Computing and Software for Big Science (2021) 5:14
https://doi.org/10.1007/s41781-021-00059-x

ORIGINAL ARTICLE

Efficiency Parameterization with Neural Networks

Francesco Armando Di Bello[1,2] · Jonathan Shlomi[3] · Chiara Badiali[1,2] · Guglielmo Frattari[1,2] · Eilam Gross[3] · Valerio Ippolito[2] · Marumi Kado[1,2,4]

Received: 17 May 2020 / Accepted: 28 April 2021

Abstract
Multidimensional efficiency maps are commonly used in high-energy physics experiments to mitigate the limitations in the generation of large samples of simulated events. Binned efficiency maps are however strongly limited by statistics. We propose a neural network approach to learn ratios of local densities to estimate in an optimal fashion efficiencies as a function of a set of parameters. Graph neural network techniques are used to account for the high dimensional correlations between different physics objects in the event. We show in a specific toy model how this method is applicable to produce accurate multidimensional efficiency maps for heavy-flavor tagging classifiers in HEP experiments, including for processes on which it was not trained.

Keywords Neural networks · Fitting methods · Performance of high energy physics detectors

Introduction

An overarching issue of Large Hadron Collider (LHC) experiments is the necessity of massive numbers of simulated collision events to estimate the rates of expected processes in very restricted regions of phase space. To mitigate this difficulty, a commonly used approach is the *event weighting technique* which replaces selection cuts with event weights. Assuming a set of N events before selection cuts that yield N_f events after the selection, the estimated relative statistical uncertainty on the number of selected events will be $1/\sqrt{N_f}$. If instead of applying selection cuts, a weight corresponding to the selection efficiency w_i is applied to each event indexed by i, then an estimate of the variance will be $\sum w_i^2$, thus yielding a relative statistical uncertainty on the estimated number of selected events of $\sqrt{(\sum_{i\leq N} w_i^2)}/N$. For the method to be effective, the variance of the weights needs to be small compared to the statistical uncertainty on N_f, which is typically the case.

So-far weights have been defined from binned efficiency maps. The difficulty in these methods is the range of applicability of efficiency maps that are limited in the number of dimensions (typically two), and subsequently, fail to capture more subtle effects that appear in specific regions of phase space. To account for these dependencies, a multidimensional mapping is required. This implies large statistical fluctuations in the map itself that defies the original purpose of the method.

A common example of the usage of event weighting techniques is typically given by analyses relying on the identification of jets originating from b-quarks (b-tagging) [1–3]. Applying a weight corresponding to the expected identification efficiency of a jet, i.e. the probability of being identified as a b-jets, instead of a direct selection cut can provide large gains in statistics (especially in cases of percent level efficiencies to be applied on several jets in an event). However, obtaining universally applicable maps requires to account for a large number of parameters. Some of which are typically not known or difficult to take into account with the binned approach.

✉ Francesco Armando Di Bello
Francesco.Armando.DiBello@roma1.infn.it
https://github.com/jshlomi/TruthTagWithNN

Jonathan Shlomi
Jonathan.shlomi@weizmann.ac.il

[1] Dipartimento di Fisica, Sapienza Università di Roma, Rome, Italy

[2] INFN Sezione di Roma, Rome, Italy

[3] Department of Particle Physics, The Weizmann Institute of Science, Rehovot, Israel

[4] Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405 Orsay, France

Published online: 28 May 2021

The goal of the proposed method is to provide higher dimensional parametrizations of efficiencies that can capture non-trivial dependencies while making optimal use of the available statistics and therefore be applicable in any analysis context considered. When achieving this goal the parameterization will be referred to as *universal*. Multidimensional reweighting techniques have been proposed in the context of HEP experiments for BDT and neural networks [4–7]. We propose an approach based on Graph Neural Networks (GNN) [8, 9]. Compared to other non-equivariant deep-learning algorithms, GNN can naturally cope with variable size datasets that have no inherent order while optimally exploiting the pair-wise dependencies between different objects in the event.

The case study used is the *b*-tagging performance in the analysis of Higgs boson decays to *b*-quarks.

The strength of the proposed method relies on its ability to model high dimensional correlations between jets. These jet-by-jet dependencies are not given explicitly as input variables to the neural network, but rather they are inferred from single-jet properties during the training of the network. In case multiple jets in the event are *b*-tagged, the jet-efficiencies provided by the NN can be combined to derive an unbiased estimator of the event tagging efficiency. A toy model is built to probe the capability of the Machine Learning (ML) approach to provide a robust parameterization of the *b*-tagging efficiency.

The paper is organized as follows. Section "Event Weighting Technique" introduces the event weighting technique and describes the main challenges and goals of the method. Section "Simulated samples" describes the simulation technique used to generate the toy data-set. Section "Efficiency Map Techniques" describes a map-based technique that is commonly used to estimate the event weight based on a parameterization of the *b*-tagging classifier performance. Section "Truth Tagging with Neural Networks" describes the GNN model, whose results are compared to the ones of the map-based technique in Section "Results". In Section "Discussion" some considerations about the usage of the proposed methodology in real experiments are presented. Conclusions are drawn in Section "Conclusions".

Event Weighting Technique

In high energy physics experiments (HEP), estimating a background rate or a signal efficiency from a selection cut is most accurately achieved by a full simulation of the event. However, the precision of such an estimate can be heavily affected by the limitation in the number of events that can be simulated in a given region of phase space. If instead of selecting events based on a classification cut, a weight corresponding to the classifier efficiency is applied, significant improvements in sensitivity can be gained. This procedure is also known as *Tag-Rate-Function (TRF) method* or *Truth Tagging (TT)* [10–12].

Selections can be interpreted as a classification depending on a vector of input variables $\mathbf{x}$. The classifier can be represented by a function $f(\mathbf{x})$ and the classification by a simple selection cut on the classifier above a given threshold T_f. The classifier can represent simple cuts or a multivariate method. Typically the variables $\mathbf{x}$ depend on several underlying variables which will be denoted by $\boldsymbol{\theta}$.

In the case of heavy-flavor tagging, $\boldsymbol{\theta}$ is typically defined as the jet transverse momentum p_T and pseudo-rapidity η [10], while $\mathbf{x}$ includes the reconstruction of secondary vertices and a combination of track impact parameter information estimated from the properties of a set of reconstructed charged-particle tracks. This information is then combined to produce a multivariate jet-based classifier $f(\mathbf{x})$. Figure 1 schematically shows the usage of the efficiency for event weighting to reduce statistical uncertainties on simulated Monte-Carlo (MC) samples.

A parametrized classifier efficiency can be defined as:

$$\epsilon_\mathrm{jet}(\boldsymbol{\theta}) = \frac{N(f(\mathbf{x}) > \mathrm{T_f}|\boldsymbol{\theta})}{N(\boldsymbol{\theta})} \tag{1}$$

where T_f is the operating working point threshold of the classifier; the numerator, the selected number of jets of a given flavor at this working point; and the denominator represents the total number of jets of the same flavor.

To achieve a parametrization of the efficiency, applicable to a large number of analyses, a set of relevant variables $\boldsymbol{\theta}$ must be defined such that the conditional probability of the classifier inputs, x, at a given value of $\boldsymbol{\theta}$, $p(\mathbf{x}|\boldsymbol{\theta})$, will be identical between samples or different regions of phase space, as illustrated in Fig. 2.

This motivates the *efficiency maps* approach, where an attempt is made to parametrize ϵ_jet binned in $\boldsymbol{\theta}$. Efficiency maps are a commonly used tool in collider experiments. However, taking into account the full dependencies of the classifier efficiency is often impractical using efficiency maps. The reason being that a small enough set of variables that fully capture these dependencies might not be available.

In the case of *b*-tagging it was found that while p_T and η are indeed the most dominant variables in determining ϵ_jet, there are other variables that affect the efficiency and could be considered had we known them, e.g. the angular separation and flavor of the adjacent jets [2, 13].

We propose a different approach to estimate ϵ_jet based on a neural network built using a GNN. The neural network takes as input a set of jet-variables $\boldsymbol{\theta}_{j_e}$ for each jet j in the event e. The input variables are the jet-(p_T, η, ϕ, flavor) and the neural network model is trained to predict the per-jet efficiency ϵ_jet. Since the true ϵ_jet is conditional

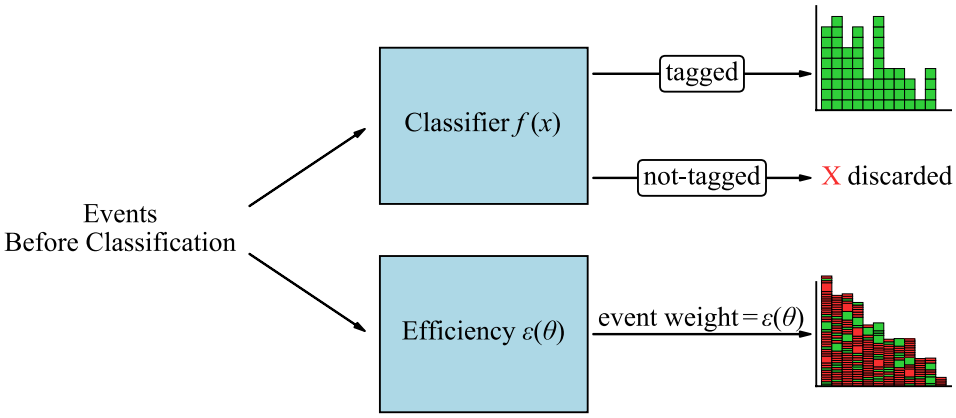


Fig. 1 Usage of event weighting to reduce MC statistical uncertainties of some observable distribution. The plot on the top shows a classifier $f(\mathbf{x})$ used to select events. The events which pass the classification requirement are represented in green while the rejected events are shown in red. The bottom panel shows the event weighting where the classifier efficiency $\epsilon(\boldsymbol{\theta})$ is used to weight the events rather than rejecting them. $\mathbf{x}$ are the variables used by the classifier. For b-tagging, $\mathbf{x}$ includes variables such as the secondary vertex information while $\boldsymbol{\theta}$ is the set of relevant variables used for the parametrization of the efficiency, such as the jet p_{T} and η

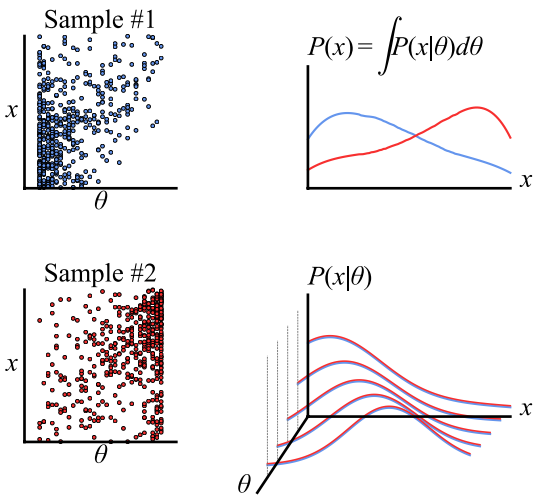


Fig. 2 Illustration of a universal parametrization of the classifier efficiency. The joint distribution of $(\mathbf{x}, \boldsymbol{\theta})$ is generally different between two samples. The top right plot shows the overall probability distribution of the input variables of the classifier, $P(\boldsymbol{x})$, for two different samples. Different $P(\mathbf{x})$ distributions lead to different overall efficiencies between the two samples. The bottom right plot shows the conditional probability distributions, $P(\mathbf{x}|\boldsymbol{\theta})$, between the two samples. The set of relevant variables $\boldsymbol{\theta}$ is defined to provide a $P(\mathbf{x}|\boldsymbol{\theta})$ which is sample independent. Under this condition, the parametrized classifier efficiency $\epsilon(\boldsymbol{\theta})$ is expected to be universal

on the jet environment, its proximity to other jets, the neural network should learn to model that dependence, even if not explicitely given as input variable.

Simulated Samples

The samples employed in this study consist of toy pp collision events with multiple jets generated with generic kinematic and flavor properties. We assume a cylindrical coordinate system where particle beams collide on the z axis, xy is denoted as the transverse plane, ϕ is the azimuthal angle, θ the polar angle, and pseudo-rapidity η is defined as $\eta = -\log\tan(\theta/2)$.

The generated events are sampled using an exponential function to fix the number of jets in the event and Gaussians or polynomial distributions to sample the jet kinematics variables and the angular distance between two jets $\Delta R(i,j) = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2}$. More details about the event generation can be found in "Appendix A".

Three separate samples of four-momenta representing b-, c- and light-jets are generated. The b-tagging efficiency is modeled using ad-hoc parameterizations using a multivariate Gaussian distribution depending on p_{T} and η which is modified by a multiplicative correction factor depending on the angular distance $\Delta R(i,j)$ of other jets in the event as well as their flavor. This efficiency is chosen to mimic the b -tagging performance of ATLAS and CMS [1, 14] and it is expressed as:

$$\epsilon_{\mathrm{jet}_i} = \epsilon_{f_i}(p_{\mathrm{T}}, \eta) \cdot \prod_j \hat{\epsilon}_{ij}\big(\Delta R(i,j), f_j\big), \tag{2}$$

where $\epsilon_{f_i}(p_{\mathrm{T}}, \eta)$ is the two-dimensional parameterisation of the efficiency to tag a jet of a given flavor f_i, and $\hat{\epsilon}_{ij}\big(\Delta R(i,j), f_j\big)$ is the one-dimensional correction factor which accounts for the effect of any close-by jet j of flavor f_j in the

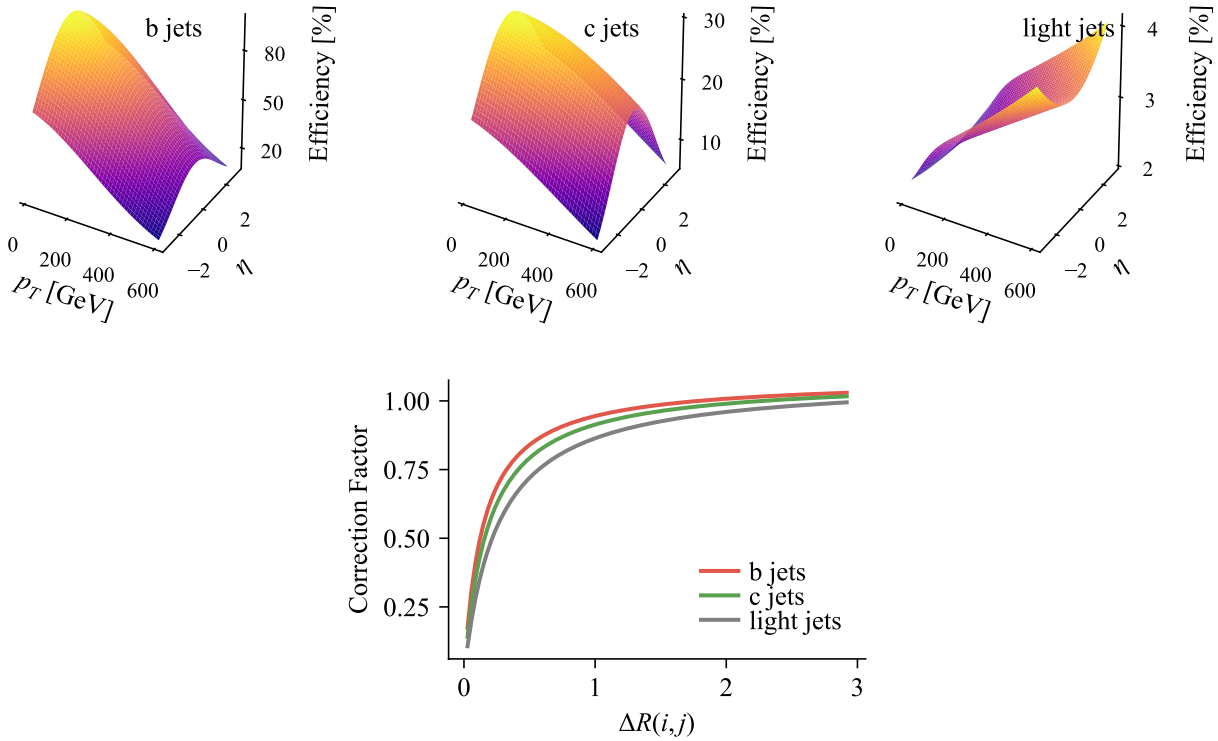


Fig. 3 The parameterized efficiencies used to emulate the performance of the flavor tagging algorithms. The efficiencies for each flavor as a function of p_{T} and η, $\epsilon_{f_i}(p_{\mathrm{T}}, \eta)$ in the top three panels. The multiplicative correction factor $\hat{\epsilon}_{ij}\left(\Delta R(i,j), f_j\right)$ which accounts for the proximity ($\Delta R(i,j)$) and flavor of the close-by-jet f_j is shown at the bottom of the figure

event. The efficiencies $\epsilon_{f_i}(p_{\mathrm{T}}, \eta)$ and the correction factors $\hat{\epsilon}_{ij}\left(\Delta R(i,j), f_j\right)$ are shown in Fig. 3.

The true b-tagging efficiency of each individual jet in the event is computed using Eq. 2. This efficiency value $\epsilon_{\mathrm{jet}_i}$ is used to emulate b-tagging by assigning a boolean value to each jet `istag` which is set to 1 based on a random score s_i sampled from a uniform distribution. Namely, if $s_i < \epsilon_{\mathrm{jet}_i}$ the i-th jet in the event is considered to be b-tagged (`istag`=1). In many physics analyses, multiple jets in the event are required to pass b-tagging selections, hence the efficiencies of the single jet need to be combined to form a per-event efficiency. In this toy analysis the event selection is based on the two jets with highest p_{T} in the event ("leading jets", labeled as 1 and 2), and it is defined depending on the number of b-tagged jets, n_{tag}:

$$\epsilon_{\mathrm{event}} = \begin{cases} (1-\epsilon_1)(1-\epsilon_2) & \text{if } n_{\mathrm{tag}} = 0, \\ \epsilon_1(1-\epsilon_2) + (1-\epsilon_1)\epsilon_2 & \text{if } n_{\mathrm{tag}} = 1, \\ \epsilon_1\epsilon_2 & \text{if } n_{\mathrm{tag}} = 2. \end{cases} \quad (3)$$

Efficiency Map Techniques

The estimation of $\epsilon_{\mathrm{event}}$ in the case of b-tagging in real experiments is commonly based on the binned two-dimensional efficiency maps in the jet p_{T}-η plane [12, 15], $\tilde{\epsilon}$, derived from MC simulation separately for b-jets, c-jets and light-jets, which are used to approximate the per-jet b-tagging efficiency of Eq. 2 as:

$$\epsilon_{\mathrm{jet}} \approx \tilde{\epsilon}_i = \tilde{\epsilon}_{f_i}(p_{\mathrm{T}}, \eta). \quad (4)$$

The choice of the variables used to parameterize $\tilde{\epsilon}$ is motivated by the expected dependency of the b-tagging performance. For example, as the transverse momentum of a b-jet increases, the dilation of its lifetime in the laboratory frame results in secondary decay vertices which are reconstructed further from the interaction point of the primary collision. The reconstruction efficiency of secondary vertices is not constant as a function of their distance to the primary vertex and this affects the response of the b-tagging classifier. Similarly, the typical configuration of multi-purpose detectors produces a dependency of track reconstruction performance on detector geometry, which in turn propagates into a dependency of the b-tagging performance on η.

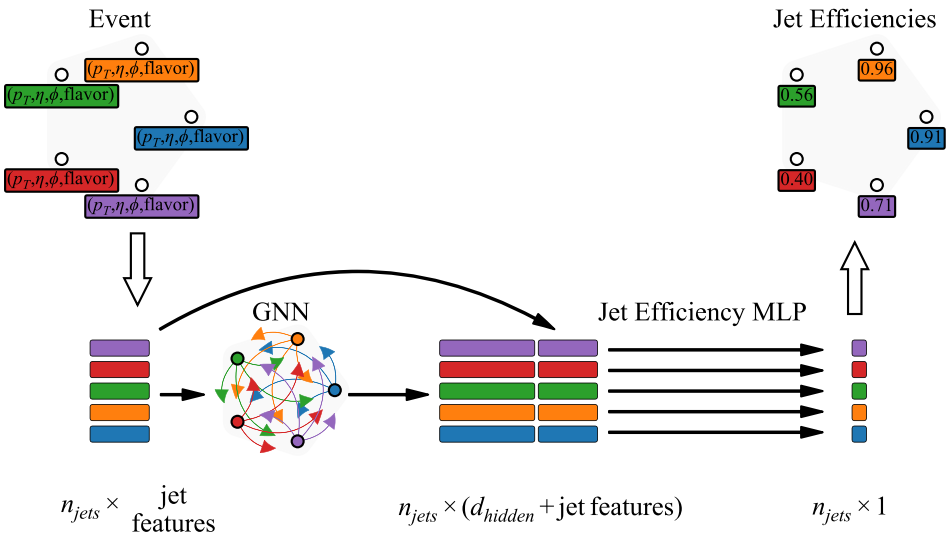


Fig. 4 Schematic representation of the neural network structure

From the per-jet efficiency maps $\tilde{\epsilon}$ the event weight ϵ_{event} is computed factorizing the contribution from the various jets, similarly to what is shown in Eq. 3.

The main limitation of this map-based approach is the assumption that correlations between jets can be neglected and that the efficiency of b-tagging a single jet only depends on its p_{T} and η. The dependency of efficiency on residual observables is marginalized out when deriving $\tilde{\epsilon}$ from MC samples, introducing a bias that is particularly significant for final states with large jet multiplicities or events where close-by or overlapping jets are reconstructed from the decay of boosted resonances. A dedicated $\Delta R(i,j)$ reweighing was derived and used to correct for this effect in previous $H \to b\bar{b}$ and $H \to c\bar{c}$ analyses [2, 13]. Given the uncertain nature of this correction and the limited statistics of the sample used to derive it, a large systematic uncertainty equal to half of the correction was assigned to the relevant MC templates. The overall uncertainty related to the statistics of the MC templates constitutes a contribution up to around 20% to the total background uncertainty [3, 16].

Additional limitations come from the binning of the two-dimensional maps. To reduce discontinuities, smoothing techniques need to be employed. However, these techniques often require a non-trivial interplay between the bin sizes and the parameters of the smoothing model which makes their implementation unpractical compared to a single unbinned neural network training. Finally, the NN technique provides a simultaneous estimate of the efficiency for each jet-flavor in contrast to the map-based approach which requires a dedicated parametrization for each of the flavors independently.

Truth Tagging with Neural Networks

Taking into account the full dependency of the jet-tagging probability on all event observables would be unpractical with a map-based approach. ML techniques, on the other hand, provide the possibility to scale the problem to higher dimensionality and, therefore, to more challenging physics topologies.

In principle, a standard feedforward neural network could be used to perform the task. However, these models are not able to optimally cope with inputs of variable sizes and thus the overall correlations between jets in the event cannot be easily exploited during the training. The technique we propose uses a GNN to capture efficiently these correlations. A GNN also offers a more natural representation of the data by exploiting pair-wise relationships between the jets. In our toy experiment, each jet is represented by a set of variables corresponding to (p_{T}, η, ϕ, flavor). The neural network takes as input these variables for each jet in the event e, $\boldsymbol{\theta}_e = ((p_{\text{T}1}, \eta_1, \phi_1, \text{flavor}_1), \ldots, (p_{\text{T}n_{\text{jets}}}, \eta_{n_{\text{jets}}}, \phi_{n_{\text{jets}}}, \text{flavor}_{n_{\text{jets}}}))$ and learns to approximate the efficiency given in Eq. 2 for each of these jets. Note that the inputs to the neural network do not include ΔR between neighboring jets, which is the variable that determines the correction applied in Eq. 2 but rather this dependency is inferred directly during the training.

Model Architecture The model, referred to as NN in the following, consists of two components: a GNN [8] and a *jet efficiency* network. The flow of information between the different parts is illustrated in Fig. 4.

The GNN takes as input the $n_{\text{jets}} \times 4$ matrix of jet features, and outputs $n_{\text{jets}} \times d_{\text{hidden}}$ matrix of jet hidden

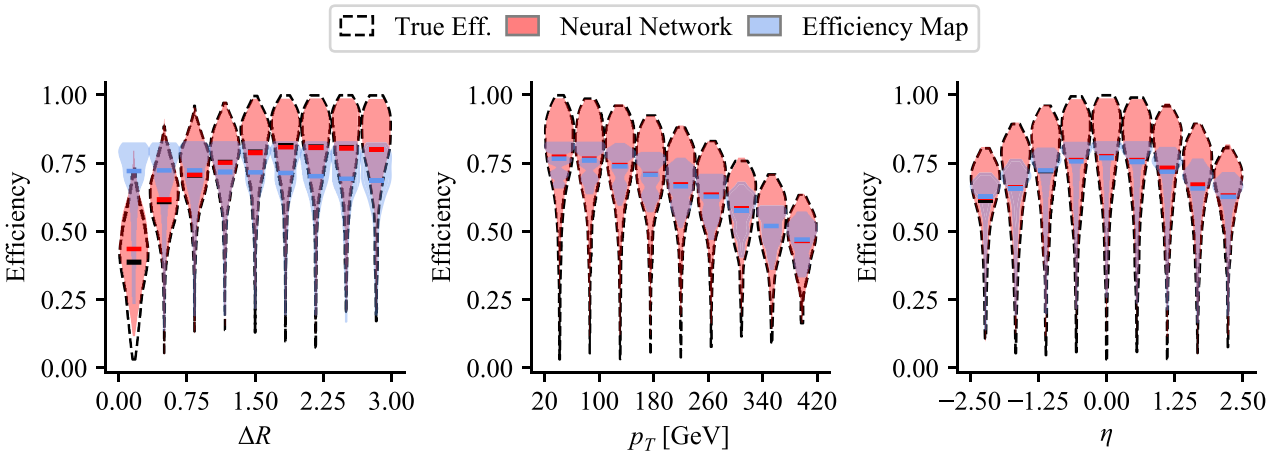


Fig. 5 Violin plot illustrating the distributions of the true and predicted efficiency as a function of different kinematic variables for *b*-jets

representations.[1] The*hidden representation* for each jet is based on the information of the other jets in the event. The jet efficiency network then operates on each jet individually. It takes as an input the jet variables and the jet hidden representation and it returns as an output the predicted ϵ_{jet} for every jet in the event. More details about the model architecture can be found in "Appendix B".

Training Procedure The network is trained to predict the $n_{\text{jets}} \times 1$ vector of efficiencies. The loss function used for training is the weighted binary cross-entropy (BCE), which for a single event it can be written as:

$$\begin{aligned}\text{BCE}_e = \frac{1}{N_{\text{jets}}} \sum_{i}^{N_{\text{jets}}} \big[-(\texttt{istag}_i) \log(\epsilon_{NN}(\boldsymbol{\Theta}_e)_i)\big] \\ - \big[\mu(1 - \texttt{istag}_i) \log(1 - \epsilon_{NN}(\boldsymbol{\Theta}_e)_i)\big],\end{aligned} \tag{5}$$

where the sum runs over the sets of jets, N_{jets}, in the event, e, which pass (istag=1) and do not pass (istag=0) b-tagging and $\epsilon_{NN}(\boldsymbol{\Theta}_e)_i$ is the i-th component of the output of the NN, a vector of variable size representing the predicted efficiency of tagging each jet in an event. The loss function being minimized is the sum of BCE_e for all the events in a batch. The factor μ controls the weight of the non-tagged events and can be used to balance the number of tagged and non-tagged jets to facilitate the training. This approach could be useful for light-jets where the number of non-tagged jets is $\mathcal{O}(100)$ larger than the tagged ones. Even if this factor was found to be helpful in tests conducted with feedforward networks, for GNNs it was found to have a negligible impact on the final results. Therefore, μ=1 is assumed in the following discussions.

Using a well-known result, the neural network trained using BCE as loss function converges to the following ratio [17]:

$$\epsilon_{NN}(\boldsymbol{\Theta}_e)_i \approx \frac{p_{\text{tag}}(\boldsymbol{\Theta}_e)_i}{p_{\text{tag}}(\boldsymbol{\Theta}_e)_i + p_{\text{non-tag}}(\boldsymbol{\Theta}_e)_i} \approx \epsilon_{\text{jet}_i}, \tag{6}$$

$\epsilon_{NN}(\boldsymbol{\Theta}_e)_i$ is the output of the network for the i-th jet in the event e which approximate the true efficiency ϵ_{jet_i} given in Eq. 2.

It is worth noticing that the NN computes directly the efficiency $\epsilon_{NN}(\boldsymbol{\Theta}_e)_i$ without regressing $p_{\text{tag}}(\boldsymbol{\Theta}_e)_i$ and $p_{\text{non-tag}}(\boldsymbol{\Theta}_e)_i$ independently.

Additional details on the training procedure can be found in "Appendix C".

Results

In this section, the result of approximating ϵ_{jet} and ϵ_{event} using the jet b-tagging efficiencies calculated from the NN are presented and compared to the results obtained with the map-based technique discussed in Section 4. Three main aspects are discussed: the modeling of single-jet distributions after jet weighting, the capability of the NN technique to provide an unbiased estimation of ϵ_{event}, and the independence of the GNN performance on the choice of the sample used for training.

The true and predicted efficiencies are shown as a function of the set of relevant parameter $\boldsymbol{\theta}$ in Fig. 5 for b-jets.[2] The relative residuals between ϵ_{true} and $\epsilon_{\text{predicted}}$ for all jets

[1] d_{hidden}, a hyperparameter of the model, is the size of this representation and it is fixed to 256.

[2] Similar results were found for c-jets and light-jets and are thus not shown for simplicity.

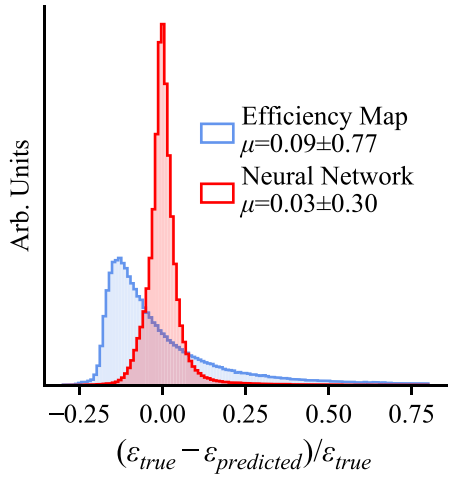


Fig. 6 Relative residuals distributions as predicted by the NN and the map-based approach for each individual jet in the event. The mean and RMS of the distributions are outlined in the plot

in the dataset is shown in Fig. 6. For both Figures, ϵ_{true} is computed during the generation of the data-set following Eq. 2. While, as expected, the map-based approach is unable to provide good modeling of the $\Delta R(i,j)$ distribution, the NN predictions are in good agreement with the distributions obtained when jets pass the tagging selection (direct tagging) and with true efficiency weights. These results give us confidence about the ability of the GNN to build an internal representation capable of capturing additional jet-to-jet information relevant to estimating the true tagging efficiency

Results of the reweighing procedure are further studied when both the leading and sub-leading jets are classified as b -jets, and compared to those from direct tagging. In this case, the event weight is simply computed as the product of the efficiencies of b-tagging each of the two jets, $\epsilon_{\text{event}} = \epsilon_1 \cdot \epsilon_2$. It is therefore important to study the modeling of distributions that capture correlations among individual jet observables, once event weights are applied.

The invariant mass distribution computed from the leading and subleading jets in each event is shown in Fig. 7. The figures are further sub-divided based on the true flavors of the two jets. The uncertainty on the efficiency prediction are estimated using a bootstrap procedure. The source of this uncertainty originates from the limited size of the training data-set and the inherent randomnesses of the training process. A more detailed discussion on the uncertainty bands can be found in Appendix 10. Similarly to the single-jet case, the NN predictions show good agreement compared to the true efficiency while the map-based approach is unable to properly capture the effect of close-by jets on b-tagging. It can also be noted that the reweighing procedure based on NN predictions improves the statistical uncertainly compared to the direct tagging.

Finally, the generality of the method is probed by using the same network to reweight events from a separate sample with different jet p_{T}, η and $\Delta R(i,j)$ distributions compared to the training sample. More details about this sample can be found in "Appendix A". Figure 8 shows the results for the angular separation between the two decay products as well as for the reconstructed invariant mass of the generated boson. An overall good agreement is found between the NN results and direct tagging, similarly to the previous cases. This gives confidence about the universality of the proposed approach: as long as the phase space is sampled adequately during training, the efficiency estimated using the neural network is expected to be independent on the chosen sample.

Discussion

In this section we summarize some of the main considerations aimed at generalizing the proposed approach for use cases beyond the toy model presented in this paper.

The size of $\boldsymbol{\theta}$: In the toy data-set we used a relatively small number of variables that control the efficiency the network was required only to infer the "hidden" variable $\Delta R(i,j)$. In more realistic applications, $\boldsymbol{\theta}$ may include more variables and the function $\epsilon(\theta)$ may be more complicated. To cope with this, the inputs features $\boldsymbol{\theta}$ may need to be extended with additional variables. The number of the model learnable parameters also needs to be large enough so that the model is sufficiently expressive to describe $\epsilon(\theta)$. Any variables potentially correlated with the tagging decision could be used to ensure that all correlations are captured. Neural networks are a particularly suitable tool to perform this task due to their flexibility to cope with higher dimensions.

The functional form of $\epsilon(\theta)$: We assumed a relatively simple efficiency in Eq. 2. In principle, the neural network can learn any function, no matter how complex the functional form is, as shown in Ref. [18]. The method can be used in scenarios where the form of $\epsilon(\boldsymbol{\theta})$ may present more complex dependencies between the efficiency and the relevant variables $\boldsymbol{\theta}$.

Systematic uncertainties: In the applications of the simple efficiency maps, the insufficient capture of the existing underlying correlations requires the introduction of systematic uncertainty. This method is aimed at avoiding this systematic error, it will, however, require thorough checks to ensure that its estimates are accurate.

Generalization of the method: In the proposed approach we have focused our studies to approximate efficiency, i.e. density ratios between two complementary classes. The method can also be generalized to

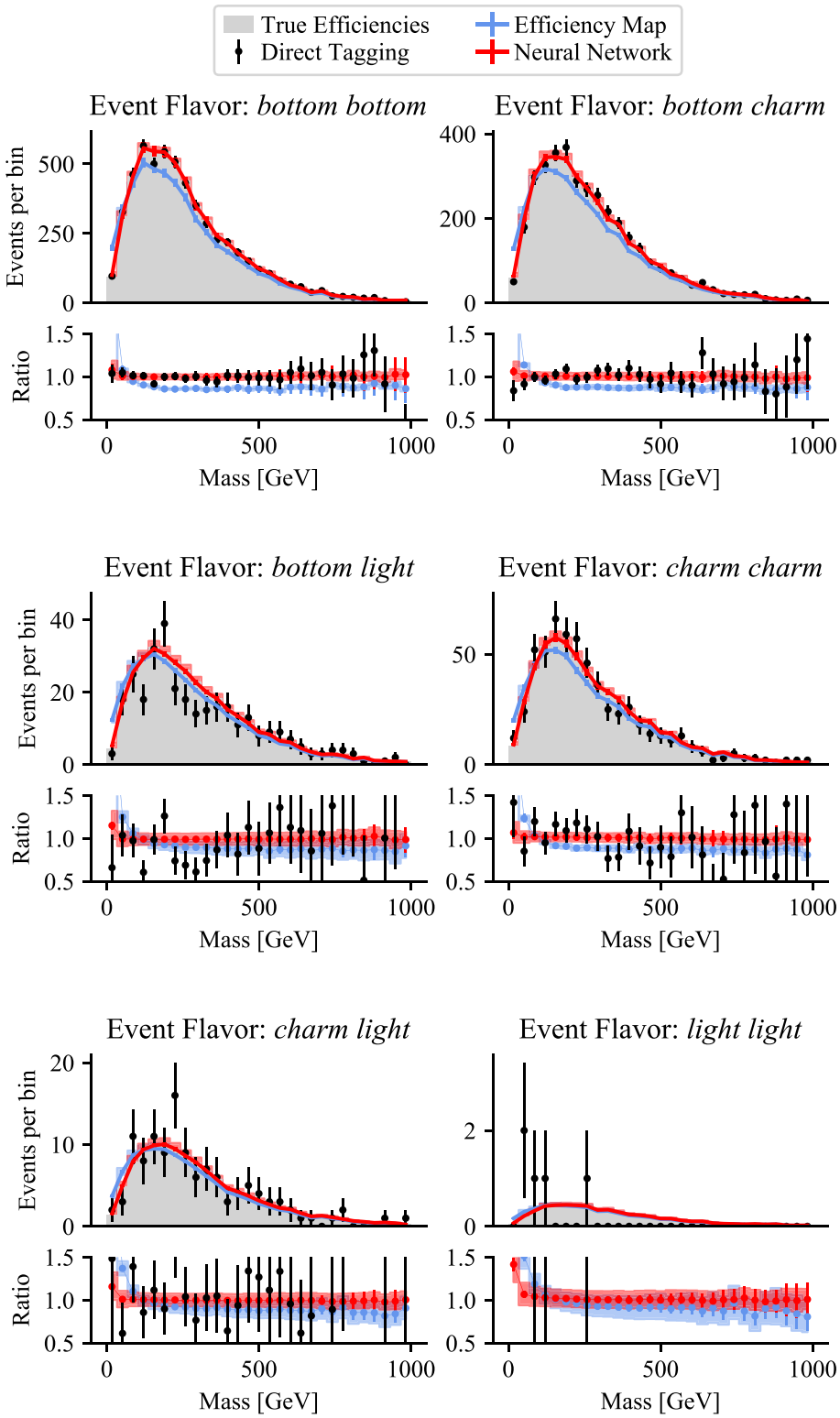


Fig. 7 istribution of the invariant mass of the two leading jets, when the events are weighted by the product of true efficiencies, as calculated in Eq. 2 (grey). Also shown is the distribution for events where both jets are b -tagged (direct tagging, black), or when the events are weighted using the estimated efficiency $\tilde{\epsilon}$ from the map-based approach (blue) or using the NN output (red). The lower pad shows the ratio between all distributions and the one obtained with true weights. Events are split into categories based on the true flavor of the two leading jets

approximate ratios between two separate classes.[3] A multidimensional ratio between two classes could be used in a variety of different applications, such as to derive multi-dimensional scale factors from data to correct the tagging efficiency in Monte Carlo simulation.

[3] In such cases, the loss function needs to be changed to cope with non-complementary classes as discussed in Ref. [17]

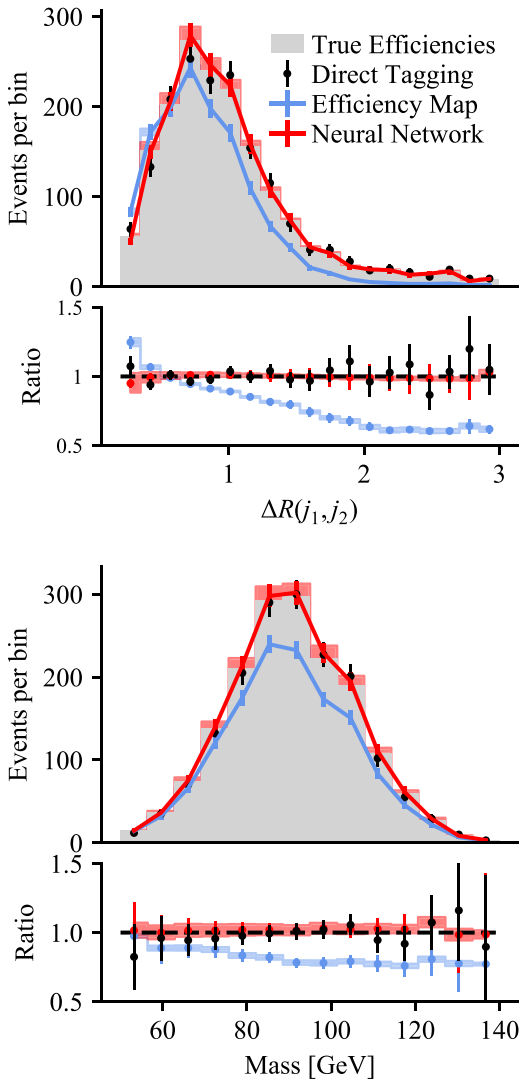


Fig. 8 Distribution of the $\Delta R(i,j)$ (top) and invariant mass (bottom) of the leading-subleading jet system, obtained for events where these jets are classified as b-tagged (black), compared to the same distributions obtained when these jets are instead weighted with their probability of passing b-tagging, calculated using the true weight ϵ from Eq. 2 (grey), using the efficiency $\tilde{\epsilon}$ from the map-based approach (blue) or using the NN output (red). The lower pad shows the ratio between the two latter distributions and the one obtained with true weights

Conclusions

The parametrization of classifier efficiencies can play an important role to mitigate the limitations in the number of simulated events at LHC experiments. To be effective, parametrized classifier efficiencies need to be accurate in any context and therefore need to capture the dependencies on event properties that are used in analyses and which entail variations of efficiencies. A new technique that optimally exploits these dependencies is proposed. This technique is based on graph neural networks that provide an estimate of ratios between multidimensional local densities. We use the case of the identification of heavy-flavor jets as a topical example building a toy model based on ad-hoc parameterizations of the classifier efficiency inspired by the observed dependencies of b-tagging performance in the ATLAS and CMS experiments. A Graph Neural Network is used to exploit correlations between jets in the event to provide a less biased parametrization compared to the canonical map-based method.

A toy example is used to probe the performance of the method, which takes as an input the true flavors and momenta of reconstructed jets, and returns the b-tagging efficiency of each. These efficiencies are used to build the per-event weights in a sample of simulated events with multiple b-tagged jets. We use the estimated efficiency for the event reweighing technique which is used to reduce the statistical fluctuations of Monte Carlo samples after classification.

Results show good compatibility between per-jet and per-event kinematic distributions obtained with the proposed approach and the distributions expected from the direct application of b-tagging. We also show that the proposed technique can generalize to samples with input distributions differing significantly compared to the training sample while covering the same phase space.

Appendix A: Sample Generation Details

This section describes the event generation of the toy model employed throughout this paper. The number of jets in the event is sampled using the following function: $e^{-\frac{N_{\text{jets}}}{4}}$. At least two jets with $p_{\text{T}} > 20$GeV and $|\eta| < 2.0$ are generated. For each jet in the event, the jet transverse momentum is sampled from a gaussian distribution centered at 20 GeV with a width of 200 GeV, the sampling range is chosen to be [20, 600] GeV. The pseudo-rapidity of the leading jet in the event is sampled from a gaussian distribution centered at 0 with a width of 0.5 while the the azimuthal angle is sampled from a uniform distribution bounded in $[0, 2\pi]$. The angular variables of the other jets in the event are chosen by sampling from the square root of the angular distance, $\sqrt{\Delta R(i,j)}$, with $\Delta R(i,j) = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2}$ computed w.r.t. the leading jet. For a given value of $\Delta R(i,j)$, the jet angles are sampled from a uniform distribution in the $\eta - \phi$ plane at the fixed $\Delta R(i,j)$ value. The masses of the single jets are fixed at 2 GeV. These parameters ensure an invariant mass distribution similar to the

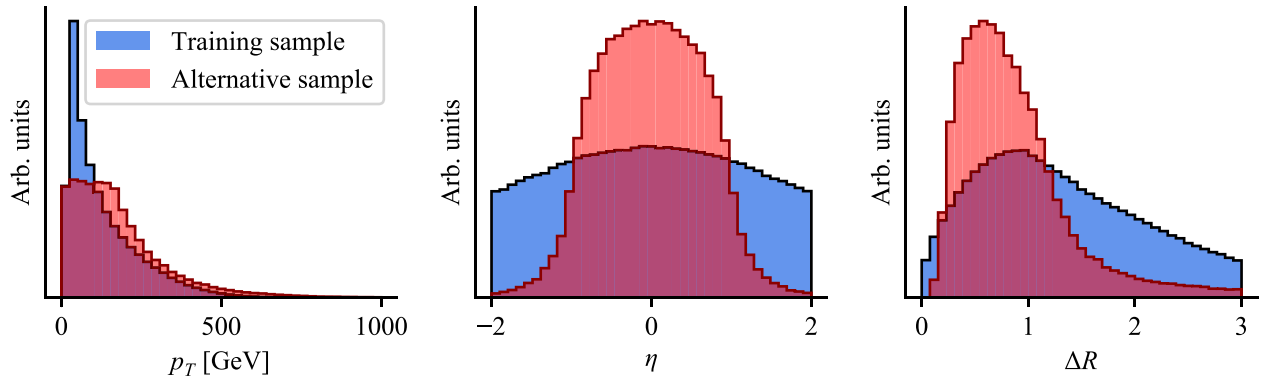
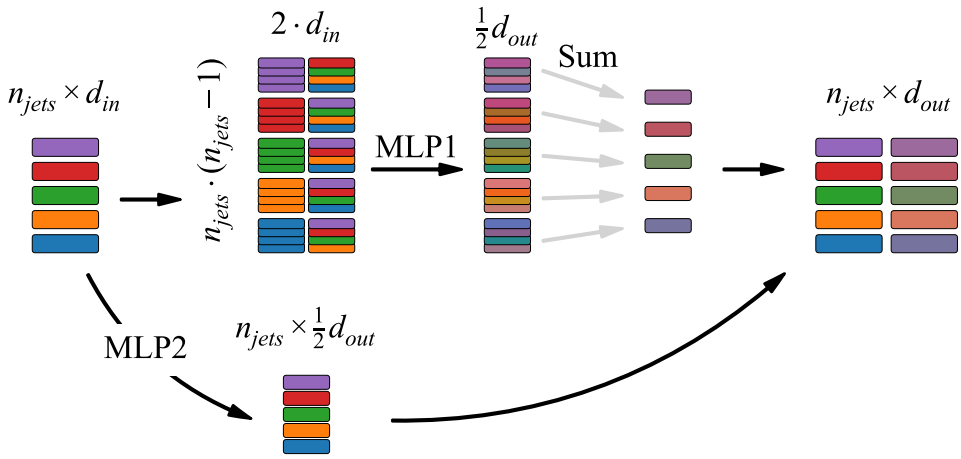


Fig. 9 Normalized distribution of the jet-p_{T}, jet-η and $\Delta R(i,j)$ for the training and the alternative sample

Fig. 10 GN block architecture



one obtained in W/Z+jets events, as mentioned in the main body of the paper.

A second sample, referred to as *alternative* sample, was generated with different kinematic distributions compared to the training sample. For this purpose, events were simulated in which a boosted scalar particle decays in exactly two jets per event, where the p_{T} of the decaying particle is generated from an exponentially decaying distribution, and its mass is generated from a Gaussian distribution peaked at 90 GeV. The boson decays with a rate of 33% to light-, c- or b-jets. A comparison of the kinematic variables between the training sample and the boson sample is shown in Fig. 9. It is worth noticing that the overall distributions are different between the training and the alternative samples but there is overlap between the jets phase space.

Appendix B: Model Architecture

GNN Architecture. The GNN is built from a stack of "GN blocks" as described in [8]. The GN block is shown schematically Fig. 10.

Each GN block takes in a matrix with shape $n_{\mathrm{jets}} \times d_{\mathrm{in}}$, where d_{in} is the size of the vector representing each jet. The output is a $n_{\mathrm{jets}} \times d_{\mathrm{out}}$ matrix where each jet representation has been updated based on the representation of the other jets in the event.

Internally, the output representation is formed from a concatenation of two components.

The first component is a jet representation created by collecting information from other jets—first the input is rearranged to form all the ordered pairs of jets ($n \cdot (n-1)$ for n jets in an event) by concatenating the input features of the two jets. A MLP is then applied to the jet-pairs (MLP1 in Fig. 10). The output is summed for groups of jet-pairs who share the same "first jet" (note the pairs are ordered), resulting in a representation of size $\frac{1}{2} d_{\mathrm{out}}$ for each of the n_{jets}. This representation is passed through another MLP (MLP3, not shown in Fig. 10), which maintains the same output size.

The second component is formed by an MLP (MLP2 in .reffig:gnnspsarch) applied to each jet, creating a representation of size $\frac{1}{2} d_{\mathrm{out}}$.

The resulting $n_{\text{jets}} \times d_{\text{out}}$ representation is normalized, such that each jet representation has Euclidean norm of 1.

The GN blocks are applied to the input data sequentially. After the application of each GN block, the initial input of size $n_{\text{jets}} \times$ jet features is concatenated with the output (a "skip connection"). This is done to optimally exploit the known dependencies of the *b*-tagging efficiency with the jet transverse momentum and pseudo-rapidity. While the output of the GN block is essential to encode the jet-by-jet as well as single-jet dependencies, the skip connection is only used to facilitate the convergence of the training procedure.

Appendix C: Model Details

GNN layer sizes (d_{in},d_{out}):

- (4, 256)
- 3 layers of $(256 + 4, 256)$

GN block MLP1: ReLU activation between each layer, and a final Tanh activation on the final layer.

- $(2 \cdot d_{in}, \frac{1}{2} \cdot (2 \cdot d_{in} + \frac{1}{2} \cdot d_{out}))$
- $(\frac{1}{2} \cdot (2 \cdot d_{\text{in}} + \frac{1}{2} \cdot d_{\text{out}}), \frac{1}{2} \cdot (2 \cdot d_{\text{in}} + \frac{1}{2} \cdot d_{\text{out}}))$
- $(\frac{1}{2} \cdot (2 \cdot d_{\text{in}} + \frac{1}{2} \cdot d_{\text{out}}), \frac{1}{2} d_{\text{out}})$

GN block MLP2: ReLU activation between each layer, and a final Tanh activation on the final layer.

- $(d_{in}, \frac{1}{2} \cdot (d_{\text{in}} + \frac{1}{2} \cdot d_{\text{out}}))$
- $(\frac{1}{2} \cdot (d_{\text{in}} + \frac{1}{2} \cdot d_{\text{out}}), \frac{1}{2} d_{\text{out}})$

GN block MLP3: ReLU activation between each layer, and a final Tanh activation on the final layer.

- $(\frac{1}{2} d_{\text{out}}, \frac{1}{2} d_{\text{out}})$
- $(\frac{1}{2} d_{\text{out}}, \frac{1}{2} d_{\text{out}})$

Jet Efficiency MLP layers (d_{in},d_{out}):

- $(256 + 4, 256)$
- (256, 128)
- (128, 50)
- (50, 1)

Training Procedure and Uncertainty Estimation

The uncertainty of the neural network estimate comes from two sources. The first, referred to as *training* uncertainty, is related to the network training procedure and the fact that it will not always lead to the same final network parameters depending on the choice of initial training parameters. This is due to the inherent randomness of the training processes with stochastic gradient descent. The second is the statistical uncertainty related to the finite size of the training sample.

The training is done on 1.5 million events, for 40 epochs, with a batch size of 5000 events. 500 K events are used as a validation set and 100 k events are used for evaluation. After each epoch of training, the loss is evaluated over the validation set and the model with the smallest validation set loss over the 40 training epochs is saved. The batch size is particularly important for this task as a significant amount of tagged and non-tagged jets needs to be present to reduce statistical fluctuations during training.

The training uncertainty can be reduced by using ensembles of networks, where for one given training dataset, the training is repeated multiple times, and the ensemble of trained models is considered as our final estimator—using the mean of the network predictions as the efficiency estimate. To estimate the training uncertainty of either a single network or the ensemble, we repeat the training 100 times, training either 100 single networks or 500 networks (100 ensembles of 5 networks).

The statistical uncertainty can be estimated by using a bootstrap procedure. Toy-data of size 1.5 M events are sampled with replacement from the original dataset. Similarly to what is done to estimate the training uncertainty, for each toy dataset 5 different networks are trained. The evaluation is run over this ensemble of networks and the standard deviation is used as an estimate of the statistical uncertainty. The uncertainty estimated with this method is expected to encompass both the uncertainty from the finite size sample and the training uncertainty. Figure 11 shows the distributions of relative uncertainties in each case, training-only for the single net and the ensemble and the total uncertainty from the bootstrap procedure. It can be noted that increasing the number of networks in an ensemble is clearly beneficial to reduce the training uncertainty. The total uncertainty estimate from the bootstrap procedure is used to define uncertainty bands on the estimated efficiency parametrization.

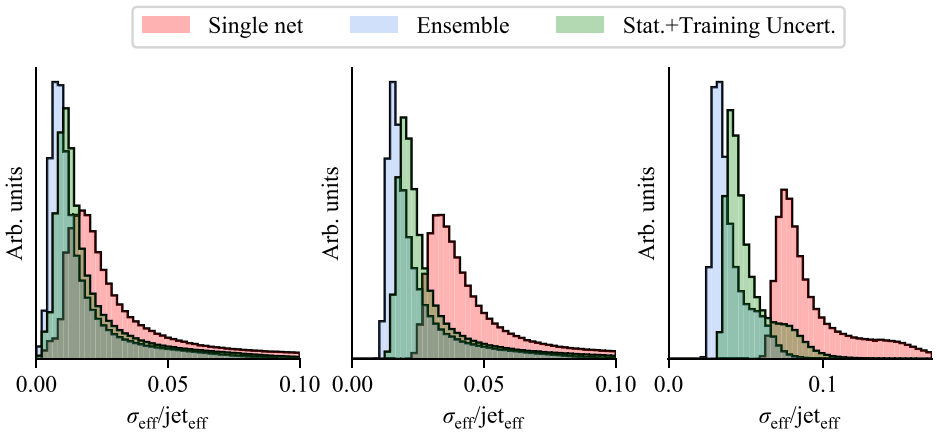


Fig. 11 The relative uncertainty on the predicted efficiency described in the text for the different jet flavors: b-jet (left), *c*-jet (middle), light-jet (right). The Stat + Training uncertainty represents the total uncertainty estimated with the bootstrap procedure

Acknowledgements EG and JS were supported by the NSF-BSF Grant 2017600 and the ISF Grant 2871/19. JS research was partially supported by the Israeli Council for Higher Education (CHE) via the Weizmann Data Science Research Center.

CB, FADB, GF, MK and VI research was partially supported by the grant "Sviluppo di algoritmi innovativi di Deep Learning per dati altamente sparsificati e applicazione all'identificazione di particelle prodotte nei decadimenti del bosone di Higgs negli esperimenti a LHC".

Funding Open access funding provided by Università degli Studi di Roma La Sapienza within the CRUI-CARE Agreement.

Declarations

Conflicts of interest On behalf of all authors, the corresponding authors state that there is no conflict of interest.

References

1. The ATLAS Collaboration (2019) ATLAS b-jet identification performance and efficiency measurement with $t\bar{t}$ events in pp collisions at $\sqrt{s} = 13TeV$. Eur Phys J C 79: 11, arXiv:1907.05120
2. The ATLAS Collaboration (2018) Search for the decay of the Higgs Boson to charm quarks with the ATLAS experiment. Phys Rev Lett 120: 211802, arXiv:1802.04329
3. The ATLAS Collaboration (2018) Observation of $H \to b\bar{b}$ decays and VH production with the ATLAS detector. Phys Lett B 786: 59, arXiv:1808.08238
4. Rogozhnikov A (2016) Reweighting with boosted decision trees. J Phys Conf Ser 762: 11, arXiv:1608.05806
5. Cranmer K, Pavez J, Louppe G (2016) Approximating likelihood ratios with calibrated discriminative classifiers. arXiv:1506.02169
6. Stoye M, Brehmer J, Louppe G, Pavez J, Cranmer K (2018) Likelihood-free inference with an improved cross-entropy estimator. arXiv:1808.00973
7. Andreassen A, Nachman B (2019) Neural networks for full phase-space reweighting and parameter tuning. arXiv:1907.08209
8. Battaglia P, et al (2018) Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261
9. Battaglia P, et al (2020) Graph neural networks in particle physics, arXiv:2007.13681
10. The ATLAS Collaboration (2007) Tagging rate function B-tagging, ATL-PHYS-PUB-2007-011
11. The D0 Collaboration (2008) Evidence for production of single top quarks, Phys Rev D 78
12. Wolf TMH (2018) Higgs from top to bottom: Discovery of the Higgs boson coupling to top quarks with the ATLAS detector, ISBN: 978-94-028-1302-9
13. The ATLAS Collaboration (2015) Search for the $b\bar{b}$ decay of the Standard Model Higgs boson in associated (W/Z)H production with the ATLAS detector. JHEP01 69, arXiv:1409.6212
14. The CMS Collaboration (2018) Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV. JINST 13: P05011, arXiv:1712.07158
15. The ATLAS Collaboration (2016) Search for an additional, heavy Higgs boson in the $H \to ZZ$ decay channel at $\sqrt{ss} = 8TeV$ in pp collision data with the ATLAS detector. Eur Phys J C 76: 45, arXiv:1507.09530
16. The CMS Collaboration (2018) Observation of Higgs boson decay to bottom quarks. Phys Rev Lett 121: 121801, arXiv:1808.08242
17. Sugiyama M, Suzuki T, Kanamori T (2012) Density ratio estimation in machine learning. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9781139035613
18. Serviansky H et al Set2Graph: learning graphs from sets, arXiv:2002.08772