

RECEIVED: July 14, 2022

REVISED: March 9, 2023

ACCEPTED: March 30, 2023

PUBLISHED: April 28, 2023

Identifying quenched jets in heavy ion collisions with machine learning

Lihan Liu,^a Julia Velkovska,^a Yilun Wu^a and Marta Verweij^b

^a*Department of Physics and Astronomy, Vanderbilt University,
PMB 401807, 2301 Vanderbilt Place, Nashville, TN 37235, U.S.A.*

^b*Department of Physics, Utrecht University,
Heidelberglaan 8 3584 CS Utrecht, The Netherlands*

E-mail: lihan.liu@vanderbilt.edu, julia.velkovska@vanderbilt.edu,
yilun.wu@vanderbilt.edu, m.verweij@uu.nl

ABSTRACT: Measurements of jet substructure in ultra-relativistic heavy ion collisions suggest that the jet showering process is modified by the interaction with the quark-gluon plasma. Modifications of the hard substructure of jets can be explored with modern data-driven techniques. In this study, a machine learning approach to the identification of quenched jets is designed. Jet showering processes are simulated with a jet quenching model JEWEL and a non-quenching model PYTHIA 8. Sequential substructure variables are extracted from the jet clustering history following an angular-ordered sequence and are used in the training of a neural network built on top of a long short-term memory network. We show that this approach successfully identifies the quenching effect in the presence of the large uncorrelated background of soft particles created in heavy-ion collisions.

KEYWORDS: Jets and Jet Substructure, Quark-Gluon Plasma

ARXIV EPRINT: [2206.01628](https://arxiv.org/abs/2206.01628)

Contents

1	Introduction	1
2	Data sample	3
3	Supervised machine learning	7
3.1	Feature engineering	7
3.2	Training of LSTM-based neural network	8
3.3	Hyper-tuning	10
3.4	Robustness	11
4	Results	13
5	Conclusion	13
A	Training with Jewel Vacuum Events	16

1 Introduction

Highly energetic partons (quarks and gluons) are expected to lose energy while traversing the extremely hot and dense quark-gluon plasma (QGP) created in ultra-relativistic heavy ion collisions [1]. This phenomenon, known as “jet quenching”, was discovered at the Relativistic Heavy Ion Collider (RHIC) through the observation of suppression of high transverse momentum (p_T) hadrons [2–5] and back-to-back dihadron correlations [6]. These observations were confirmed at the Large Hadron Collider (LHC) and extended into a larger kinematic range [7–11].

In addition to measurements on high p_T hadrons, which are the leading fragments of jets, the LHC opened new opportunities for direct observations of jet quenching and jet modifications in the QGP. Significant dijet transverse momentum asymmetry [12–15] and suppressed jet production [16–18] were observed in lead-lead (PbPb) collisions. Jet-hadron correlations gave insights into how the energy lost by the jets is distributed in the medium produced in the collisions. Measurements of the jet shapes (transverse momentum radial profile) [19–21] and the jet fragmentation functions [22, 23] were performed to further investigate the mechanism of the jet shower modifications in the QGP. To quantify the medium modifications, measurements in proton-proton (pp) collisions [24–32] are used as a reference for jet modifications in the medium. The jet splitting function has been measured in both pp and PbPb collisions [33–35], through the usage of a jet grooming algorithm that is able to split (“decluster”) a single jet into two subjets and locate the hard splitting by removing softer wide-angle radiation contributions. The hard splitting, characterized by

two well-separated subjects (“two-prong” structure), provides access to the early stages in the parton shower evolution.

In recent years many machine learning applications have been developed for studies in high-energy physics [36]. For the identification of quenched jets, a previous study of this classification problem [37] explored several ways of jet representation and different neural network architectures. It showed some promising results of deep learning techniques applied for the study of quenched jets in QGP at simulation level. However, the application of the convolutional neural network (CNN) often involves image preprocessing techniques, such as rotation, and normalization of the pixel intensity, which are not universal. These techniques are usually designed for a specific jet species, for example the boosted W/Z jets as studied in ref. [38], and may use a particular jet substructure feature, as studied in refs. [39, 40] for QCD jets. In addition, as a common practice in the implementation of modern Monte Carlo event generators, events associated with small cross-sections which often produce high p_T jets, can be over-sampled for statistical benefits, and event weights are assigned accordingly for compensation. How the event weights fit into the development of a machine learning approach is not well studied. Such problem is addressed in ref. [39] and a re-weighting method is designed. The sample weights are assigned based on the so-called effective sample number [41], which are not related to event cross-sections. For a more straight forward application to experimental data, it is desirable to have sample distributions corresponding to realistic cross-sections. Another challenge is that in the experiment the jets exist in the presence of a thermal background of uncorrelated particles created in heavy-ion collisions. Therefore, before the feasibility of machine learning can be established, techniques such as event mixing and background subtraction should be applied to simulated events, after which the jet finding algorithms can be performed. Such problem is addressed in ref. [39] and the authors found a decrease in the performance due to the presence of the underlying event. A similar conclusion is reached in ref. [42].

In this study, we use the successive splittings of the parton shower as input for a long short-term memory (LSTM) [43, 44] neural network. These splittings are obtained using jet declustering procedures in which all the splittings from the primary Lund plane [45] are extracted. The LSTM neural network is a special type of recurrent neural network that is capable of learning long-term dependencies and, therefore, has the potential to provide insight on the parton shower evolution in the medium. It has been used for jet tagging based on the primary Lund sequence [45, 46], showing a reasonable performance in W tagging, Top tagging and quark/gluon discrimination. Such tagging problems have also been studied in heavy-ion collisions with CNN [47]. In addition, the LSTM neural network shows its ability of jet grooming with a carefully designed training strategy [48]. We use the LSTM neural network and perform supervised machine learning in which a well-trained classifier is able to distinguish quenched jets from vacuum jets on a jet-by-jet basis. A similar approach has previously been explored in ref. [37], but without consideration of the recoiling scattering centers and the underlying event.

We present a machine learning approach to the identification of quenched jets in the presence of a large uncorrelated underlying event as present in heavy ion collisions. In section 2, we describe the procedure of event simulation and the subtraction procedure on the uncorrelated underlying event. In section 3, we describe how the supervised machine learning is performed, including a general feature engineering method on jets, a training

method with event weights taken into consideration, a method of hyper-tuning and a study of robustness. In section 4, we test the discrimination power of the trained classifier for quenched against non-quenched jets. Finally, we present our conclusions and outlook in section 5. The full code and data used in this study are available as open-source [49].

2 Data sample

In this study, we simulate hard scattering events with Monte Carlo event generators PYTHIA 8 [50] and JEWEL v2.2.0 [51, 52] at a center-of-mass energy per nucleon pair ($\sqrt{s_{\text{NN}}}$) of 5.02 TeV. A minimum outgoing parton transverse momentum $\hat{p}_{\text{T}} = 100$ GeV. PYTHIA 8 is used to simulate dijet events in pp collisions, while JEWEL simulates dijet events in heavy-ion collisions including parton-medium interactions. An important consequence of the parton-medium interactions implemented in JEWEL is the medium response. In our studies, we use JEWEL with keeping track of the recoils to explore the possible extreme. It has to be pointed out that such implementation of the medium response is a model-dependent component, while other generators may implement their own descriptions of the medium response. More details can be found in ref. [53]. In addition to the quenching descriptions, JEWEL provides simulations on the parton showering process in “vacuum” mode, which relies on PYTHIA6. Jet substructures observables are not identical in JEWEL vacuum mode and PYTHIA8, but the differences are small. Considering that in heavy-ion studies PYTHIA8 events are commonly used and compared to data, we choose PYTHIA8 events over JEWEL vacuum events as a baseline in which quenching effects are absent. More details about the differences between these two models can be found in appendix A, including the comparison on jet substructure variables and the comparison between two neural networks that are trained with the two models treated as the negative class respectively.

To simulate the thermal background originating from the hadronization of all partons in the quark-gluon plasma, which is not correlated with the jets, dijet events from both event generators are embedded into underlying events, which have general features similar to those observed experimentally. Following the implementation in ref. [53], the particles in the uncorrelated underlying events are generated from a Boltzmann distribution in p_{T} , and uniform distribution in pseudorapidity η and azimuthal angle φ . We prepare the uncorrelated underlying events using two different multiplicity settings, corresponding roughly to the mid-central (40-50%) and the most-central (0-10%) PbPb collisions at $\sqrt{s_{\text{NN}}} = 5.02$ TeV, as inferred from published experiment results [54]. More details about the uncorrelated underlying events are listed in table 1. With the inclusion of correlated and uncorrelated underlying event backgrounds, the influence on substructure variables introduced by particles not originating from the parton shower can be investigated.

The mixed event, constructed by embedding a dijet event into an uncorrelated underlying event, contains a hard scattering in the presence of the soft thermal background in heavy-ion collisions. The recoiling partons simulated from JEWEL are independent from the underlying event, but they are also present in the mixed event. Their 4-momenta have two components: a part which comes from the interactions with jet partons, and a thermal component that should be properly subtracted. To eliminate the energy and momentum

Values	Mid-central	Most-central
Pseudorapidity Interval	$ \eta < 3$	$ \eta < 3$
Total Event Multiplicity	1700	7000
Corresponding Centrality Interval	40-50%	0-10%
Average Yield $\langle \frac{dN}{d\eta} \rangle$	283	1167
Average Transverse Momentum $\langle p_T \rangle$	0.9 GeV	1.2 GeV
Average Transverse Momentum Density $\langle \rho \rangle$	35.0 GeV/Area	227.1 GeV/Area

Table 1. Parameters of the simulated underlying events.

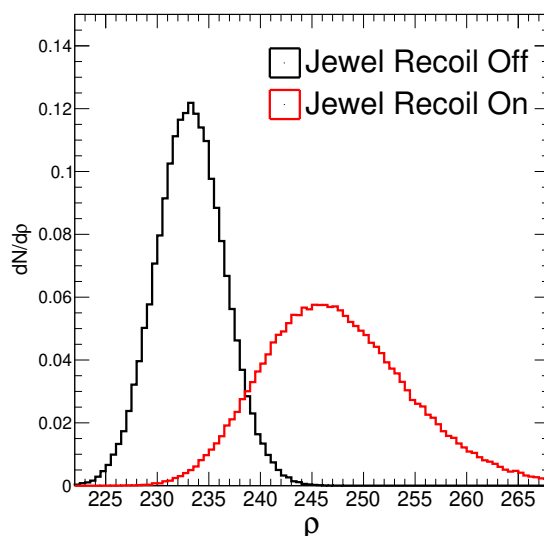


Figure 1. The median grid energy density ρ (GeV per unit area) used in the background subtraction procedure, for JEWEL dijet events with recoils on/off mixed with most-central thermal events.

contribution of thermal background particles into the jets, we perform background subtraction on the mixed events. The event-wide constituent subtraction method [55] is used with maximum distance parameter between the signal and background particles $\Delta R_{\max} = 0.3$, using p_T weight $\alpha = 1$. To illustrate how the recoiling partons are handled in the background subtraction procedure, in figure 1 we compare the distribution of the median grid energy density ρ , which is used in the background subtraction procedure, with recoiling partons turned on or off. The value of ρ with recoils on is larger than that with recoils off, which implies that more thermal energy is subtracted when the recoiling partons are present, as should be the case. Particles that remain after subtraction are clustered into jets with the anti- k_T algorithm [56] with a distance parameter $R = 0.4$.

After jet finding, each jet is reclustered with the Cambridge/Aachen algorithm [57] from its constituents to form a pairwise and angular-ordered structure. We then apply the soft drop declustering [58] procedure, continuously removing soft branches and keeping the

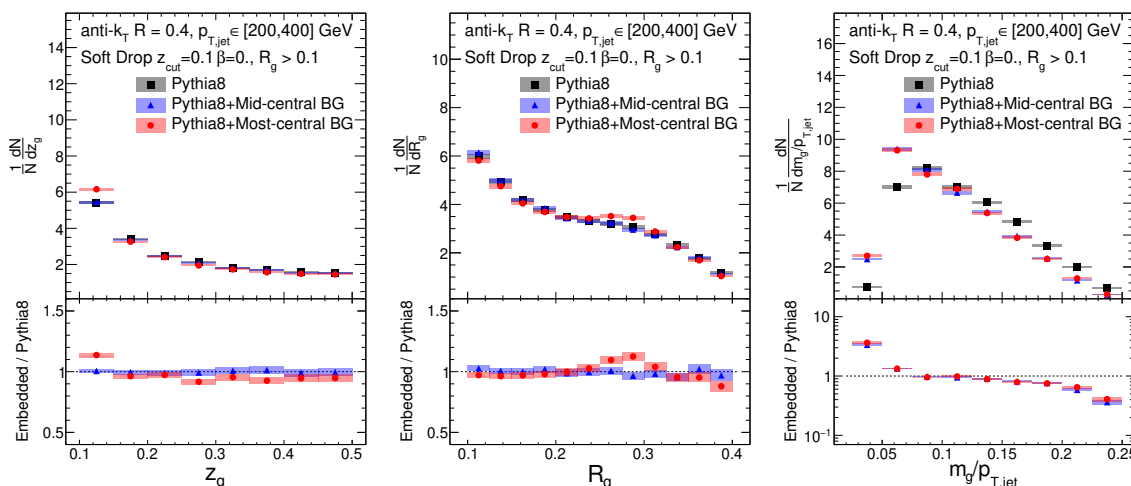


Figure 2. Distribution of substructure variables of the groomed jets with and without embedding.

splittings that satisfy the soft drop condition,

$$z_g \equiv \frac{\min(p_{T1}, p_{T2})}{p_{T1} + p_{T2}} > z_{\text{cut}} \left(\frac{\Delta R}{R_0} \right)^\beta, \quad (2.1)$$

where p_{T1} and p_{T2} are the transverse momenta of the two separated subjects, ΔR is their angular distance, and R_0 is the jet distance parameter (or jet radius), in our case $R_0 = 0.4$. In this study, we set the free parameters to values $z_{\text{cut}} = 0.1$, and $\beta = 0$. We also applied a cut of $p_{T,\text{jet}} > 200$ GeV to optimize the purity in locating the hard splittings [59]. To evaluate the background subtraction procedure in figure 2 we compare the distributions of jet substructure variables for jets simulated with PYTHIA and for PYTHIA jets embedded into thermal background events with particle multiplicities corresponding to mid-central and most-central PbPb collision events after the background subtraction. Ideally, if the background subtraction corrects both the jet energy and the jet substructure, the results will be identical. The top panels of figure 2 show the jet splitting function z_g , the angular distance between the subjects R_g , and the groomed jet mass m_g divided by the jet p_T , for jets propagating in vacuum (simulated with PYTHIA) before and after the embedding into the underlying PbPb events and the corresponding background subtraction. The bottom panels show the closure test of the subtraction procedure, e.g. the ratio of the jet substructure variables for jets embedded in the PbPb-backgrounds after subtraction, and the original PYTHIA jets. We find that for most of the phase space of interest the subtraction procedure works well, although there are some discrepancies that can be associated with mis-tagging of the sub-leading subject due to fluctuations in the thermal background, as suggested in ref. [59]. This indicates that large background fluctuations may produce effects that mimic a jet quenching signal when the soft drop grooming is applied. Other grooming methods may help control the mis-tagging and could be a subject of further investigations. In this study, we use the soft drop grooming imposing a jet p_T cut of $p_{T,\text{jet}} > 200$ GeV, which reduces mis-tagging.

To investigate the quenching effects on groomed jets, we compare the substructure variables of quenched jets with respect to vacuum jets, where both undergo embedding and

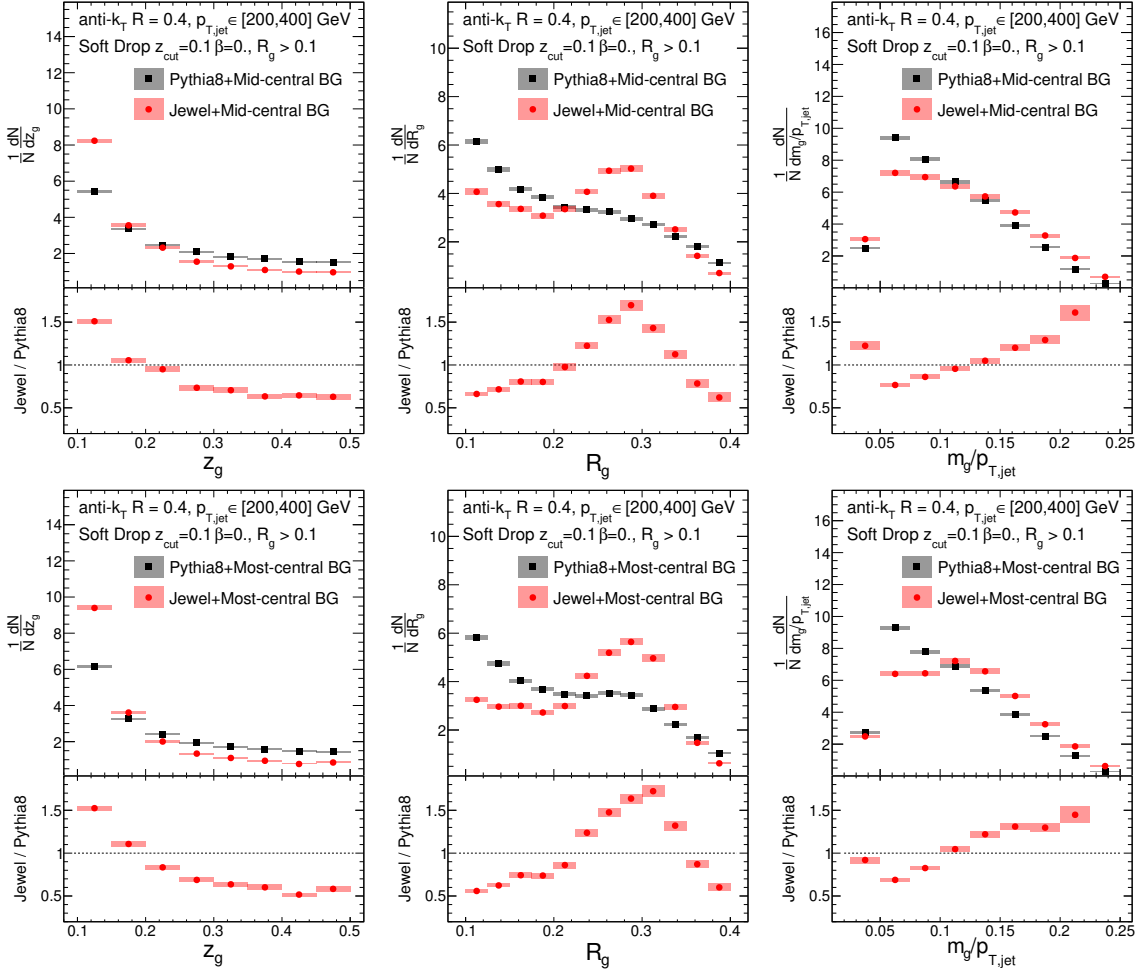


Figure 3. Distribution of substructure variables of the groomed jets for the mid-central (upper) and the most-central (lower) mixing scenarios.

background subtraction. The distributions of substructure variables of the groomed jets in the mid-central and the most-central mixing scenarios are shown in figure 3. Significant modifications are seen on all jet substructure variables. Additionally, in figure 4 we plot the primary Lund plane density [45],

$$\rho(z_g R_g, R_g) = \frac{1}{N_{\text{jet}}} \frac{d^2 N}{d \ln(z_g R_g) d \ln(1/R_g)}, \quad (2.2)$$

where z_g and R_g are taken from the first pair of hard branches that pass the soft drop condition (eq. (2.1)) for each jet. Strong modifications of the hard splittings with enhancement in wider (larger R_g) and more unbalanced (lower z_g) splittings are apparent in the JEWEL events. Next, we implement a machine learning approach to classify quenched and non-quenched jets.

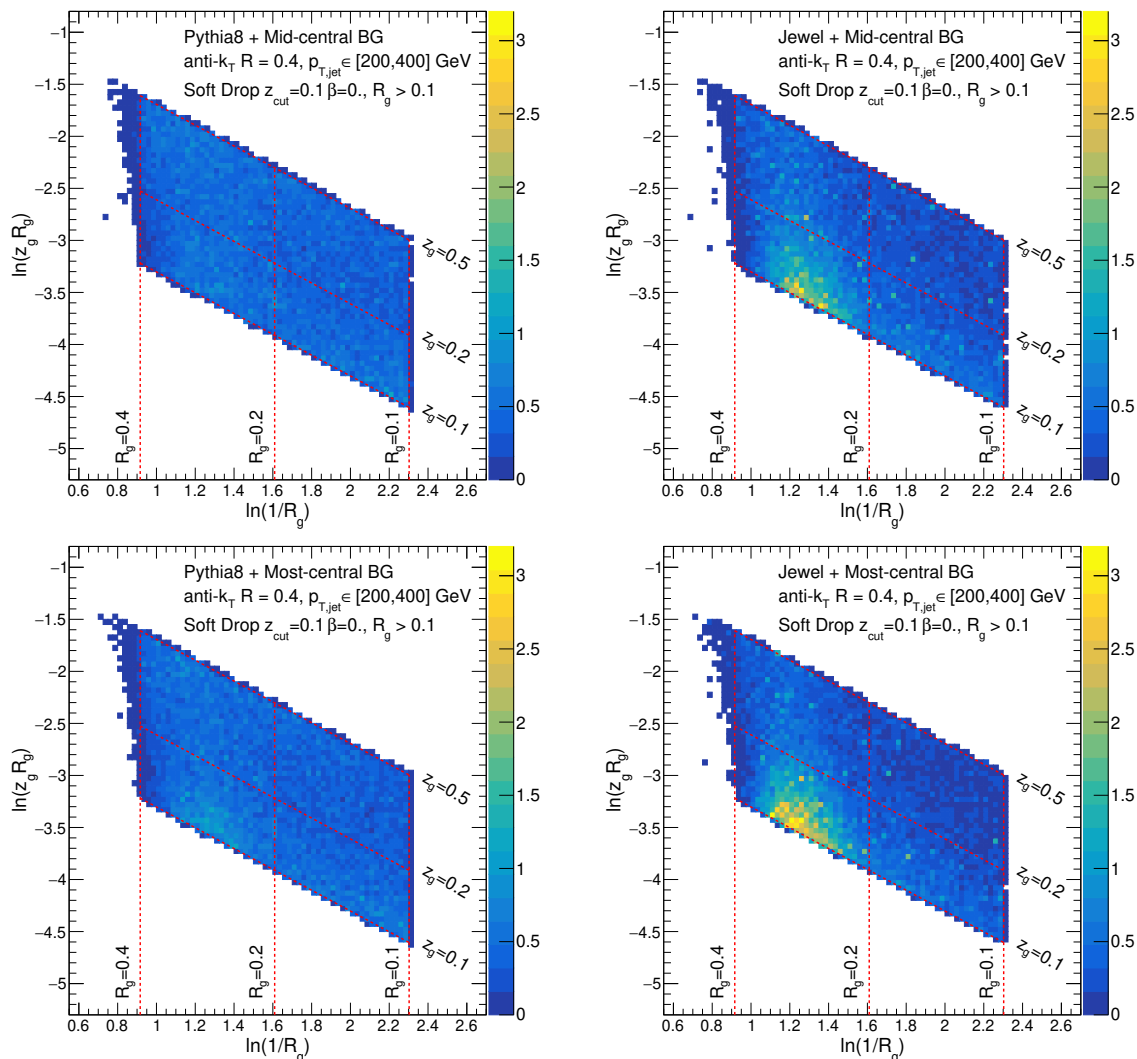


Figure 4. Primary Lund plane density for the mid-central (upper) and the most-central (lower) mixing scenarios.

3 Supervised machine learning

3.1 Feature engineering

The feature engineering procedure starts from jets reclustered with the Cambridge/Aachen algorithm. This leads to a clustering history with a binary tree structure, of which branches are separated based on their angular distance. The soft drop groomer continuously removes the softer branch, until finding the hard splitting that satisfies the soft drop condition (eq. (2.1)), after which the declustering continues on the harder subjet, as illustrated in figure 5.

Each splitting can be described with a feature vector, x_t , consisting of substructure variables including, but not limited to, the momentum fraction z , angular distance ΔR , perpendicular momentum k_{\perp} , and the invariant jet mass m_{inv} , which are all calculated

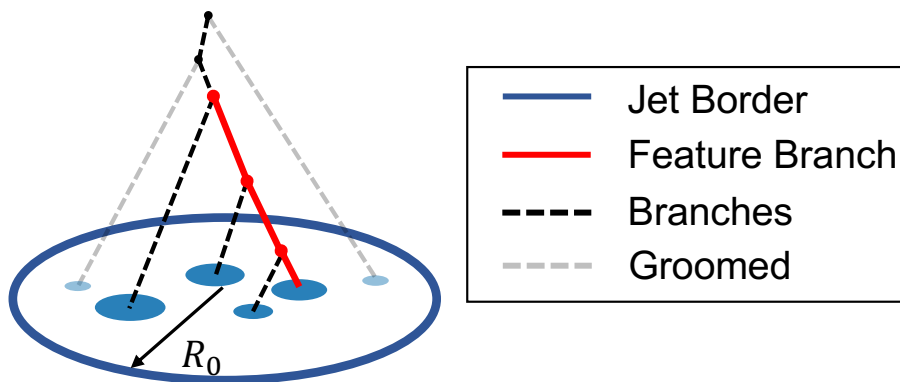


Figure 5. Jet clustering history with a binary tree structure.

from the pair of declustered subjects with:

$$\begin{aligned}
 z &= \frac{\min(p_{T,i}, p_{T,j})}{p_{T,i} + p_{T,j}}, \\
 \Delta R &= \sqrt{(\phi_i - \phi_j)^2 + (\eta_i - \eta_j)^2}, \\
 k_{\perp} &= \min(p_{T,i}, p_{T,j}) * \Delta R, \\
 m_{\text{inv}} &= \sqrt{(E_i + E_j)^2 - (\mathbf{p}_i + \mathbf{p}_j)^2}, \\
 x_t &= [z, \Delta R, k_{\perp}, m_{\text{inv}}],
 \end{aligned}
 \tag{3.1}$$

where i, j denote subjects at the declustering step t . In this way, sequential vectors $[x_0, \dots, x_t, \dots]$ are extracted, and are then used in the training of a neural network.

3.2 Training of LSTM-based neural network

A classification problem, that is, whether the hot and dense QGP medium modifies the jet showering process, is investigated based on a supervised machine learning strategy. Jets simulated with PYTHIA 8 are labeled as 0, in the later context referring to the non-quenched or the negative class. Conversely, those simulated with JEWEL are labeled as 1, referring to the quenched or the positive class. Both the positive and the negative class contains jet samples clustered from 400k events. Each dataset is split into a training and a validation set in a 1:1 ratio. After cuts on jet kinematics ($p_{T,jet}, |\eta|$), and cuts on jet substructure (z_g, R_g) are applied, there are about 40k jets in the training/validation datasets. For each jet, sequential feature vectors are extracted from their clustering history, as described in section 3.1. Considering that the lengths of the sequential feature vectors may vary, all the sequences in the same batch have post-sequence zeros as pads, such that they have the same length after padding. The zero-padded batch of sequential feature vectors is fed to a neural network built with LSTM layers and fully connected (FC) layers, of which the implementation is provided by the PyTorch framework [60]. Details about the parameters of the neural network architecture are further discussed in section 3.3.

A common LSTM unit [44] is composed of a cell, an input gate, an output gate, and a “forget” gate. The cell remembers relevant information throughout the processing of

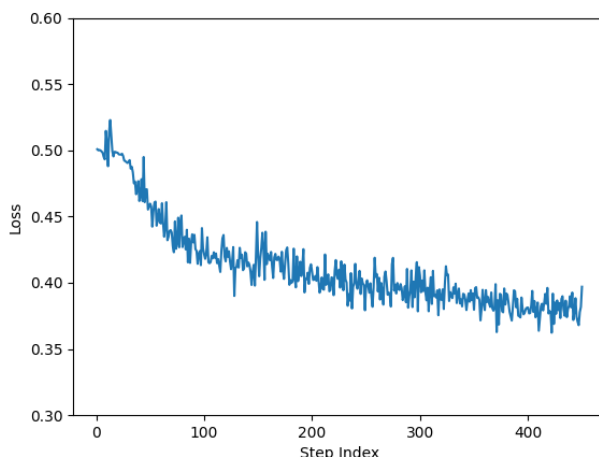


Figure 6. Examples of batch loss decreasing in the training process.

sequences, and the three gates regulate the flow of information into and out of the cell. The output of the LSTM network at the last step¹ is directed to FC layers which map high-dimensional values to a vector of predictive values, from which the batch losses are calculated. We use the weighted mean squared error (MSE) loss function:

$$l_{\text{MSE}} = \frac{\sum_{\text{batch}} \omega_i * (x_i - y_i)^2}{\sum_{\text{batch}} \omega_i}, \quad (3.2)$$

where x_i, y_i refer to the predictive label and the truth label of the i th jet, and ω_i is the event weight scaled from the event cross-section. During the training, learnable neural network parameters (e.g., coefficients or biases) are updated to minimize the batch loss during training, as shown in figure 6, using the “gradient decent” algorithm.

As mentioned previously, event weights come from generators due to over-sampling on small cross-section events, for the purpose of compensation. Such over-sampling can be configured with both JEWEL and PYTHIA8. Should the over-sampling issue be considered in the training process and what benefits may result from such consideration are not well-studied. In our study, jet samples have weights associated to the corresponding event cross-sections. The event weight is accounted for in the batch loss calculation (eq. 3.2), referring to as the weighted training scheme. For comparison, in the batch loss calculation the over-sampling can be ignored by setting weights to be unity, referring to as the un-weighted training scheme, even though over-sampling happens in event simulation. In reality, event weights can be replaced with unity for unbiased experimental samples. In our study the introduction of event weights has brought challenges to the training process,

¹The output from LSTM layer has a dimension of [batch size, sequence length, hidden size] (<https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html>). The hidden size is a configurable parameter, in other words a hyper-parameter.

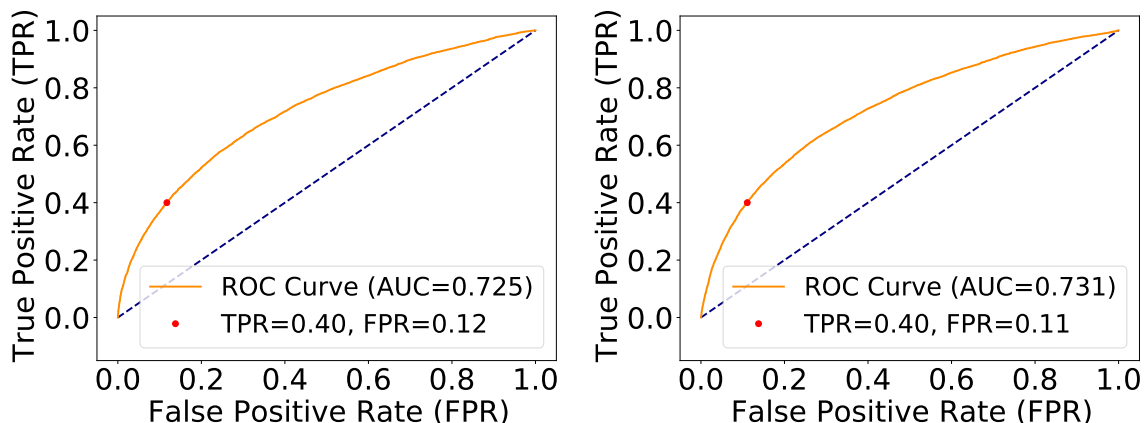


Figure 7. Comparison between ROC curves of the unweighted training scheme (left) and the weighted training scheme (right).

showing significant loss fluctuation when batch size is small. Larger batch sizes help to regulate the batch loss fluctuation, with the trade-off of making the computation heavier.

Here we compare the performance between the weighted and the unweighted training scheme, by looking into their receiver operating characteristic (ROC) curves. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The FPR quantifies the percentage of samples in the negative class that are falsely predicted as positive. For illustration, with a decision threshold of 0, all negative samples should have predictions larger than 0, which leads to $FPR=1$. The ROC curves are obtained with the help of the SCIKIT-LEARN package [62]. It should be pointed out that sample weights are taken into consideration in the calculation of TPR/FPR, details about which can be found online.² In our study, no significant difference is found between the performances of the two training schemes. Their area under curve (AUC) values are slightly different, while hyper-parameters may also affect AUC as studied in later section. In addition, no significant LSTM prediction dependence on jet p_T is found.

3.3 Hyper-tuning

The so-called hyper-parameters of a model, those that are not trainable but specify the details of the architecture and its training, are tuned with the Hyperopt [61] package. The hyper-tuning procedure helps locate the optimal hyper-parameter set on its search space, such that a local minimum can be avoided. The hyper-parameters related to the neural network architecture are the number of LSTM layers, and the input dimensions of two FC layers.³ The hyper-parameters related to the training process include the number of training epochs, batch size, learning rate and its decay factor. We scan over the hyper-parameter search space 50 times and arrange three trainings for each hyper-parameter set. The distribution of validation loss (defined in the same way as the batch loss, but

²See “sample_weight” parameter https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html.

³The input dimension of the first FC layer is equal to the LSTM hidden size.

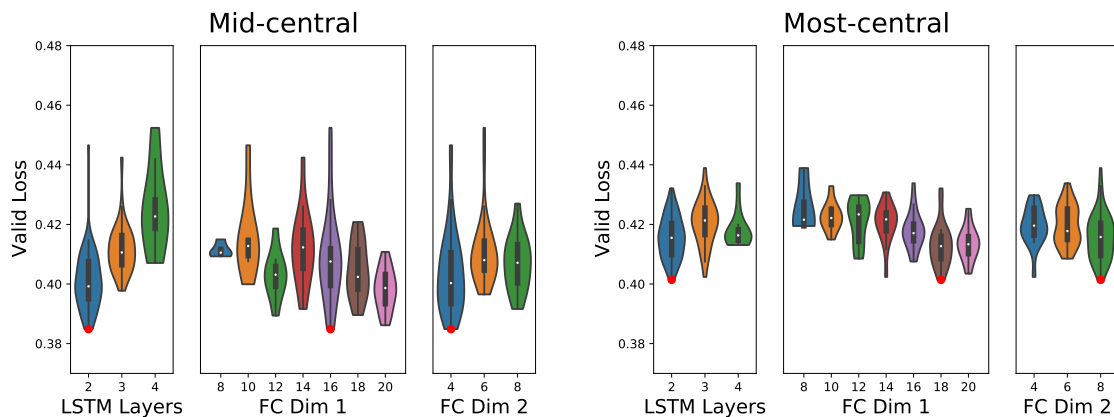


Figure 8. Distribution of validation loss for different architecture configurations, for the mid-central event mixing (left) and the most-central event mixing (right) respectively. The best models are indicated in red dots. The outer probability density contours cut off at the min/max values of the validation loss. The inner white point represents the median value. The inner thick bar spans the first and the third quartiles of the distributions.

Parameters	Mid-central	Most-central
No. of LSTM Layers	2	2
FC Dim 1	16	18
FC Dim 2	4	8
No. of Epochs	50	45
Batch Size	10000	18000
Learning Rate	0.0250	0.0371
Decay Factor	0.9689	0.9896

Table 2. List of hyper-parameters related to the neural network architecture and the training process, and their optimal values after hyper-tuning.

calculated with the validation dataset) for discrete hyper-parameter options are displayed in figure 8 with violin plots, showing both the probability density of the validation loss and its quartiles. The trained model with minimal validation loss shown with the red dot is selected as the best trained model, with the optimal hyper-parameter values listed in table 2.

3.4 Robustness

The trained neural network can be used as a classifier, making predictions on the probability for a jet to be quenched. Here, the distributions of the raw LSTM output from the top three best-trained networks are shown in figure 9, which exhibit a non-deterministic behavior in the metric. During the training, samples of the negative class are labeled with 0, while samples of the positive class are labeled with 1, without intermediate labels in between. Hence, the metric that quantifies the probability for a jet to be quenched, which is established during the training process, suffer from the stochastic nature of machine

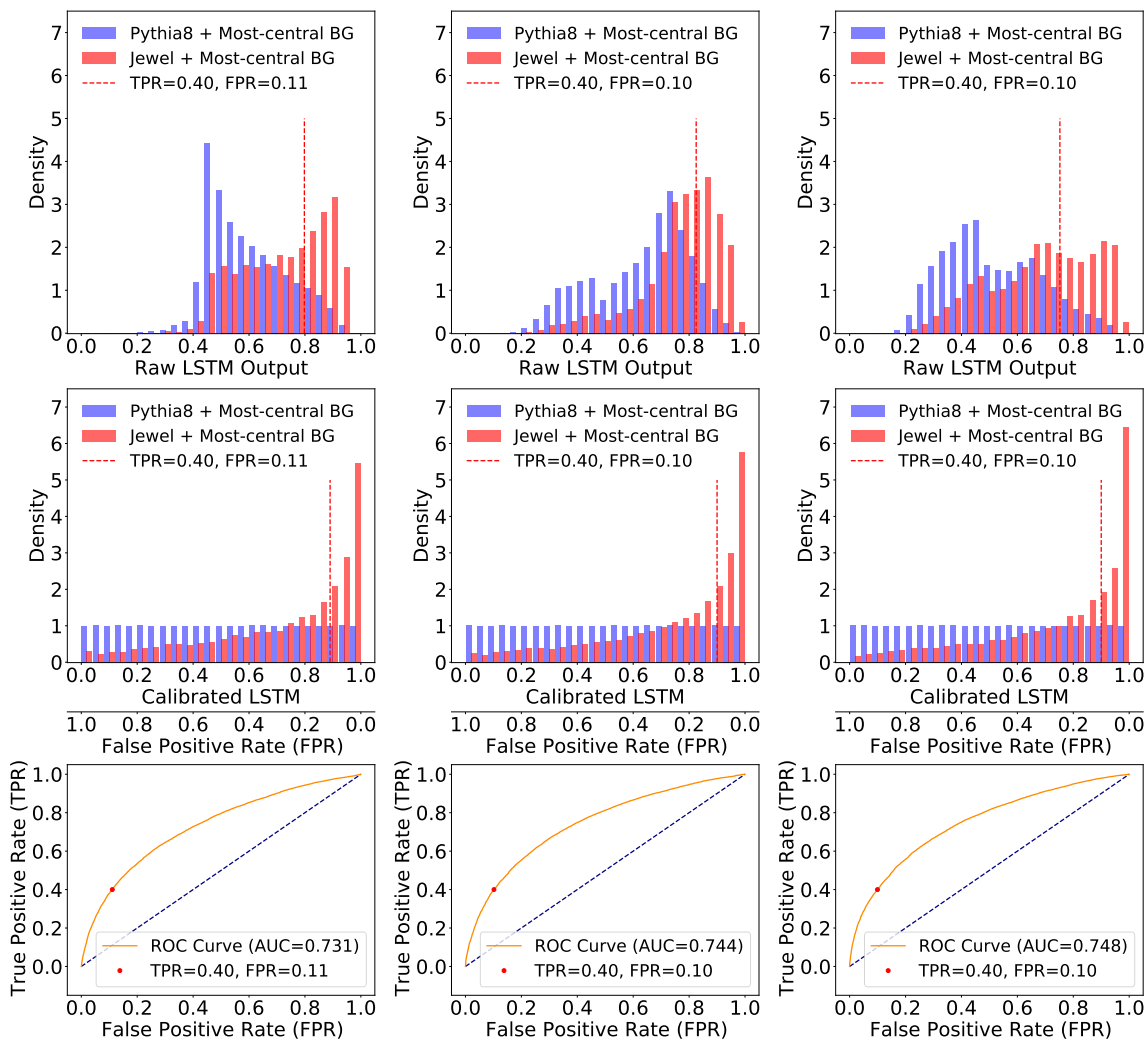


Figure 9. Distributions of the raw LSTM output from top three best-trained neural networks (top panels). Distribution of calibrated LSTM output (middle panels), and the corresponding ROC curve (bottom panels). The red lines/points corresponds to thresholds determined with $\text{TPR}=0.4$.

learning. Such non-determinism is rooted in the randomness of selecting samples to form a batch. To solve this issue, a calibration method is designed.

The calibration method is designed such that samples of the negative class are uniformly distributed along the newly established metric, meanwhile the relative quantitative relationships between the samples' raw LSTM outputs are kept in order. The new metric can be obtained with the help of the ROC curve. The new metric is bound to the FPR, meanwhile the threshold on the raw LSTM output corresponding to a specific FPR value can be determined from the ROC curve. The distribution of the raw LSTM output from trained neural networks and their values after calibration are shown in figure 9 (top and middle panels, respectively). For the three well-trained classifiers our calibration method produces almost identical distributions of the calibrated LSTM values.

Given the distributions of the LSTM outputs, either raw or calibrated, it can be seen that not all samples in the positive class (jets from JEWEL) are predicted to be quenched by the trained neural network. To determine if a jet is quenched, a threshold should be used, such that jets that have predictions larger than such a threshold would be classified as quenched, while those with predictions smaller than the threshold would be considered non-quenched. In the subsequent discussion use the threshold determined from $\text{TPR}=0.4$. With this threshold, the positive samples from JEWEL are divided into top 40% subset, and bottom 60% subset. It can be seen from the red point on the ROC curve that this TPR corresponds to a FPR of roughly 10%, indicating that about 10% of the samples in the negative class have predictions larger than the threshold. Further, we consider the top 40% JEWEL jets as quenched and compare their substructure variables with the remaining 60%, as well as those simulated with PYTHIA.

4 Results

Samples of the positive class (JEWEL) are divided into two subsets, consisting of top 40% and bottom 60% samples, respectively, based on the raw LSTM output. To better understand the relation between the LSTM output and the quenching effects, we plot the distributions of the substructure variables of the groomed jets in figure 10, and the primary Lund plane density for the hard splittings after grooming in figure 11, for two subsets separately. Samples of the negative class (PYTHIA) that serve as a baseline are also shown for comparison.

These two subsets of samples show quenching effects at different levels. The top 40% samples of the positive class show severe modifications of the jet substructure with enhancement of wider and softer splittings. In comparison, the bottom 60% samples exhibit a quenching pattern that is similar to that of the negative class. Two neural networks that are trained at two different multiplicity scenarios behave consistently, and show a similar ability in quantifying quenching effects. This indicates that the machine learning approach presented here successfully handles the uncorrelated underlying events independent of the background density.

5 Conclusion

Our study has shown a promising machine learning approach to identifying jet quenching in the presence of uncorrelated underlying event background. Sequential substructure variables extracted from the jet clustering history carry information about the parton showering process with or without quenching effects, and can be learned by an LSTM neural network. The well-trained neural network can then be used to select jets at different quenching levels. We also studied the non-deterministic behavior of the supervised machine learning algorithm. With the help of the ROC curve, a calibration method is designed to avoid the non-determinism.

The procedure of extracting sequential substructure variables can be used as a general feature engineering method on jets. In comparison with jet image approaches, which utilize

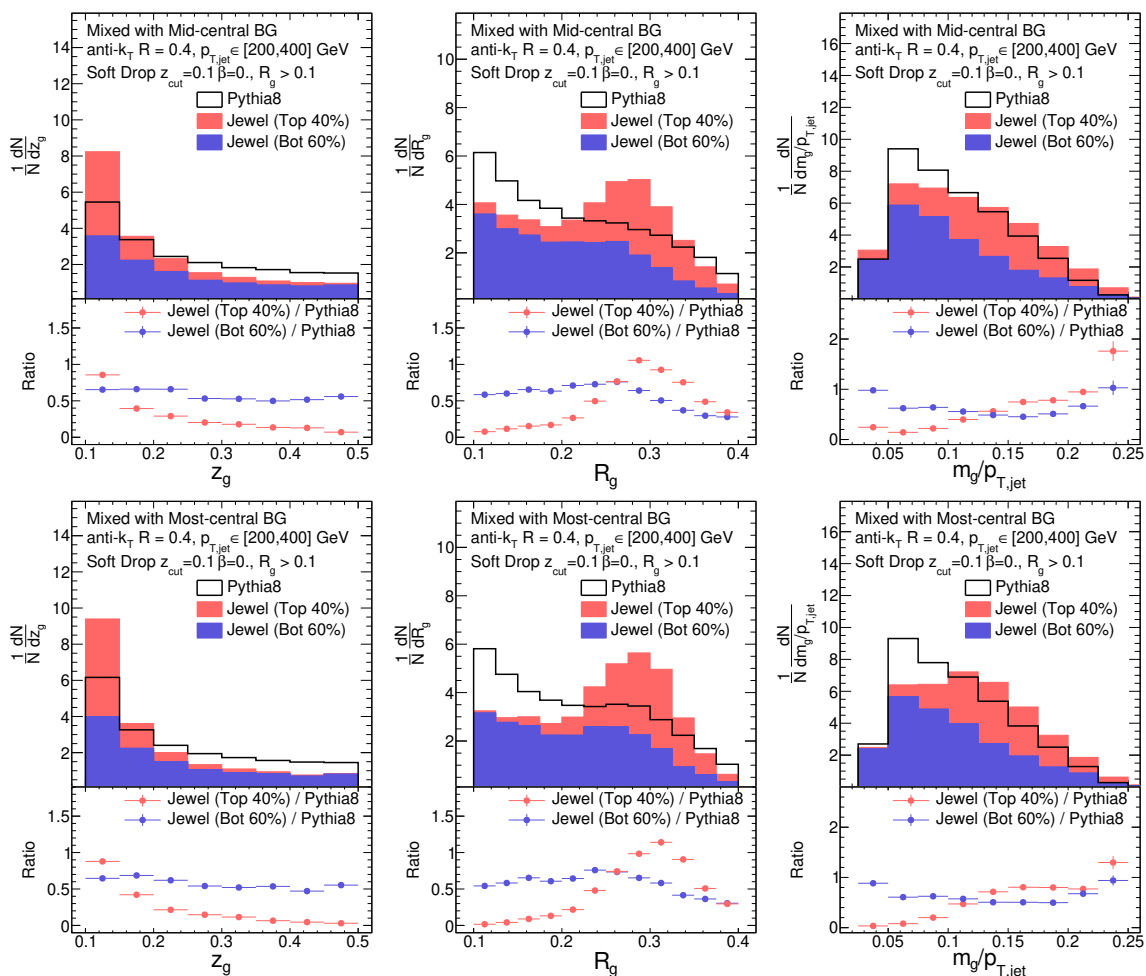


Figure 10. Distribution of the substructure variables of the groomed jets in the mid-central events (top) and the most-central events (bottom). Samples of the positive class (JEWEL) are plotted with stacked histograms.

jet substructure in image pre-processing, our feature engineering method extracts sequential variables directly and explicitly from the jet substructure. The angular-ordered clustering history, which is in the basis of the sequential features carries information about the parton showering process.

In the supervised machine learning setup, the truth labels, either quenched or non-quenched, are related to two different Monte Carlo simulators that produce two types of dijet events. In a real experiment, such labels can simply be replaced with the collision systems, either pp collisions or heavy ion collisions. Unlike the study in ref. [39] which uses the jet energy loss ratio χ_{jh} in a regression setup, our approach does not rely on simulation-only variables.

The well-trained neural network is capable of quantifying quenching effects, making it possible to divide jets simulated from a quenching model (JEWEL) into two subsets. The two separated subsets of jets have shown differences in the jet substructure variables,

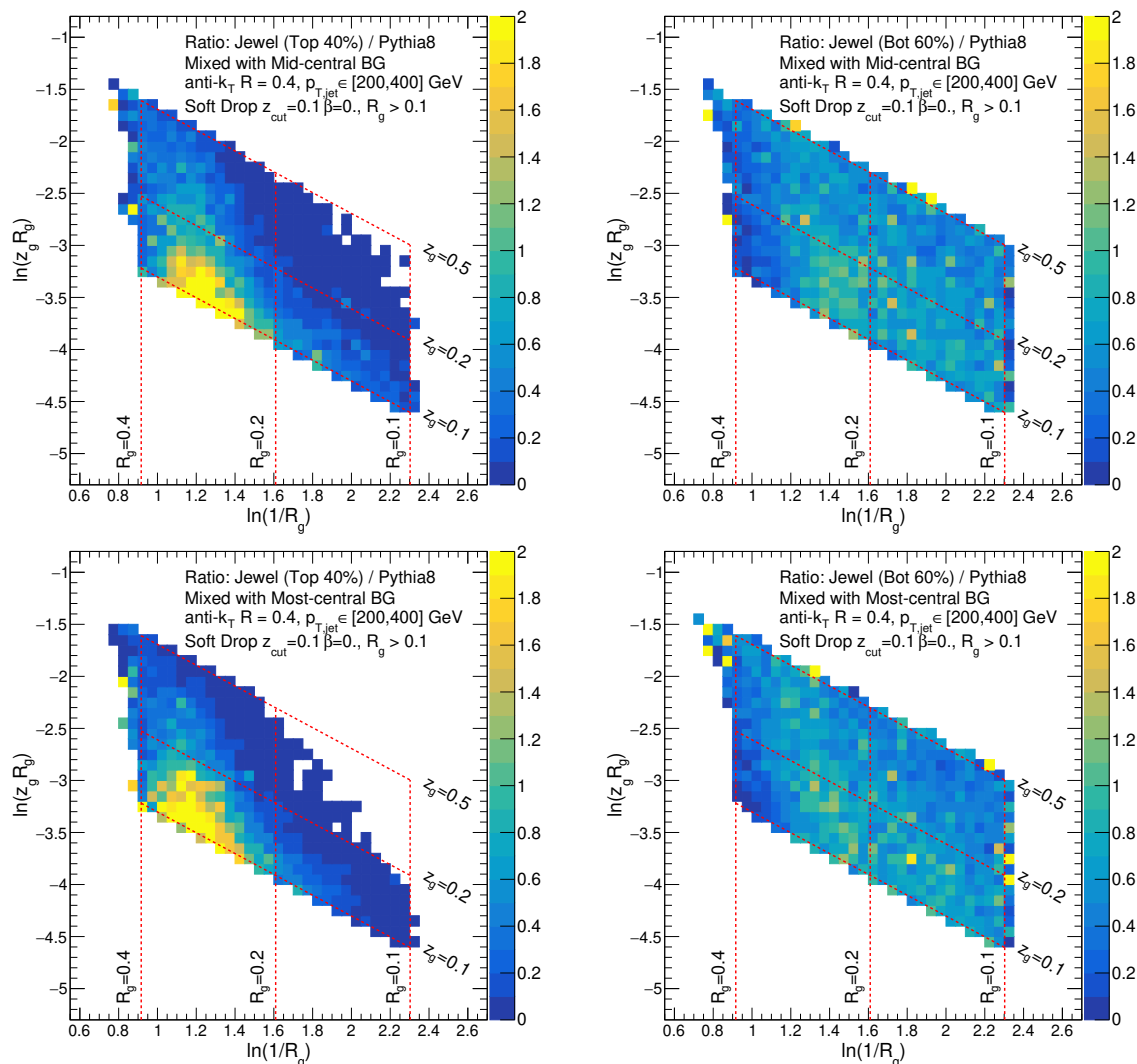


Figure 11. Primary Lund plane density for the top 40% JEWEL samples (left) and the bottom 60% JEWEL samples (right). Plots for the mid-central and the most-central event mixing scenarios are shown in the upper and lower panels, respectively.

related to two different parton showering patterns: a vacuum-like pattern, similar to what is implemented in PYTHIA 8, and a medium-like pattern, which produces highly quenched jets. Moreover, the medium-like pattern tends to mark up a zone on the Lund diagram, which may help extract information about medium properties. As pointed in ref. [53], the jet transport coefficient \hat{q} , which is of interest, is connected to boundaries of the quenching zone on the Lund diagram.

The techniques involved in our study, such as the background subtraction and the jet grooming, have been previously applied to experimental data and their effectiveness is well-studied. Thus, our machine learning approach is very promising in the exploration of different jet classification topics with real experimental data.

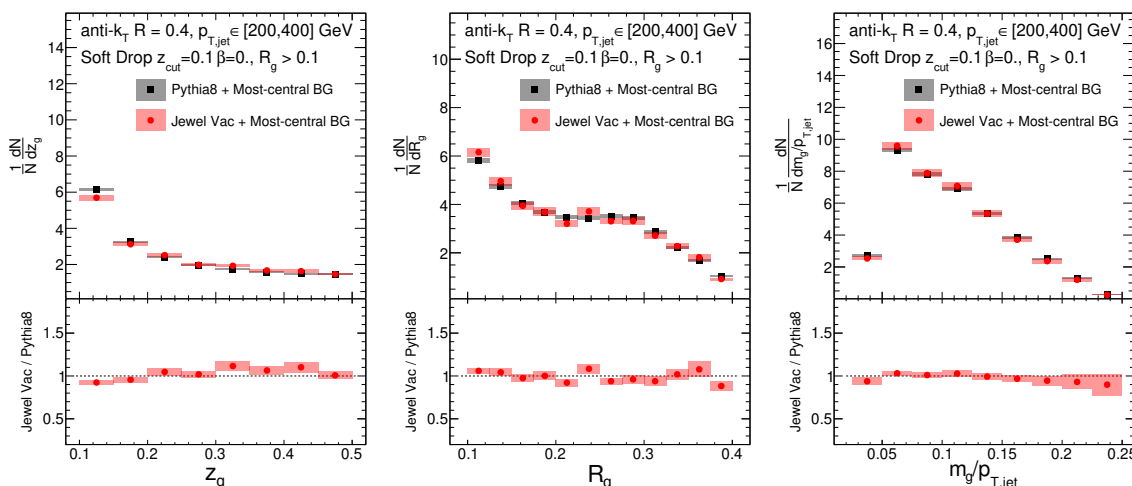


Figure 12. Comparison on the substructure variables between two event generators: JEWEL Vacuum and PYTHIA8, with the same embedding + subtraction procedures applied as those described in section 2.

Acknowledgments

We thank James Mulligan, Raghav Kunnawalkam Elayavalli, and Yi Chen for their helpful suggestions. We thank the Advanced Computing Center for Research and Education (ACCRE) at Vanderbilt University for providing computing resources. This work was supported in part by the U.S. Department of Energy Grant No. DE-FG05-92ER40712, and by the Alfred P. Sloan Foundation, whose contributions are gratefully acknowledged.

A Training with Jewel Vacuum Events

As in section 2, jet substructure observables are not identical in JEWEL vacuum mode and PYTHIA8, but the differences are small, which can be seen from figure 12.

The training procedure described in section 3.2 is repeated with JEWEL medium events as the positive class and JEWEL vacuum events as the negative class. Events of both classes have gone through the embedding and the background subtraction procedures, the same as those illustrated in section 2. The training is done with the weighted scheme, where event weights are taken into consideration in the loss calculation.

The well-trained LSTM network is then used as a classifier, producing the distribution of the raw LSTM output and their calibrated values, as shown in In figure 13. Such classifier shows its power in identifying quenching effects, similar to the one that is trained with JEWEL Medium and PYTHIA8. We also present the distributions of the substructure variables of the groomed jets and the primary Lund diagrams, for the top 40% and the bottom 60% JEWEL medium samples respectively, as shown in figure 14 and figure 15.

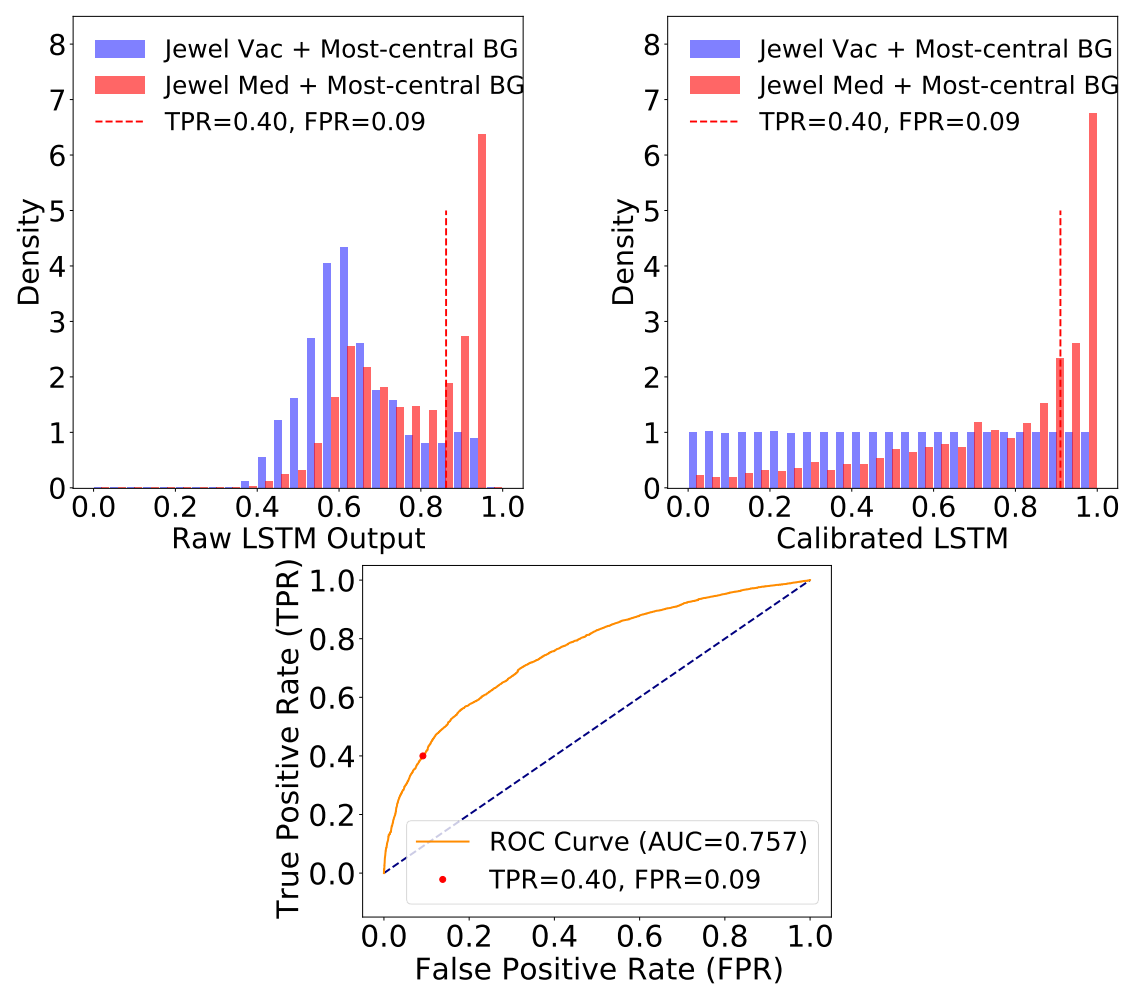


Figure 13. Distributions of the raw LSTM output (upper left), their calibrated values (upper right) and the corresponding ROC curve (lower).

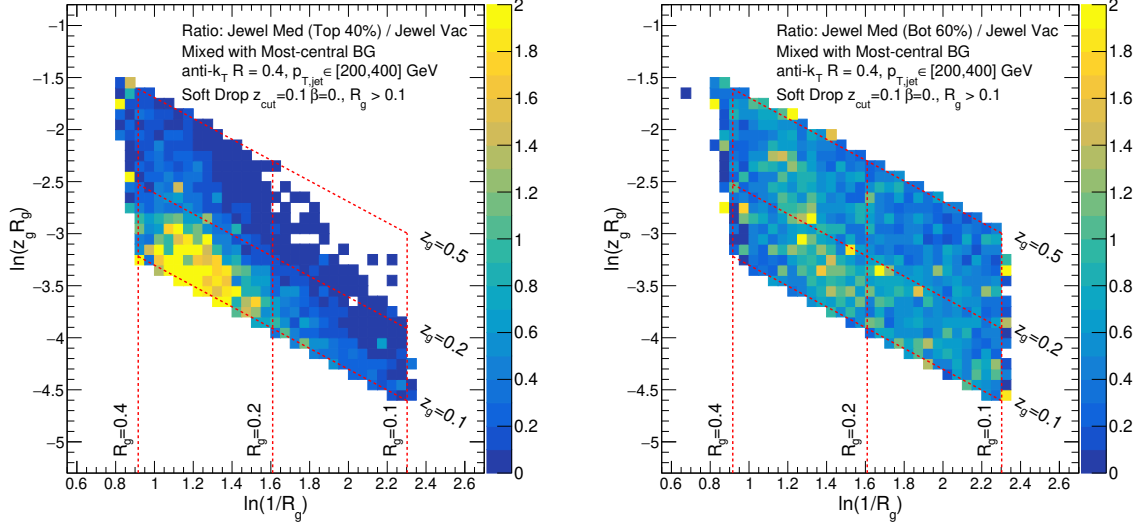


Figure 14. Primary Lund plane density for the top 40% JEWEL medium samples (left) and the bottom 60% samples (right), compared with JEWEL vacuum samples.

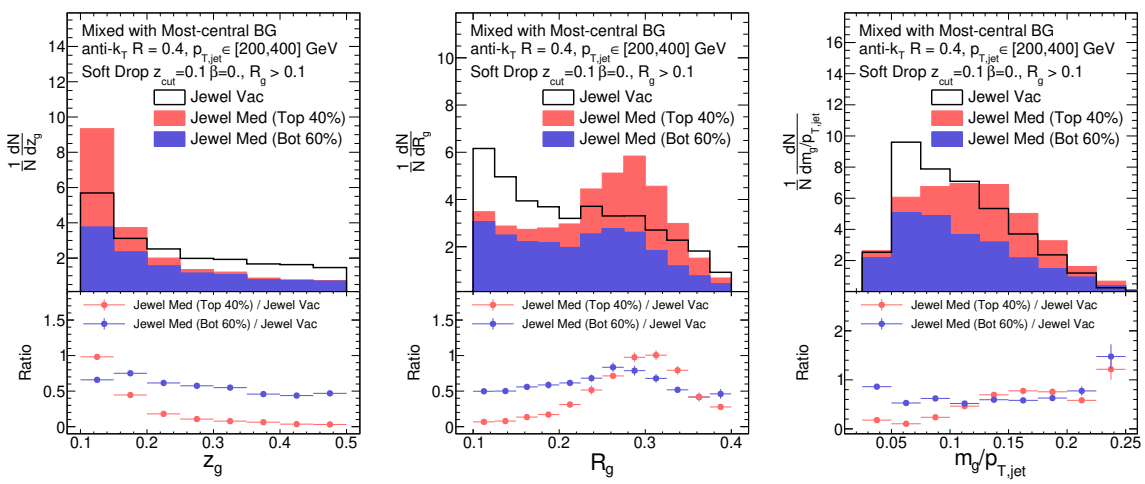


Figure 15. Distribution of the substructure variables of the groomed jets. Samples of the positive class (JEWEL medium) are plotted with stacked histograms.

Open Access. This article is distributed under the terms of the Creative Commons Attribution License ([CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)), which permits any use, distribution and reproduction in any medium, provided the original author(s) and source are credited. SCOAP³ supports the goals of the International Year of Basic Sciences for Sustainable Development.

References

- [1] J.D. Bjorken, *Energy loss of energetic partons in quark-gluon plasma: possible extinction of high p_T jets in hadron-hadron collisions*, [FERMILAB-PUB-82-059-THY](#), Fermilab, Batavia, IL, U.S.A. (1982) [[INSPIRE](#)].
- [2] PHENIX collaboration, *Suppression of hadrons with large transverse momentum in central Au+Au collisions at $\sqrt{s_{NN}} = 130$ GeV*, *Phys. Rev. Lett.* **88** (2002) 022301 [[nucl-ex/0109003](#)] [[INSPIRE](#)].
- [3] STAR collaboration, *Centrality dependence of high p_T hadron suppression in Au+Au collisions at $\sqrt{s_{NN}} = 130$ GeV*, *Phys. Rev. Lett.* **89** (2002) 202301 [[nucl-ex/0206011](#)] [[INSPIRE](#)].
- [4] PHENIX collaboration, *Suppressed π^0 production at large transverse momentum in central Au+Au collisions at $\sqrt{s_{NN}} = 200$ GeV*, *Phys. Rev. Lett.* **91** (2003) 072301 [[nucl-ex/0304022](#)] [[INSPIRE](#)].
- [5] STAR collaboration, *Transverse momentum and collision energy dependence of high p_T hadron suppression in Au+Au collisions at ultrarelativistic energies*, *Phys. Rev. Lett.* **91** (2003) 172302 [[nucl-ex/0305015](#)] [[INSPIRE](#)].
- [6] STAR collaboration, *Disappearance of back-to-back high p_T hadron correlations in central Au+Au collisions at $\sqrt{s_{NN}} = 200$ GeV*, *Phys. Rev. Lett.* **90** (2003) 082302 [[nucl-ex/0210033](#)] [[INSPIRE](#)].
- [7] ALICE collaboration, *Suppression of charged particle production at large transverse momentum in central Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV*, *Phys. Lett. B* **696** (2011) 30 [[arXiv:1012.1004](#)] [[INSPIRE](#)].
- [8] ALICE collaboration, *Particle-yield modification in jet-like azimuthal di-hadron correlations in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV*, *Phys. Rev. Lett.* **108** (2012) 092301 [[arXiv:1110.0121](#)] [[INSPIRE](#)].
- [9] CMS collaboration, *Study of high- p_T charged particle suppression in Pb-Pb compared to pp collisions at $\sqrt{s_{NN}} = 2.76$ TeV*, *Eur. Phys. J. C* **72** (2012) 1945 [[arXiv:1202.2554](#)] [[INSPIRE](#)].
- [10] ATLAS collaboration, *Measurement of charged-particle spectra in Pb+Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV with the ATLAS detector at the LHC*, *JHEP* **09** (2015) 050 [[arXiv:1504.04337](#)] [[INSPIRE](#)].
- [11] CMS collaboration, *Charged-particle nuclear modification factors in PbPb and pPb collisions at $\sqrt{s_{NN}} = 5.02$ TeV*, *JHEP* **04** (2017) 039 [[arXiv:1611.01664](#)] [[INSPIRE](#)].
- [12] ATLAS collaboration, *Observation of a centrality-dependent dijet asymmetry in lead-lead collisions at $\sqrt{s_{NN}} = 2.77$ TeV with the ATLAS detector at the LHC*, *Phys. Rev. Lett.* **105** (2010) 252303 [[arXiv:1011.6182](#)] [[INSPIRE](#)].

- [13] CMS collaboration, *Observation and studies of jet quenching in PbPb collisions at nucleon-nucleon center-of-mass energy = 2.76 TeV*, *Phys. Rev. C* **84** (2011) 024906 [[arXiv:1102.1957](#)] [[INSPIRE](#)].
- [14] CMS collaboration, *Jet momentum dependence of jet quenching in PbPb collisions at $\sqrt{s_{NN}} = 2.76$ TeV*, *Phys. Lett. B* **712** (2012) 176 [[arXiv:1202.5022](#)] [[INSPIRE](#)].
- [15] CMS collaboration, *Measurement of transverse momentum relative to dijet systems in PbPb and pp collisions at $\sqrt{s_{NN}} = 2.76$ TeV*, *JHEP* **01** (2016) 006 [[arXiv:1509.09029](#)] [[INSPIRE](#)].
- [16] ATLAS collaboration, *Measurement of the jet radius and transverse momentum dependence of inclusive jet suppression in lead-lead collisions at $\sqrt{s_{NN}} = 2.76$ TeV with the ATLAS detector*, *Phys. Lett. B* **719** (2013) 220 [[arXiv:1208.1967](#)] [[INSPIRE](#)].
- [17] ALICE collaboration, *Measurement of charged jet suppression in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV*, *JHEP* **03** (2014) 013 [[arXiv:1311.0633](#)] [[INSPIRE](#)].
- [18] ALICE collaboration, *Measurement of jet suppression in central Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV*, *Phys. Lett. B* **746** (2015) 1 [[arXiv:1502.01689](#)] [[INSPIRE](#)].
- [19] CMS collaboration, *Modification of jet shapes in PbPb collisions at $\sqrt{s_{NN}} = 2.76$ TeV*, *Phys. Lett. B* **730** (2014) 243 [[arXiv:1310.0878](#)] [[INSPIRE](#)].
- [20] ALICE collaboration, *Measurement of jet radial profiles in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV*, *Phys. Lett. B* **796** (2019) 204 [[arXiv:1904.13118](#)] [[INSPIRE](#)].
- [21] ATLAS collaboration, *Measurement of angular and momentum distributions of charged particles within and around jets in Pb+Pb and pp collisions at $\sqrt{s_{NN}} = 5.02$ TeV with the ATLAS detector*, *Phys. Rev. C* **100** (2019) 064901 [*Erratum ibid.* **101** (2020) 059903] [[arXiv:1908.05264](#)] [[INSPIRE](#)].
- [22] CMS collaboration, *Measurement of jet fragmentation in PbPb and pp collisions at $\sqrt{s_{NN}} = 2.76$ TeV*, *Phys. Rev. C* **90** (2014) 024908 [[arXiv:1406.0932](#)] [[INSPIRE](#)].
- [23] ATLAS collaboration, *Measurement of jet fragmentation in Pb+Pb and pp collisions at $\sqrt{s_{NN}} = 2.76$ TeV with the ATLAS detector at the LHC*, *Eur. Phys. J. C* **77** (2017) 379 [[arXiv:1702.00674](#)] [[INSPIRE](#)].
- [24] CMS collaboration, *Study of quark and gluon jet substructure in Z+jet and dijet events from pp collisions*, *JHEP* **01** (2022) 188 [[arXiv:2109.03340](#)] [[INSPIRE](#)].
- [25] CMS collaboration, *Measurements of the differential jet cross section as a function of the jet mass in dijet events from proton-proton collisions at $\sqrt{s} = 13$ TeV*, *JHEP* **11** (2018) 113 [[arXiv:1807.05974](#)] [[INSPIRE](#)].
- [26] ATLAS collaboration, *Measurement of soft-drop jet observables in pp collisions with the ATLAS detector at $\sqrt{s} = 13$ TeV*, *Phys. Rev. D* **101** (2020) 052007 [[arXiv:1912.09837](#)] [[INSPIRE](#)].
- [27] ATLAS collaboration, *Measurement of jet-substructure observables in top quark, W boson and light jet production in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector*, *JHEP* **08** (2019) 033 [[arXiv:1903.02942](#)] [[INSPIRE](#)].
- [28] ATLAS collaboration, *Measurement of the Lund jet plane using charged particles in 13 TeV proton-proton collisions with the ATLAS detector*, *Phys. Rev. Lett.* **124** (2020) 222002 [[arXiv:2004.03540](#)] [[INSPIRE](#)].

- [29] ATLAS collaboration, *Measurement of the soft-drop jet mass in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector*, *Phys. Rev. Lett.* **121** (2018) 092001 [[arXiv:1711.08341](#)] [[INSPIRE](#)].
- [30] ALICE collaboration, *Measurements of the groomed and ungroomed jet angularities in pp collisions at $\sqrt{s} = 5.02$ TeV*, *JHEP* **05** (2022) 061 [[arXiv:2107.11303](#)] [[INSPIRE](#)].
- [31] ALICE collaboration, *Measurements of the groomed jet radius and momentum splitting fraction with the soft drop and dynamical grooming algorithms in pp collisions at $\sqrt{s} = 5.02$ TeV*, [arXiv:2204.10246](#) [CERN-EP-2022-056] [[INSPIRE](#)].
- [32] ALICE collaboration, *First measurement of jet mass in Pb-Pb and p-Pb collisions at the LHC*, *Phys. Lett. B* **776** (2018) 249 [[arXiv:1702.00804](#)] [[INSPIRE](#)].
- [33] CMS collaboration, *Measurement of the splitting function in pp and Pb-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV*, *Phys. Rev. Lett.* **120** (2018) 142302 [[arXiv:1708.09429](#)] [[INSPIRE](#)].
- [34] STAR collaboration, *Differential measurements of jet substructure and partonic energy loss in Au+Au collisions at $\sqrt{s_{NN}} = 200$ GeV*, *Phys. Rev. C* **105** (2022) 044906 [[arXiv:2109.09793](#)] [[INSPIRE](#)].
- [35] ALICE collaboration, *Measurement of the groomed jet radius and momentum splitting fraction in pp and Pb-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV*, *Phys. Rev. Lett.* **128** (2022) 102001 [[arXiv:2107.12984](#)] [[INSPIRE](#)].
- [36] M. Feickert and B. Nachman, *A living review of machine learning for particle physics*, [arXiv:2102.02770](#) [[INSPIRE](#)].
- [37] L. Apolinário et al., *Deep learning for the classification of quenched jets*, *JHEP* **11** (2021) 219 [[arXiv:2106.08869](#)] [[INSPIRE](#)].
- [38] S. Macaluso and D. Shih, *Pulling out all the tops with computer vision and deep learning*, *JHEP* **10** (2018) 121 [[arXiv:1803.00107](#)] [[INSPIRE](#)].
- [39] Y.-L. Du, D. Pablos and K. Tywoniuk, *Deep learning jet modifications in heavy-ion collisions*, *JHEP* **21** (2020) 206 [[arXiv:2012.07797](#)] [[INSPIRE](#)].
- [40] Y.-L. Du, D. Pablos and K. Tywoniuk, *Jet tomography in heavy-ion collisions with deep learning*, *Phys. Rev. Lett.* **128** (2022) 012301 [[arXiv:2106.11271](#)] [[INSPIRE](#)].
- [41] Y. Cui et al., *Class-balanced loss based on effective number of samples*, [arXiv:1901.05555](#).
- [42] Y.S. Lai, J. Mulligan, M. Płoskoń and F. Ringer, *The information content of jet quenching and machine learning assisted observable design*, *JHEP* **10** (2022) 011 [[arXiv:2111.14589](#)] [[INSPIRE](#)].
- [43] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, *Neural Comput.* **9** (1997) 1735 [[INSPIRE](#)].
- [44] A. Sherstinsky, *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network*, *Physica D* **404** (2020) 132306.
- [45] F.A. Dreyer, G.P. Salam and G. Soyez, *The Lund jet plane*, *JHEP* **12** (2018) 064 [[arXiv:1807.04758](#)] [[INSPIRE](#)].
- [46] F.A. Dreyer and H. Qu, *Jet tagging in the Lund plane with graph networks*, *JHEP* **03** (2021) 052 [[arXiv:2012.08526](#)] [[INSPIRE](#)].
- [47] Y.-T. Chien and R. Kunnawalkam Elayavalli, *Probing heavy ion collisions using quark and gluon jet substructure*, [arXiv:1803.03589](#) [MIT-CTP 4947] [[INSPIRE](#)].

- [48] S. Carrazza and F.A. Dreyer, *Jet grooming through reinforcement learning*, *Phys. Rev. D* **100** (2019) 014014 [[arXiv:1903.09644](#)] [[INSPIRE](#)].
- [49] L. Liu, *Github repository for jet quenching studies with machine learning*, <https://github.com/ustcllh/JetML> (2022).
- [50] T. Sjostrand, S. Mrenna and P.Z. Skands, *A brief introduction to PYTHIA 8.1*, *Comput. Phys. Commun.* **178** (2008) 852 [[arXiv:0710.3820](#)] [[INSPIRE](#)].
- [51] K.C. Zapp, F. Krauss and U.A. Wiedemann, *A perturbative framework for jet quenching*, *JHEP* **03** (2013) 080 [[arXiv:1212.1599](#)] [[INSPIRE](#)].
- [52] R. Kunnawalkam Elayavalli and K.C. Zapp, *Medium response in JEWEL and its impact on jet shape observables in heavy ion collisions*, *JHEP* **07** (2017) 141 [[arXiv:1707.01539](#)] [[INSPIRE](#)].
- [53] H.A. Andrews et al., *Novel tools and observables for jet physics in heavy-ion collisions*, *J. Phys. G* **47** (2020) 065102 [[arXiv:1808.03689](#)] [[INSPIRE](#)].
- [54] ALICE collaboration, *Centrality dependence of the pseudorapidity density distribution for charged particles in Pb-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV*, *Phys. Lett. B* **772** (2017) 567 [[arXiv:1612.08966](#)] [[INSPIRE](#)].
- [55] P. Berta, M. Spousta, D.W. Miller and R. Leitner, *Particle-level pileup subtraction for jets and jet shapes*, *JHEP* **06** (2014) 092 [[arXiv:1403.3108](#)] [[INSPIRE](#)].
- [56] M. Cacciari, G.P. Salam and G. Soyez, *The anti- k_t jet clustering algorithm*, *JHEP* **04** (2008) 063 [[arXiv:0802.1189](#)] [[INSPIRE](#)].
- [57] Y.L. Dokshitzer, G.D. Leder, S. Moretti and B.R. Webber, *Better jet clustering algorithms*, *JHEP* **08** (1997) 001 [[hep-ph/9707323](#)] [[INSPIRE](#)].
- [58] A.J. Larkoski, S. Marzani, G. Soyez and J. Thaler, *Soft drop*, *JHEP* **05** (2014) 146 [[arXiv:1402.2657](#)] [[INSPIRE](#)].
- [59] J. Mulligan and M. Ploskon, *Identifying groomed jet splittings in heavy-ion collisions*, *Phys. Rev. C* **102** (2020) 044913 [[arXiv:2006.01812](#)] [[INSPIRE](#)].
- [60] A. Paszke et al., *PyTorch: an imperative style, high-performance deep learning library*, in *Advances in neural information processing systems* 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett eds., Curran Associates Inc. (2019), p. 8024 [[arXiv:1912.01703](#)] [[DOI:10.48550/arXiv.1912.01703](#)].
- [61] J. Bergstra, D. Yamins and D. Cox, *Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures*, in *Proceedings of the 30th international conference on machine learning*, S. Dasgupta and D. McAllester eds., volume 28 of *Proceedings of Machine Learning Research*, PMLR, Atlanta, GA, U.S.A., 17–19 June 2013, p. 115.
- [62] F. Pedregosa et al., *Scikit-learn: machine learning in Python*, *J. Machine Learn. Res.* **12** (2011) 2825.