38,39], and the Projected Entangled Pair States[40,41].

For a supervised learning problem, a TN can be used as the *weight tensorW*[13,27,30], a high-order tensor which acts as classifier for the input data $\{\mathbf{x}\}$: Each sample $\mathbf{x}$ is encoded by a *feature map* $\Phi(\mathbf{x})$ and subsequently classified by the *weight tensor* W. The final confidence of the classifier for a certain class labelled by l is given by the probability

$$\mathcal{P}_l(\mathbf{x}) = W_l \cdot \Phi(\mathbf{x})\,. \tag{1}$$

In the following, we use a TTN Ψ to represent W (see Fig. 1, bottom right) which can be described as a contraction of its N hierarchically connected local tensors $T_{\{\chi\}}$

$$\Psi = \sum_{X} T^{[1]}_{l,X_1,X_2} \prod_{\eta=2}^{N} T^{[\eta]}_{X_n,X_{2n},X_{2n+1}} \tag{2}$$

where $n \in [1, N]$. Therefore, we can interpret the TTN classifier Ψ as well as a set of quantum many-body wavefunctions $|\psi_l\rangle$—one for each of the class labels l (see Supplementary Methods). For the classification, we represent each sample x by a product state $\Phi(x)$. Therefore, we map each feature $x_i \in \mathbf{x}$ into a quantum spin by choosing the *feature map* $\Phi(\mathbf{x})$ as a Kronecker product of $N+1$ *local feature maps*

$$\Phi^{[i]}(x_i) = \left[\cos\left(\frac{\pi x'}{2}\right), \sin\left(\frac{\pi x'_i}{2}\right)\right] \tag{3}$$

where $x' \equiv x_i/x_{i,\max} \in [0, 1]$ is the re-scaled value with respect to the maximum $x_{i,\max}$ within all samples of the training set.

Accordingly, we classify a sample x by computing the overlap $\langle\Phi(x)|\psi_l\rangle$ for all labels l with the product state $\Phi(x)$ resulting in the weighted probabilities

$$\mathcal{P}_l = \frac{|\langle\Phi(x)|\psi_l\rangle|^2}{\sum_l |\langle\Phi(x)|\psi_l\rangle|^2} \tag{4}$$

for each class. We point out, that we can encode the input data in different non-linear feature maps as well (see Supplementary Notes).

One of the major benefits of TNs in quantum mechanics is the accessibility of information within the network. They allow to efficiently measure information quantities such as entanglement entropy and correlations. Based on these quantum-inspired measurements, we here introduce the QuIPS protocol for the TN application in ML, which exploits the information encoded and accessible in the TN in order to rank the input features according to their importance for the classification.

In information theory, entropy as such is a measure of the information content inherent in the possible outcomes of variables, such as e.g. a classification[42–44]. In TNs such information content can be assessed by means of the entanglement entropy S which describes the shared information between TN bipartitions. The entanglement S is measured via the Schmidt decomposition, that is, decomposing $|\psi\rangle$ into two bipartitions $|\psi^A_\alpha\rangle$ and $|\psi^B_\alpha\rangle$[44] such that

$$\Psi = \sum_{\alpha}^{\chi} \lambda_\alpha |\Psi^A_\alpha\rangle \otimes |\Psi^B_\alpha\rangle, \tag{5}$$

where λ_α are the Schmidt-coefficients (non-zero, normalised singular values of the decomposition). The entanglement entropy is then defined as $S = -\sum_\alpha \lambda^2_\alpha \ln \lambda^2_\alpha$. Consequently, the minimal entropy $S = 0$ is obtained only if we have one single non-zero singular value $\lambda_1 = 1$. In this case, we can completely separate the two bipartitions as they share no information. On the contrary, higher S means that information is shared among the bipartitions.

In the ML context, the entropy can be interpreted as follows: If the features in one bipartition provide no valuable information for the classification task, the entropy is zero. On the contrary, S increases the more information between the two bipartitions are exploited. This analysis can be used to optimise the learning procedure: whenever $S = 0$, the feature can be discarded with no loss of information for the classification. Thereby, a second model with fewer features and fewer tensors can be introduced. This second, more efficient model results in the same predictions in less time. On the contrary, a high bipartition entropy highlights which feature—or combination of features—are important for the correct predictions.

The second set of measurements we take into account are the correlation functions

$$C^{l}_{i,j} = \frac{\langle\psi_l|\sigma^z_i\sigma^z_j|\psi_l\rangle}{\langle\psi_l|\psi_l\rangle} \tag{6}$$

for each pair of features (located at site i and j) and for each class l. The correlations offer an insight into the possible relation among the information that the two features provide. In case of maximum correlation or anti-correlation among them for all classes l, the information of one of the features can be obtained by the other one (and vice versa), thus one can be neglected. In case of no correlation among them, the two features may provide fundamentally different information for the classification. The correlation analysis allows pinpointing if two features give independent information. However, the correlation itself—in contrast to the entropy—does not tell if this information is important for the classification.

In conclusion, based on the previous insights, namely: (i) a low entropy of a feature bipartition signals that one of the two bipartitions can be discarded, providing negligible loss of information and (ii) if two features are completely (anti-)correlated we can neglect at least one of them, the QuIPS enables to filter out the most valuable features for the classification.

Nowadays, in particle physics, ML is widely used for the classification of jets, i.e. streams of particles produced by the fragmentation of quarks and gluons. The jet substructure can be exploited to solve such classification problems[45]. ML techniques have been proposed to identify boosted, hadronically decaying top quarks[46], or to identify the jet charge[47]. The ATLAS and CMS collaborations developed ML algorithms in order to identify jets generated by the fragmentation of b-quarks[48–50]: a comprehensive review on ML techniques at the LHC can be found in[51].

The LHCb experiment in particular is, among others, dedicated to the study of the physics of b- and c-quarks produced in proton–proton collisions. Here, ML methods have been introduced recently for the discrimination between b- and c-jets by using Boosted Decision Tree classifiers[52]. However, a crucial topic for the LHCb experiment, which is yet unexploited by ML, is the identification of the charge of a b-quark, i.e. discriminating between a b or $\overline{b}$. Such identification can be used in many physics measurements, and it is the core of the determination of the charge asymmetry in b-pairs production, a quantity sensitive to physics beyond the Standard Model[53]. Whenever produced in a scattering event, b-quarks have a short lifetime as free particles; indeed, they manifest themselves as bound states (hadrons) or as narrow cones of particles produced by the hadronization (jets). In the case of the LHCb experiment, the b-jets are detected by the apparatus located in the forward region of proton–proton collisions (see Fig. 1, left)[54]. The LHCb detector includes a particle identification system that distinguishes different types of charged particles within the jet, and a high-precision tracking system able to measure the momentum of each particles[55]. Still, the separation between b- and $\overline{b}$-jets is a highly difficult task because the b-quark fragmentation produce dozens of particles via non-perturbative Quantum Chromodynamics processes, resulting in non-trivial correlations between the jet particles and the original quark.

The algorithms used to identify the charge of the b-quarks based on information on the jets are called tagging methods. The tagging algorithm performance is typically quantified with the *tagging power* ϵ_{tag}, representing the effective fraction of jets that contribute to the statistical uncertainty in an asymmetry measurement[56,57]. In particular, the tagging power ϵ_{tag} takes into account the efficiency ϵ_{eff} (the fraction of jets for which the classifier takes a decision) and the prediction accuracy a (the fraction of correctly classified jets among them) as follows:

$$\epsilon_{tag} = \epsilon_{eff} \cdot (2a - 1)^2 . \tag{7}$$

To date, the muon tagging method gives the best performance on the b- vs. $\overline{b}$-jet discrimination using the dataset collected in the LHC Run I[58]: here, the muon with the highest momentum in the jet is selected, and its electric charge is used to decide on the b-quark charge.

For the ML application, we now formulate the identification of the b-quark charge in terms of a supervised learning problem. As described above, we implemented a TTN as a classifier and applied it to the LHCb problem analysing its performance. Alongside, a DNN analysis is performed to the best of our capabilities, and both algorithms are compared with the muon tagging approach. Both the TTN and the DNN, use as input for the supervised learning 16 features of the jet substructure from the official simulation data released by the LHCb collaboration[33,34]. The 16 features are determined as follows: the muon with the highest p_T among all other detected muons in the jet is selected and the same is done for the highest p_T kaon, pion, electron, and proton, resulting in 5 different selected particles. For each particle, three observables are considered: (i) The momentum relative to the jet axis (p_T^{rel}), (ii) the particle charge (q), and (iii) the distance between the particle and the jet axis (ΔR), for a total of 5×3 observables. If a particle type is not found in a jet, the related features are set to 0. The 16th feature is the total jet charge Q, defined as the weighted average of the particles charges q_i inside the jet, using the particles p_T^{rel} as weights:

$$Q = \frac{\sum_i (p_T^{rel})_i q_i}{\sum_i (p_T^{rel})_i} . \tag{8}$$

RESULTS

Analysis framework

In the following, we present the jet classification performance for the TTN and the DNN applied to the LHCb dataset, also comparing both ML techniques with the muon tagging approach. For the DNN we use an optimised network with three hidden layers of 96 nodes (see Supplementary Methods for details). Hereafter, we aim to compare the best possible performance of both approaches therefore, we optimised the hyperparameters of both methods in order to obtain the best possible results from each of them, TTN and DNN. Therefore, we split the dataset of about 700k events (samples) into two sub-sets: 60% of the samples are used in the training process while the remaining 40% are used as test set to evaluate and compare the different methods. For each event prediction after the training procedure, both ML models output the probability $\mathcal{P}_b$ to classify the event as a jet generated by a b- or a $\overline{b}$-quark. A threshold Δ around $\mathcal{P}_b = 0.5$ is then defined in which we classify the quark as unknown in order to optimise the overall tagging power ϵ_{tag}.

Jet classification performance

We obtain similar performances in terms of the raw prediction accuracy applying both ML approaches after the training procedure on the test data: the TTN takes a decision on the charge of the quark in $\epsilon_{eff}^{TTN} = 54.5\%$ of the cases with an overall accuracy of $a^{TTN} = 70.56\%$, while the DNN decides in $\epsilon_{eff}^{DNN} = 55.3\%$ of the samples with $a^{DNN} = 70.49\%$. We checked both approaches for biases in physical quantities to ensure that both methods are able to properly capture the physical process behind the problem and thus that they can be used as valid tagging methods for LHCb events (see Supplementary Methods).

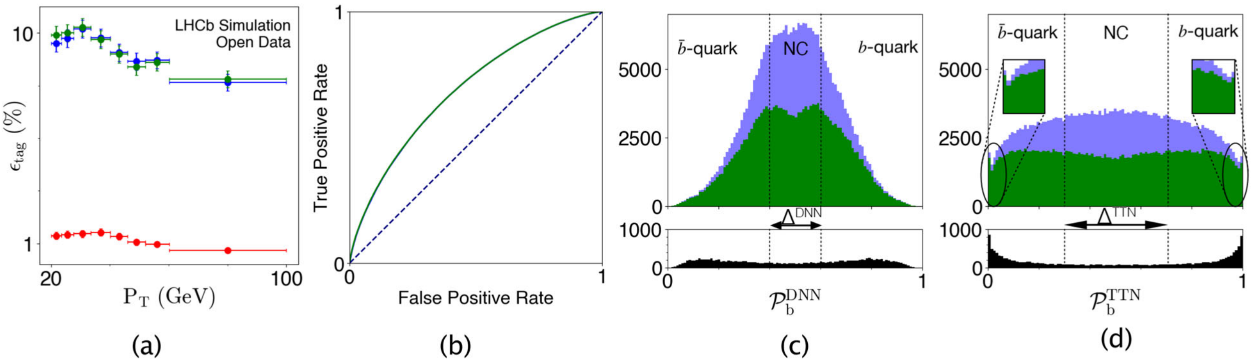

Fig. 2 Comparison of the DNN and TNN analysis. a Tagging power for the DNN (green), TTN (blue) and the muon tagging (red), (**b**) ROC curves for the DNN (green) and the TTN (blue, but completely covered by DNN), compared with the *line of no-discrimination* (dotted navy-blue line), (**c**) probability distribution for the DNN and (**d**) for the TTN. In the two distributions (**c**, **d**), the correctly classified events (green) are shown in the total distribution (light blue). Below, in black all samples where a muon was detected in the jet.

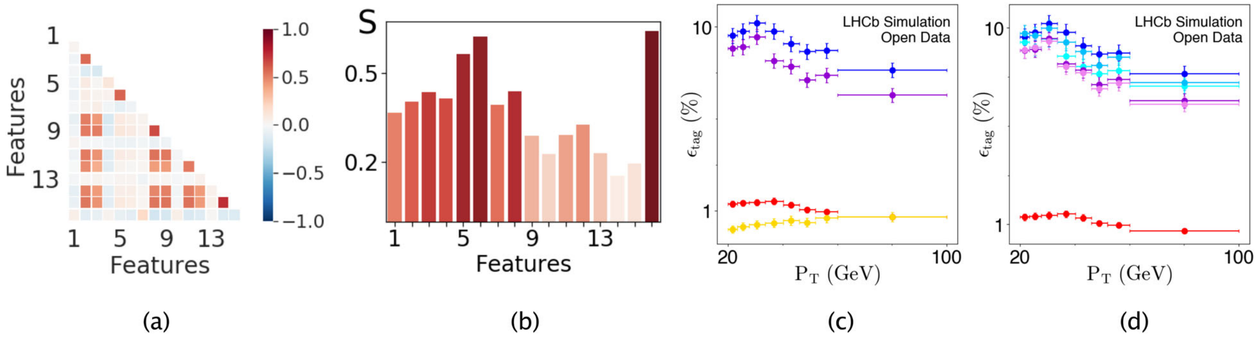

Fig. 3 Exploiting the information provided by the learned TTN classifier. a Correlations between the 16 input features (blue for anti-correlated, white for uncorrelated, red for correlated). The numbers indicate q, p_T^{rel}, ΔR of the muon (1–3), kaon (4–6), pion (7–9), electron (10–12), proton (13–15) and the jet charge Q (16). **b** Entropy of each feature as the measure for the information provided for the classification. **c** Tagging power for learning on all features (blue), the best eight proposed by QuIPS exploiting insights from (**a**, **b**) (magenta), the worst eight (yellow) and the muon tagging (red). **d** Tagging power for decreasing bond-dimension truncated after training: The complete model (blue shades for $\chi = 100$, $\chi = 50$, $\chi = 5$), for using the QuIPS best 8 features only (violet shades for $\chi = 16$, $\chi = 5$), and the muon tagging (red).

In Fig. 2a we present the tagging power of the different approaches as a function of the jet transverse momentum p_T. Evidently, both ML methods perform significantly better than the muon tagging approach for the complete range of jet transverse momentum p_T, while the TTN and DNN display comparable performances within the statistical uncertainties.

In Fig. 2c, d we present the histograms of the confidences for predicting a b-flavoured jet for all samples in the test dataset for the DNN and the TTN respectively. Interestingly, even though both approaches give similar performances in terms of overall precision and tagging power, the prediction confidences are fundamentally different. For the DNN, we see a Gaussian-like distribution with, in general, not very high confidence for each prediction. Thus, we obtain less correct predictions with high confidences, but at the same time, fewer wrong predictions with high confidences compared to the TTN predictions. On the other hand, the TTN displays a flatter distribution including more predictions—correct and incorrect—with higher confidence. Remarkably though, we can see peaks for extremely confident predictions (around 0 and around 1) for the TTN. These peaks can be traced back to the presence of the muon; noting that the charge of which is a well-defined predictor for a jet generated by a b-quark. The DNN lacks these confident predictions exploiting the muon charge. Further, we mention that using different cost functions for the DNN, i.e. cross-entropy loss function and the Mean Squared Error, lead to similar results (see Supplementary Methods).

Finally, in Fig. 2b we present the Receiving Operator Characteristic (ROC) curves for the TTN and the DNN together with the *line of no-discrimination*, which represents a randomly guessing classifier: the two ROC curves for TTN and DNN are perfectly coincident, and the Area Under the Curve (AUC) for the two classifiers is the almost same ($AUC^{TTN} = 0.689$ and $AUC^{DNN} = 0.690$). The graph illustrates the similarity in the outputs between TTN and DNN despite the different confidence distributions. This is further confirmed by a Pearson correlation factor of $r = 0.97$ between the outputs of the two classifiers.

In conclusion, the two different approaches result in similar outcomes in terms of prediction performances. However, the underlying information used by the two discriminators is inherently different. For instance, the DNN predicts more conservatively, in the sense that the confidences for each prediction tend to be lower compared with the TTN. Additionally, the DNN does not exploit the presence of the muon as strongly as the TTN, even though the muon is a good predictor for the classification.

Exploiting insights into the data with TTN

As previously mentioned, the TTN analysis allows to efficiently measure the captured correlations and the entanglement within the classifier. These measurements give insight into the learned data and can be exploited via QuIPS to identify the most important features typically used for the classifications.

In Fig. 3a we present the correlation analysis allowing us to pinpoint if two features give independent information. For both labels ($l = b, \overline{b}$) the results are very similar, thus in Fig. 3a we present only $l = b$. We see among others that the momenta p_T^{rel} and distance ΔR of all particles are correlated except for the kaon.

Table 1. TTN prediction time.

	Model M_{16} (incl. all 16 features)			Model B_8 (best 8 features determined by QuIPS)		
χ	Prediction time	Accuracy	Free parameters	Prediction time	Accuracy	Free parameters
200	345 μs	70.27% (63.45%)	51,501	–	–	–
100	178 μs	70.34% (63.47%)	25,968	–	–	–
50	105 μs	70.26% (63.47%)	13,214	–	–	–
20	62 μs	70.31% (63.46%)	5576	–	–	–
16	–	–	–	19 μs	69.10% (62.78%)	264
10	40 μs	70.36% (63.44%)	1311	19 μs	69.01% (62.78%)	171
5	37 μs	69.84% (62.01%)	303	19 μs	69.05% (62.76%)	95

Prediction time, accuracy with (and without) applied cuts Δ and number of free parameters of the TTN for different bond-dimension χ when we reduce the TTN model with QIANO, both for the complete 16 (left) and the QuIPS reduced 8 features (right). For the model M_{16} with all 16 features (left), we trained the TTN with $\chi = 200$ and truncate from there while for the reduced model B_8 (right), the original bond-dimension was $\chi = 16$ (being the maximum χ in this subspace).

Thus this particle provides information to the classification which seems to be independent of the information gained by the other particles. However, the correlation itself does not tell if this information is important for the classification. Thus, we compute the entanglement entropy S of each feature, as reported in Fig. 3b. Here, we conclude that the features with the highest information content are the total charge and p_T^{rel} and distance ΔR of the kaon. Driven by these insights, we employ the QuIPS to discard half of the features by selecting the eight most important ones: i.–iii. charge, momenta and distance of the muon, iv.–vi. charge, momenta and distance of the kaon, vii. charge of the pion and viii. total detected charge. To test the QuIPS performance, we compared it with an independent but more time-expensive analysis on the importance of the different particle types: the two approaches perfectly matched. Further, we studied two new models, one composed of the eight most important features proposed by the QuIPS, and, for comparison, another with the eight discarded features. In Fig. 3c we show the tagging power for the different analysis with the complete 16-sites (model M_{16}), the best 8 (B_8), the worst 8 (W_8) and the muon tagging. Remarkably, we see that the models M_{16} and B_8 give comparable results, while model W_8 results are even worse than the classical approach. These performances are confirmed by the prediction accuracy of the different models: While only less than 1% of accuracy is lost from M_{16} to B_8, the accuracy of the model W_8 drastically drops to around 52%—that is, almost random predictions. Finally, in this particular run, the model B_8 has been trained 4.7 times faster with respect to model M_{16} and predicts 5.5 times faster as well (The actual speed-up depends on the bond-dimension and other hyperparameters).

A critical point of interest in real-time ML applications is the prediction time. For example, in the LHCb Run 2 data-taking, the high-level software trigger takes a decision approximately every 1 μs[55] and shorter latencies are expected in future Runs. Consequently, with the aid of the QuIPS protocol, we can efficiently reduce the prediction computational time while maintaining a comparable high prediction power. However, with TTNs, we can undertake an even further step to reduce the prediction time by reducing the bond-dimension χ after the training procedure. Here, we introduce the QIANO performing this truncation by means of the well-established SVD for TN[18,23,25] in a way ensuring to introduce the least infidelity possible. In other words, QIANO can adjust the bond-dimension χ to achieve a targeted prediction time while keeping the prediction accuracy reasonably high. We stress that this can be done without relearning a new model, as would be the case with NN.

Finally, we apply QuIPS and QIANO to reduce the information in the TTN in an optimal way for a targeted balance between prediction time and accuracy. In Fig. 3d we show the tagging power taking the original TTN and truncate it to different bond dimensions χ. We can see, that even though we compress quite heavily, the overall tagging power does not change significantly. In fact, we only drop about 0.03% in the overall prediction accuracy, while at the same time improving the average prediction time from 345 to 37 μs (see Table 1). Applying the same idea to the model B_8 we can reduce the average prediction time effectively down to 19 μs on our machines, a performance compatible with current real-time classification rate.

DISCUSSION

We analysed an LHCb dataset for the classification of b- and $\overline{b}$-jets with two different ML approaches, a DNN and a TTN. We showed that we obtained with both techniques a tagging power about one order of magnitude higher than the classical muon tagging approach, which up to date is the best-published result for this classification problem. We pointed out that, even though both approaches result in similar tagging power, they treat the data very differently. In particular, TTN effectively recognises the importance of the presence of the muon as a strong predictor for the jet classification. Here, we point out that we only used a conjugate gradient descent for the optimisation of our TTN classifier. Deploying more sophisticated optimisation procedures which have already been proven to work for Tensor Trains, such as stochastic gradient descent[59] or Riemannian optimisation[28], may further improve the performance (in both time and accuracy) in future applications.

We further explained the crucial benefits of the TTN approach over the DNNs, namely (i) the ability to efficiently measuring correlations and the entanglement entropy, and (ii) the power of compressing the network while keeping a high amount of information (to some extend even lossless compression). We showed how the former quantum-inspired measurements help to set up a more efficient ML model: in particular, by introducing an information-based heuristic technique, we can establish the importance of single features based on the information captured within the trained TTN classifier only. Using this insight, we introduced the QuIPS, which can significantly reduce the model complexity by discarding the least-important features maintaining high prediction accuracy. This selection of features based on their informational importance for the trained classifier is one major advantage of TNs targeting to effectively decrease training and prediction time. Regarding the latter benefit of the TTN, we introduced the QIANO, which allows to decrease the TTN prediction time by optimally decreasing its representative power based on information from the quantum entropy, introducing the least possible infidelity. In contrast to DNNs, with the QIANO we do not need to set up a new model and train it from scratch, but we can optimise the network post-learning adaptively to the specific conditions, e.g. the used CPU or the required prediction time of the final application.

Finally, we showed that using QuIPS and QIANO we can effectively compress the trained TTN to target a given prediction time. In particular, we decreased our prediction times from 345 to 19 µs. We stress that, while we only used one CPU for the predictions, in future application we might obtain a speed-up from 10 to 100 times by parallelising the tensor contractions on GPUs[60]. Thus, we are confident that it is possible to reach a MHz prediction rate while still obtaining results significantly better than the classical muon tagging approach. Here, we also point out that, for using this algorithm on the LHCb real-time data acquisition system, it would be necessary to develop custom electronic cards like FPGAs, or GPUs with an optimised architecture. Such solutions should be explored in the future.

Given the competitive performance of the presented TTN method at its application in high-energy physics, we envisage a multitude of possible future applications in high-energy experiments at CERN and in other fields of science. Future applications of our approach in the LHCb experiment may include the discrimination between b-jets, c-jets and light flavour jets[52]. A fast and efficient real-time identification of b- and c-jets can be the key point for several studies in high-energy physics, ranging from the search for the rare Higgs boson decay in two c-quarks, up to the search for new particles decaying in a pair of heavy-flavour quarks ($b\bar{b}$ or $c\bar{c}$).

METHODS

LHCb particle detection

LHCb is fully instrumented in the phase space region of proton–proton collisions defined by the pseudo-rapidity (η) range [2, 5], with η defined as

$$\eta = -\log\left[\tan\left(\frac{\theta}{2}\right)\right], \tag{9}$$

where θ is the angle between the particle momentum and the beam axis (see Fig. 4). The direction of particles momenta can be fully identified by η and by the azimuthal angle ϕ, defined as the angle in the plane transverse to the beam axis. The projection of the momentum in this plane is called transverse momentum (p_T). The energy of charged and neutral particles is measured by electromagnetic and hadronic calorimeters. In the following, we work with physics natural units.

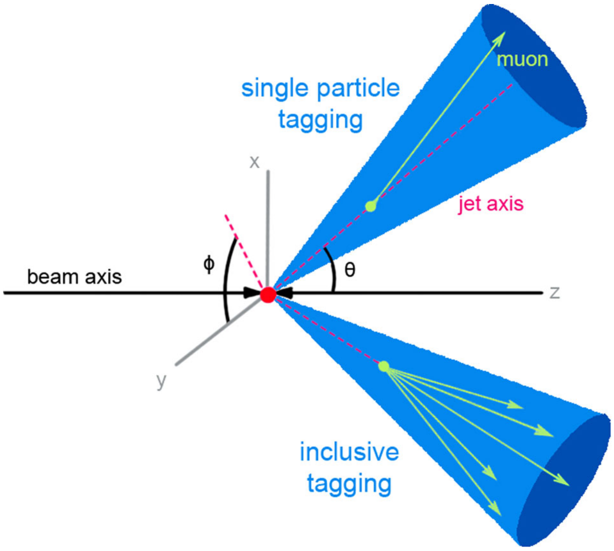

Fig. 4 Illustrative sketch showing an LHCb experiment and the two possible tagging algorithms. A single particle tagging algorithm, exploiting information coming from one single particle (muon), and the inclusive tagging algorithm which exploits the information on all the jet constituents.

At LHCb jets are reconstructed using a Particle Flow algorithm[61] for charged and neutral particles selection and using the anti-k_t algorithm[62] for clusterization. The jet momentum is defined as the sum of the momenta of the particles that form the jet, while the jet axis is defined as the direction of the jet momentum. Most of the particles that form the jet are contained in a cone of radius $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} = 0.5$, where $\Delta\eta$ and $\Delta\phi$ are respectively the pseudo-rapidity difference and the azimuthal angle difference between the particles momenta and the jet axis. For each particle inside the jet cone, the momentum relative to the jet axis (p_T^{rel}) is defined as the projection of the particle momentum in the plane transverse to the jet axis.

LHCb dataset

Differently from other ML performance analyses, the dataset used in this paper has been prepared specifically for this LHCb classification problem, therefore baseline ML models and benchmarks on it do not exist. In particle physics, features are strongly dependent on the detector considered (i.e. different experiments may have a different response on the same physical object) and for this reason the training has been performed on a dataset that reproduces the LHCb experimental conditions, in order to obtain the optimal performance with this experiment.

The LHCb simulation datasets used for our analysis are produced with a Monte Carlo technique using the framework GAUSS[63], which makes use of PYTHIA 8[64] to generate proton–proton interactions and jet fragmentation and uses EvtGen[65] to simulate b-hadrons decay. The GEANT4 software[66,67] is used to simulate the detector response, and the signals are digitised and reconstructed using the LHCb analysis framework.

The used dataset contains b and $\bar{b}$-jets produced in proton–proton collisions at a centre-of-mass energy of 13 TeV[33,34]. Pairs of b-jets and $\bar{b}$-jets are selected by requiring a jet p_T greater than 20 GeV and η in the range [2.2, 4.2] for both jets.

Muon tagging

LHCb measured the $b\bar{b}$ forward-central asymmetry using the dataset collected in the LHC Run I[58] using the muon tagging approach: In this method, the muon with the highest momentum in the jet cone is selected, and its electric charge is used to decide on the b-quark charge. In fact, if this muon is produced in the original semi-leptonic decay of the b-hadron, its charge is totally correlated with the b-quark charge. Up to date, the muon tagging method gives the best performance on the b- vs. $\bar{b}$-jet discrimination. Although this method can distinguish between b- and $\bar{b}$-quark with good accuracy, its efficiency is low as it is only applicable on jets where a muon is found and it is intrinsically limited by the b-hadrons branching ratio in semi-leptonic decays. Additionally, the muon tagging may fail in some scenarios, where the selected muon is produced not by the decay of the b-hadron but in other decay processes. In these cases, the muon may not be completely correlated with the b-quark charge.

Machine learning approaches

We train the TTN and analyse the data with different bond dimensions χ. The auxiliary dimension χ controls the number of free parameters within the variational TTN ansatz. While the TTN is able to capture more information from the training data with increasing bond-dimension χ, choosing χ too large may lead to overfitting and thus can worsen the results in the test set. For the DNN we use an optimised network with three hidden layers of 96 nodes (see Supplementary Methods for details).

For each event prediction, both methods give as output the probability $\mathcal{P}_b$ to classify a jet as generated by a b- or a $\bar{b}$-quark. This probability (i.e. the confidence of the classifier) is normalised in the following way: for values of probability $\mathcal{P}_b > 0.5$ ($\mathcal{P}_b < 0.5$) a jet is classified as generated by a b-quark ($\bar{b}$-quark), with an increasing confidence going to $\mathcal{P}_b = 1$ ($\mathcal{P}_b = 0$). Therefore a completely confident classifier returns a probability distribution peaked at $\mathcal{P}_b = 1$ and $\mathcal{P}_b = 0$ for jets classified as generated by b- and $\bar{b}$-quark respectively.

We introduce a threshold Δ symmetrically around the prediction confidence of $\mathcal{P}_b = 0.5$ in which we classify the event as unknown. We optimise the cut on the predictions of the classifiers (i.e. their confidences) to maximise the tagging power for each method based on the training samples. In the following analysis we find $\Delta^{\mathrm{TTN}} = 0.40$ ($\Delta^{\mathrm{DNN}} = 0.20$) for the TTN (DNN). Thereby, we predict for the TTN (DNN) a b-quark with confidences $\mathcal{P}_b > C^{\mathrm{TTN}} = 0.70$ ($\mathcal{P}_b > C^{\mathrm{DNN}} = 0.60$), a $\bar{b}$-quark with confidences $\mathcal{P}_b < 0.30$ ($\mathcal{P}_b < 0.40$) and no prediction for the range in between (see Fig. 2c, d).

DATA AVAILABILITY

This paper is based on data obtained by the LHCb experiment, but is analyzed independently, and has not been reviewed by the LHCb collaboration. The data are available in the official LHCb open data repository[33,34].

CODE AVAILABILITY

The software code used for the analysis of the Deep Neural Network can be freely acquired when contacting gianelle@pd.infn.it and it is permitted to use it for any kind of private or commercial usage including modification and distribution without any liabilities or warranties. The software code for the TTN analysis is currently not available for public use. For more information, please contact timo.felser@physik.uni-saarland.de.

Received: 26 October 2020; Accepted: 27 May 2021;
Published online: 15 July 2021

REFERENCES

1. Bishop, C. M. *Neural Networks for Pattern Recognition* (Oxford University Press, 1996).
2. Haykin, S. S. et al. *Neural networks and learning machines*, vol. 3 (Pearson, 2009).
3. Nielsen, M. A. *Neural networks and deep learning* (Determination press, 2015).
4. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT press, 2016).
5. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–44 (2015).
6. Silver, D. et al. Mastering the game of go with deep neural networks and tree search. *Nature* **529**, 484 (2016).
7. Carleo, G. et al. Machine learning and the physical sciences. *Rev. Mod. Phys.* **91**, 045002 (2019).
8. Deng, D.-L., Li, X. & Das Sarma, S. Machine learning topological states. *Phys. Rev. B* **96**. https://doi.org/10.1103/physrevb.96.195145 (2017).
9. Nomura, Y., Darmawan, A. S., Yamaji, Y. & Imada, M. Restricted boltzmann machine learning for solving strongly correlated quantum systems. *Phys. Rev. B* **96**. https://doi.org/10.1103/physrevb.96.205152 (2017).
10. Carleo, G. & Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **355**, 602–606 (2017).
11. Schuld, M & Petruccione, F. *Supervised Learning with Quantum Computers* (Springer, 2018).
12. Das Sarma, S., Deng, D.-L. & Duan, L.-M. Machine learning meets quantum physics. *Phys. Today* **72**, 48–54 (2019).
13. Stoudenmire, E. M. Learning relevant features of data with multi-scale tensor networks. *Quantum Sci. Technol.* **3**, 034003 (2018).
14. Collura, M., Dell'Anna, L., Felser, T. & Montangero, S. On the descriptive power of Neural-Networks as constrained Tensor Networks with exponentially large bond dimension. *SciPost Phys. Core* **4**, 1 (2021).
15. Chen, J., Cheng, S., Xie, H., Wang, L. & Xiang, T. Equivalence of restricted boltzmann machines and tensor network states. *Phys. Rev. B* **97**. https://doi.org/10.1103/physrevb.97.085104 (2018).
16. Levine, Y., Yakira, D., Cohen, N. & Shashua, A. Deep learning and quantum entanglement: Fundamental connections with implications to network design. Preprint at https://arxiv.org/abs/1704.01552 (2017).
17. McCulloch, I. P. From density-matrix renormalization group to matrix product states. *J. Stat. Mech. Theory Exp.* **2007**, P10014–P10014 (2007).
18. Schollwöck, U. The density-matrix renormalization group in the age of matrix product states. *Ann. Phys.* **326**, 96–192 (2011).
19. Singh, S. & Vidal, G. Global symmetries in tensor network states: Symmetric tensors versus minimal bond dimension. *Phys. Rev. B* **88**, 115147 (2013).
20. Dalmonte, M. & Montangero, S. Lattice gauge theory simulations in the quantum information era. *Contemp. Phys.* **57**, 388–412 (2016).
21. Gerster, M., Rizzi, M., Silvi, P., Dalmonte, M. & Montangero, S. Fractional quantum hall effect in the interacting hofstadter model via tensor networks. *Phys. Rev. B* **96**, 195123 (2017).
22. Bañuls, M. C. et al. Simulating lattice gauge theories within quantum technologies. *The European Phy. J. D* **74**, https://doi.org/10.1140/epjd/e2020-100571-8 (2020).
23. Silvi, P. et al. The tensor networks anthology: simulation techniques for many-body quantum lattice systems. *SciPost Phys. Lect. Notes* **8** (2019).
24. Felser, T., Silvi, P., Collura, M. & Montangero, S. Two-dimensional quantum-link lattice quantum electrodynamics at finite density. *Phys. Rev. X* **10**, 041040. https://doi.org/10.1103/PhysRevX.10.041040 (2020).
25. Montangero, S. *Introduction to Tensor Network Methods* (Springer International Publishing, 2018).
26. Bañuls, M. C. & Cichy, K. Review on novel methods for lattice gauge theories. *Rep. Prog. Phys.* **83**, 024401 (2020).
27. Stoudenmire, E. & Schwab, D. J. Supervised learning with tensor networks. In (eds Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I. & Garnett, R.) *Advances in Neural Information Processing Systems 29*, 4799–4807. http://papers.nips.cc/paper/6211-supervised-learning-with-tensor-networks.pdf (Curran Associates, Inc., 2016).
28. Novikov, A., Trofimov, M. & Oseledets, I. Exponential machines. Preprint at https://arxiv.org/abs/1605.03795 (2016).
29. Khrulkov, V., Novikov, A. & Oseledets, I. Expressive power of recurrent neural networks. Preprint at https://arxiv.org/abs/1711.00811 (2017).
30. Liu, D. et al. Machine learning by unitary tensor network of hierarchical tree structure. *N. J. Phys.* **21**, 073059 (2019).
31. Roberts, C. et al. Tensornetwork: A library for physics and machine learning. Preprint at https://arxiv.org/abs/1905.01330 (2019).
32. Glasser, I., Pancotti, N. & Cirac, J. I. From probabilistic graphical models to generalized tensor networks for supervised learning. Preprint at https://arxiv.org/abs/1806.05964 (2018).
33. Aaij, R. et al. LHCb open data website. http://opendata.cern.ch/docs/about-lhcb (2020).
34. Aaij, R. et al. Simulated jet samples for quark flavour identification studies. https://doi.org/10.7483/OPENDATA.LHCB.N75T.TJPE (2020).
35. Tucker, L. R. Some mathematical notes on three-mode factor analysis. *Psychometrika* **31**, 279–311 (1966).
36. Östlund, S. & Rommer, S. Thermodynamic limit of density matrix renormalization. *Phys. Rev. Lett.* **75**, 3537–3540 (1995).
37. Oseledets, I. V. Tensor-train decomposition. *SIAM J. Sci. Comput.* **33**, 2295–2317 (2011).
38. Gerster, M. et al. Unconstrained tree tensor network: an adaptive gauge picture for enhanced performance. *Phys. Rev. B* **90**, 125154 (2014).
39. Hackbusch, W. & Kühn, S. A new scheme for the tensor representation. *J. Fourier Anal. Appl.* **15**, 706–722 (2009).
40. Verstraete, F. & Cirac, J. I. Renormalization algorithms for quantum-many body systems in two and higher dimensions. Preprint at https://arxiv.org/abs/cond-mat/0407066 (2004).
41. Orús, R. A practical introduction to tensor networks: matrix product states and projected entangled pair states. *Ann. Phys.* **349**, 117–158 (2014).
42. Shannon, C. E. A mathematical theory of communication. *Bell Syst. Tech. J.* **27**, 379–423 (1948).
43. Shannon, C. E. A mathematical theory of communication. *Bell Syst. Tech. J.* **27**, 623–656 (1948).
44. Nielsen, M. & Chuang, I. *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).
45. Larkoski, A. J., Moult, I. & Nachman, B. Jet Substructure at the large hadron collider: a review of recent advances in theory and machine learning. *Phys. Rept.* **841**, 1–63 (2020).
46. Butter, A. et al. The machine learning landscape of top taggers. *SciPost Phys.* **7**, 014 (2019).
47. Fraser, K. & Schwartz, M. D. Jet charge and machine learning. *JHEP* **10**, 093 (2018).
48. ATLAS Collaboration. Deep Sets based Neural Networks for Impact Parameter Flavour Tagging in ATLAS. Tech. Rep. ATL-PHYS-PUB-2020-014 (CERN, 2020).
49. ATLAS Collaboration. Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment. Tech. Rep. ATL-PHYS-PUB-2017-003 (CERN, 2017).
50. CMS Collaboration. Performance of b tagging algorithms in proton-proton collisions at 13 TeV with Phase 1 CMS detector. Tech. Rep. CMS-DP-2018-033 (CERN, 2018).
51. Kogler, R. et al. Jet substructure at the large hadron collider: experimental review. *Rev. Mod. Phys.* **91**, 045003 (2019).
52. Aaij, R. et al. Identification of beauty and charm quark jets at LHCb. *JINST* **10**, P06013 (2015).
53. Murphy, C. W. Bottom-Quark Forward-Backward and Charge Asymmetries at Hadron Colliders. *Phys. Rev.* **D92**, 054003 (2015).
54. Alves Jr., A. A. et al. The LHCb detector at the LHC. *JINST* **3**, S08005 (2008).
55. Aaij, R. et al. LHCb detector performance. *Int. J. Mod. Phys.* **A30**, 1530022 (2015).
56. D0 collaboration. Measurements of B_d mixing using opposite-side flavor tagging. *Phys. Rev.* **D74**, 112002 (2006).

57. Giurgiu, Gavril A. B Flavor tagging calibration and search for B_s^0 oscillations in semileptonic decays with the cdf detector at fermilab. United States: N. p., 2005. https://doi.org/10.2172/879144.
58. Aaij, R. et al. First measurement of the charge asymmetry in beauty-quark pair production. *Phys. Rev. Lett.* **113**, 082003 (2014).
59. Miller, J. Torchmps. https://github.com/jemisjoky/torchmps (2019).
60. Milsted, A., Ganahl, M., Leichenauer, S., Hidary, J. & Vidal, G. Tensornetwork on tensorflow: a spin chain application using tree tensor networks. Preprint at https://arxiv.org/abs/1905.01331 (2019).
61. ALEPH collaboration. ALEPH detector performance. *Nucl. Instrum. Meth. A* **360**, 481 (1994).
62. Cacciari, M., Salam, G. P. & Soyez, G. The anti-k_t jet clustering algorithm. *JHEP* **04**, 063 (2008).
63. Clemencic, M. et al. The lhcb simulation application, Gauss: design, evolution and experience. *J. Phys. Conf. Ser.* **331**, 032023 (2011).
64. Sjöstrand, T., Mrenna, S. & Skands, P. A brief introduction to PYTHIA 8.1. *Comput. Phys. Commun.* **178**, 852–867 (2008).
65. Lange, D. J. The EvtGen particle decay simulation package. *Nucl. Instrum. Meth.* **A462**, 152–155 (2001).
66. Agostinelli, S. et al. Geant4: A simulation toolkit. *Nucl. Instrum. Meth.* **A506**, 250 (2003).
67. Allison, J. et al. Geant4 developments and applications. *IEEE Trans. Nucl. Sci.* **53**, 270 (2006).

ACKNOWLEDGEMENTS

We are very grateful to Konstantin Schmitz for valuable comments and discussions on the ML comparison. We thank Miles Stoudenmire for fruitful discussions on the application of the TNs ML code. This work is partially supported by the Italian PRIN 2017 and Fondazione CARIPARO, the Horizon 2020 research and innovation programme under grant agreement No. 817482 (Quantum Flagship—PASQuanS) and the QuantERA projects QTFLAG and QuantHEP. We acknowledge computational resources by CINECA and the Cloud Veneto. The work is partially supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) and the European Social Fund (ESF) as part of the EXIST programme under the project *Tensor Solutions*. We acknowledge the LHCb Collaboration for the valuable help and the Istituto Nazionale di Fisica Nucleare and the Department of Physics and Astronomy of the University of Padova for the support.

AUTHOR CONTRIBUTIONS

Conceptualisation (T.F., D.L. and S.M.); data analysis (D.Z., L.S., T.F., M.T. and A.G.); funding acquisition (D.L., S.M.); investigation (D.Z., L.S., T.F. and S.M.); methodology (T.F., S.M.); tensor network software development (T.F. using private resources); validation (D.Z., L.S., T.F. and M.T.); writing—original draft (T.F., S.M.); writing—review and editing (all authors).

FUNDING

Open Access funding enabled and organized by Projekt DEAL.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41534-021-00443-w.

Correspondence and requests for materials should be addressed to T.F.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.