Computing and Software for Big Science (2019) 3:12
https://doi.org/10.1007/s41781-019-0028-1

ORIGINAL ARTICLE

Topology Classification with Deep Learning to Improve Real-Time Event Selection at the LHC

T. Q. Nguyen[1] · D. Weitekamp III[2] · D. Anderson[1] · R. Castello[3] · O. Cerri[1] · M. Pierini[3] · M. Spiropulu[1] · J-R. Vlimant[1]

Received: 1 August 2018 / Accepted: 21 August 2019 / Published online: 31 August 2019

Abstract
We show how an event topology classification based on deep learning could be used to improve the purity of data samples selected in real time at the Large Hadron Collider. We consider different data representations, on which different kinds of multi-class classifiers are trained. Both raw data and high-level features are utilized. In the considered examples, a filter based on the classifier's score can be trained to retain ∼ 99% of the interesting events and reduce the false-positive rate by more than one order of magnitude. By operating such a filter as part of the online event selection infrastructure of the LHC experiments, one could benefit from a more flexible and inclusive selection strategy while reducing the amount of downstream resources wasted in processing false positives. The saved resources could translate into a reduction of the detector operation cost or into an effective increase of storage and processing capabilities, which could be reinvested to extend the physics reach of the LHC experiments.

Keywords Trigger · Deep learning · Topology classification · LHC

Introduction

The CERN Large Hadron Collider (LHC) collides protons every 25 ns. Each collision can result in any of hundreds of physics processes. The total data volume exceeds by far what the experiments could record. This is why the incoming data flow is typically filtered through a set of rule-based algorithms, designed to retain only events with particular signatures (e.g., the presence of a high-energy particle of some kind). Such a system, commonly referred to as *trigger*, consists of hundreds of algorithms, each designed to accept events with a specific topology. The ATLAS [1] and CMS [2] trigger systems are based on this idea. In their current implementation, given the throughput capability and the typical event size, these two experiments can write on disk ∼ 1000 events/s. A few processes, e.g., QCD multijet production, constitute the vast majority of the produced events. One is typically interested to select a fraction of these events for further studies. On the other hand, the main interest of the LHC experiments is related to selecting and studying the many rare processes which occur at the LHC. In a typical data flow, these events are overwhelmed by the large amount of QCD multijet events. The trigger system is put in place to make sure that the majority of these rare events are part of the stored ∼ 1000 events/s.

Trigger algorithms are typically designed to maximize the efficiency (i.e., the true-positive rate), resulting in a non-negligible false-positive rate and, consequently, in a substantial waste of resources at trigger level (i.e., data throughput that could have been used for other purposes) and downstream (i.e., storage disk, processing power, etc.).

The most commonly used selection rules are *inclusive*, i.e., more than one topology is selected by the same requirement. The so-called isolated-lepton triggers are a typical example of these kinds of algorithms. These triggers select events with a high-momentum electron or muon and no surrounding energetic particle, a typical signature of an interesting rare process, e.g., the production of a W boson decaying to a neutrino and an electron or muon. With such a requirement, one can simultaneously collect W bosons produced in the primary interaction (W events) or from the cascade decay of other particles, e.g., top quarks (mainly in $t\bar{t}$ events

✉ T. Q. Nguyen
thong@caltech.edu

[1] California Institute of Technology, Pasadena, USA

[2] University of California at Berkeley, Berkeley, USA

[3] Experimental Physics Department, CERN, Geneva, Switzerland

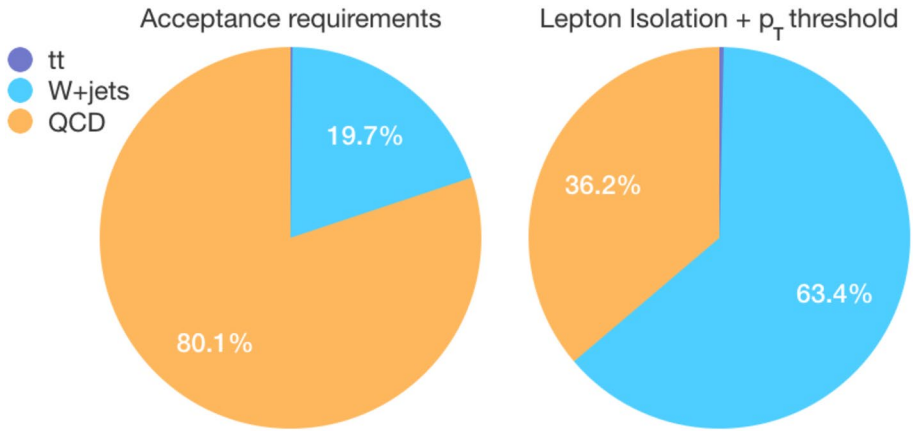

Fig. 1 Relative composition of the isolated-lepton sample after the acceptance requirement (left) and the trigger selection (right), as described in the text

where a top quark–antiquark pair is produced). A sample selected this way is dominated by W events, but it retains a substantial ($> 10\%$) contamination from QCD multijet. The $t\bar{t}$ contribution is smaller than 1%. Events from $t\bar{t}$ production are sometimes triggered by a set of dedicated lepton+jets algorithms, capable of using looser requirements on the lepton at the cost of introducing requirements on jets.[1] Due to this additional complexity, the use of these triggers in a data analysis comes with additional complications. For instance, the applied jet requirements produce distortions on offline distributions of jet-related quantities. To avoid having this effect, any typical data analysis applies a tighter offline selection. This means that many of the selected events close to the online-selection threshold are discarded. This is not necessarily the most cost-effective way to retain an unbiased data set for offline analysis.

In this paper, we investigate the possibility of using machine learning to classify events based on their topologies, serving as an additional clean-up algorithm at the trigger level. Doing so, one could customize the trigger-selection strategy on individual processes (depending on the physics goals) while keeping the selection loose and simple. As a benchmark case, we consider a stream of data selected by requiring the presence of one electron or muon with transverse momentum $p_{\mathrm{T}} > 23$ GeV [2] and a loose requirement on the isolation. Details on the applied selection can be found in Section "Data Set".

The considered benchmark sample is dominated by direct W production, with a sizable contamination from QCD multijet events and a small contribution of $t\bar{t}$ events. Other interesting processes (e.g., WW, WZ, and ZZ production) are usually selected with more exclusive and dedicated trigger algorithms (e.g., di-muon or di-electron triggers), or share the same kinematic properties of the two main interesting processes (W and $t\bar{t}$). For the sake of simplicity, we ignore these sub-leading processes in our study, without compromising the validity of our conclusions. Figure 1 shows the composition of a sample with one electron or muon within the defined acceptance ($p_{\mathrm{T}} > 22$ GeV and pseudorapidity $|\eta| = |-\log[\tan(\theta/2)]| < 2.6$, where θ is the polar angle), before and after applying the trigger requirements ($p_{\mathrm{T}} > 23$ GeV and loose isolation).

Such a loose set of requirements would translate into an event acceptance rate of ~ 690 Hz for a luminosity of 2×10^{34} cm^{-2} s^{-1}, well beyond the currently allocated budget for these triggers (typically ~ 200 Hz). We suggest that, using the score of our topology classifier, one could tune the amount of each process to be stored for further analysis, within the boundaries of the allocated resources. For instance, one might be interested to retain all the $t\bar{t}$ events and some fraction of W events, while rejecting the QCD multijet events. We envision two main applications: for a given total rate, one could loosen the baseline trigger requirements, increasing the acceptance efficiency at no cost. Or, for a given acceptance efficiency (true-positive rate), one could save resources by reducing the overall rate, rejecting the contribution of unwanted topologies (see Appendix A).

We consider several topology classifiers based on deep learning model architectures: fully connected deep neural networks (DNNs), convolutional neural networks (CNNs) [3], and recurrent neural networks such as Long–Short-Term-Memory networks (LSTMs) [4] and gated recurrent units (GRUs) [5]. We consider four different representations of the collision events: (1) a set of physics-motivated high-level features, (2) the raw image of the detector hits, (3) a sequence of particles, characterized by a limited set of basic features (energy, direction, etc.), and (4) an *abstract* representation of this list of particles as an image.

The paper is structured as follows. In the section "Data Set", we describe the four data representations. In the section "Model description", we describe the corresponding

[1] A jet is a spray of hadrons, typically originating from the hadronization of gluons and quarks produced in the proton collisions.

[2] In this paper, we set units in such a way that $c = \hbar = 1$.

classification models. Results are discussed in the section "Results". In the section "Impact on other topologies", we investigate the generalization properties of the four classifiers to scenarios of other topologies. We study the robustness of our classifiers against Monte Carlo simulation inaccuracy with pseudo-data in section "Robustness study". In the section "Related works", we briefly discuss applications of machine learning algorithms to similar problems. Conclusions are given in section "Conclusions". Appendix A describes a different scenario, in which the classifier is used to save resources by reducing the trigger acceptance rate, as opposed to using it to sustain a loose trigger selection that could otherwise require too many resources.

Data Set

Synthetic data corresponding to W, $t\bar{t}$, and QCD multijet production topologies are generated with 10^5 events per process ($3 \cdot 10^5$ events in total) using the PYTHIA8 event generation library [6]. The setup of the proton-beam simulation is loosely inspired by the LHC running configuration in 2015–2016: two proton beams, each with 6.5 TeV, generate on average 20 proton–proton collisions per crossing following a Poisson distribution.

Generated samples are processed with the DELPHES library [7], which applies a parametric model of a detector response. Detector performances are tuned to the CMS upgrade design foreseen for the High-Luminosity LHC [8], as implemented in the corresponding default card provided with DELPHES. We run the DELPHES *particle flow* (PF) algorithm, which combines the information from all the CMS detector components to derive a list of reconstructed particles, the so-called PF candidates. For each particle, the algorithm returns the measured energy and flight direction. Each particle is associated to one of three classes: charged particles, photons, and neutral hadrons. Jets are clustered from the reconstructed PF candidates, using the FASTJET [9] implementation of the anti-k_T jet algorithm [10], with jet-size parameter $R = 0.4$. The jet's b-tagging efficiency is parameterized as a function of jet's p_T and η in the default DELPHES CMS upgrade design card. The parameterized b-tagging efficiency is shown to provide a reasonable agreement with CMS [7].

The basic event representation consists of a list of reconstructed PF candidates. For each candidate q, the following information is given: (1) the particle four-momentum in Cartesian coordinates (E, p_x, p_y, p_z); (2) The particle three-momentum, computed from (1), in cylindrical coordinates: the transverse momentum p_T, the pseudorapidity η, and the azimuthal angle ϕ; (3) The Cartesian coordinates (x_vtx, y_vtx, z_vtx) of the particle point of origin. For all neutral particles, (0, 0, 0) is used in the absence of pointing information; (4) The electric charge; (5) The particle isolation with respect to charged particles (ChPFIso), photons (GammaPFIso), or neutral hadrons (NeuPFIso). For each particle class, the isolation is quantified as follows:

$$\mathrm{ISO} = \frac{\sum_{p \neq q} p_\mathrm{T}^p}{p_\mathrm{T}^q}, \tag{1}$$

where the sum extends over all the particles of the appropriate class with angular distance $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} < 0.3$ from the particle q.

The particle identity is categorized via a one-hot-encoded representation (*isChPar*, *isNeuHad*, *isGamma*), corresponding to a charged particle, a neutral hadron, or a photon. In addition, two boolean flags are stored (*isEle* and *isMu*) to identify if a given particle is an electron or a muon. In total, each particle is then described by 19 features.

The trigger selection is emulated by requiring all the events to include one isolated electron or muon with transverse momentum $p_\mathrm{T} > 23$ GeV and particle-based isolation ChISO + GammaISO + NeuISO < 0.45. This baseline selection, which follows the typical requirements of an inclusive single-lepton trigger algorithm, accepts ≈ 100 QCD multijet events and ≈ 176 W events for every $t\bar{t}$ event. Despite its large W and $t\bar{t}$ efficiency, this trigger selection comes with a large cost in terms of QCD multijet events written on disk and processed offline. The cost is even larger if the main physics target is $t\bar{t}$ events and the W contribution is seen as an additional source of background (e.g., in a high-statistics scenario, with all measurements of W properties limited in precision by systematic uncertainties).

All particles are ranked in decreasing order of p_T. For each event, the isolated lepton is the first entry of the list of particles. To avoid double counting of this isolated lepton ℓ as a charged particle, each charged particle q is required to have $\Delta R(q, \ell) > 10^{-4}$. In addition to the isolated lepton, we consider the first 450 charged particles, the first 150 photons, and the first 200 neutral hadrons. This corresponds to a total of 801 particles per event, each characterized by the 19 features described above. The choice of the numbers of particles is made, such that, on average, only 5% charged particles, 5% neutral hadrons, and 1% photons are ignored. Thanks to p_T ordering by particle category, what we remove carries small information. In early stages of this work, we experimented with tighter cuts on particle multiplicity without observing substantial difference. We verified that the particles which we ignore have typical p_T below 1 GeV. If fewer particles are found in the event, zero padding is used to guarantee a fixed length of the particle list across different events. The events are then stored as

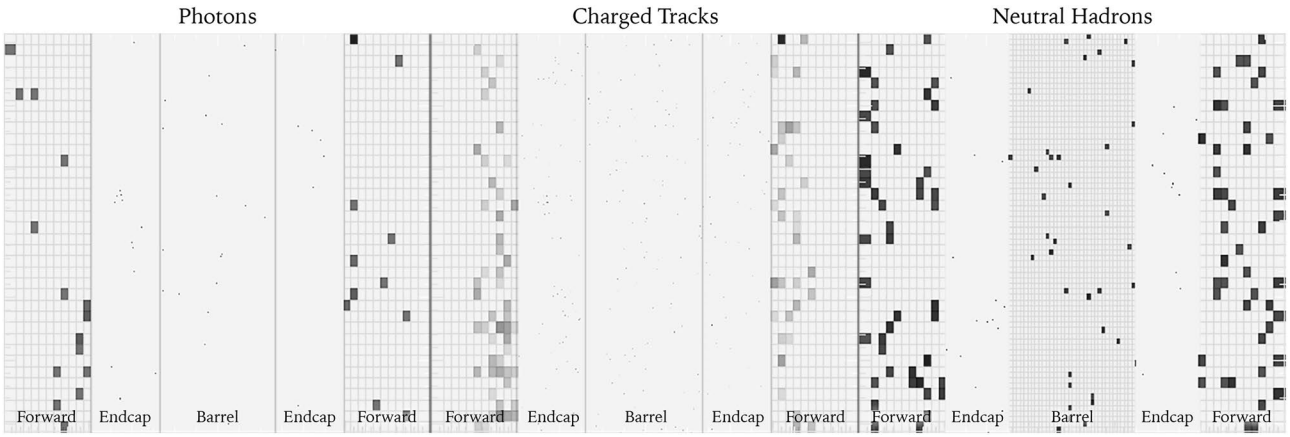

Fig. 2 An example of a $t\bar{t}$ event as the input of the raw-image classifier. Vertical and horizontal axes are the ϕ and η coordinates, respectively, of the sub-detectors

NumPy arrays in a set of compressed HDF5 files. The data set is planned to be released on the CERN OpenData portal, accessible at `opendata.cern.ch`.

In addition to this raw-event representation, we provide a list of physics-motivated high-level features, computed from the full event (the `HLF` data set):

- The scalar sum, S_{T}, of the p_{T} of all the jets, leptons, and photons in the event with $p_{\mathrm{T}} > 30$ GeV and $|\eta| < 2.6$.
- The missing transverse energy $E_{\mathrm{T}}^{\mathrm{miss}}$, defined as the absolute value of the missing transverse momentum, computed summing over the full list of reconstructed PF candidates:

$$E_{\mathrm{T}}^{\mathrm{miss}} = \left|\mathbf{p}_{\mathrm{T}}^{\mathrm{miss}}\right| = \left|-\sum_{q} \mathbf{p}_{\mathrm{T}}^{q}\right|. \tag{2}$$

- The squared transverse mass, M_{T}^2, of the isolated lepton ℓ and the $E_{\mathrm{T}}^{\mathrm{miss}}$ system, defined as follows:

$$M_{\mathrm{T}}^2 = 2p_{\mathrm{T}}^{\ell}E_{\mathrm{T}}^{\mathrm{miss}}(1 - \cos\Delta\phi) \tag{3}$$

  with p_{T}^{ℓ} the transverse momentum of the lepton and $\Delta\phi$ the azimuthal separation between the lepton and $\mathbf{p}_{\mathrm{T}}^{\mathrm{miss}}$ vector.
- The azimuthal angle of the $\mathbf{p}_{\mathrm{T}}^{\mathrm{miss}}$ vector, ϕ^{miss}.
- The number of jets entering the S_{T} sum.
- The number of these jets identified as originating from a b quark.
- The isolated-lepton momentum, expressed in polar coordinates (p_{T}, η, ϕ).
- The three isolation quantities (`ChPFIso`, `NeuPFIso`, `GammaPFIso`) for the isolated lepton.
- The lepton charge.
- The *isEle* flag for the isolated lepton.

The list of 801 particles is used to generate two visual representations of the events: *raw representation* and *abstract representation*. In the *raw representation*, the (η, ϕ) plane corresponding to the detector acceptance is divided into a barrel region ($|\eta| < 1.5$), two end-cap regions ($1.5 \leq \eta < 3.0$ and $-3.0 < \eta \leq -1.5$), and two forward regions ($3.0 \leq \eta < 5.0$ and $-5.0 < \eta \leq -3.0$). The barrel and end-cap regions of the electromagnetic calorimeter, as well as the end-cap of the hadronic calorimeter (HCAL), are binned in cells of size 0.0187×0.0187. The barrel region of the HCAL is binned with cells of size 0.087×0.087. The forward regions are binned with cells of size 0.175 in η, while the dimension in ϕ varies from 0.175 to 0.35. Each cell is filled with the scalar sum of the p_{T} of the particles pointing to that cell. The three classes of particles (charged particles, photons, and neutral hadrons) are considered separately, resulting in three channels. An example is shown in Fig. 2 for a $t\bar{t}$ event. This representation corresponds to the raw image recorded by the detector.

Recently, it was proposed to represent LHC collision events as abstract images where reconstructed physics objects (jets, in that case) are represented as geometric shapes whose size reflects the energy of the particle [11]. We generalize this *abstract representation* approach by applying it to the full list of particles. Each particle is represented as a unique geometric shape, centered at the particle's (η, ϕ) coordinates and with size proportional to its $\log p_{\mathrm{T}}$. The geometric shapes are chosen as follow: (1) pentagons for the selected isolated electron or muon; (2) triangles for photons; (3) squares for charged particles; (4) hexagons for neutral hadrons. The images are digitized as arrays of size $5 \times 150 \times 94$, where each of the first four channels contains a separated particle class, and the last channel contains the $E_{\mathrm{T}}^{\mathrm{miss}}$, represented as a circle. As an example, the abstract representation for the event in Fig. 2 is shown in Fig. 3.

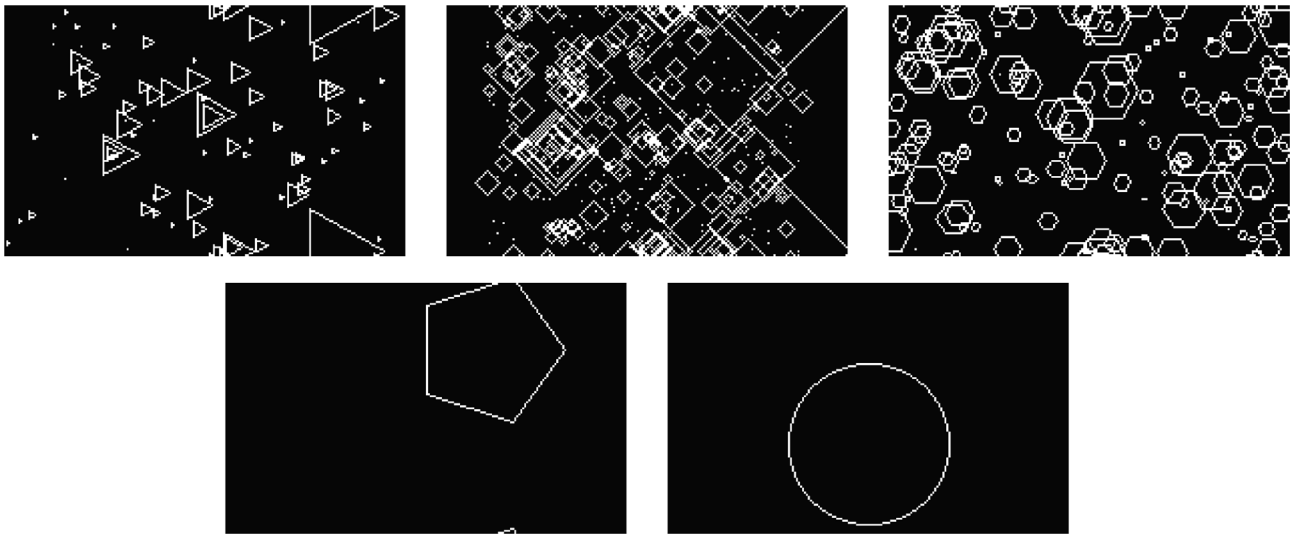

Fig. 3 Example of a $t\bar{t}$ event, represented as a five-channel abstract images of photons (top-left), charged hadrons (top-center), neutral hadrons (top-right), the isolated lepton (bottom-left), and the event $E_{\mathrm{T}}^{\mathrm{miss}}$ (bottom-right)

This abstract representation allows mitigating the sparsity problem of the raw images. On the other hand, there is no guarantee that the physics information is fully retained in this translation. As a result, there could be a reduction of discrimination power. This is one of the points that we aim to investigate in this study.

Model Description

In this section, we describe five types of multi-class classifiers, trained on the four data representations described in the previous section. We start by considering a state-of-the art HEP application, based on the high-level features listed in section "Data Set". We then consider a convolutional neural network taking as input the raw images. This model offers the baseline point of comparison for the classifier using the abstract images. To have a fair comparison between the two approaches, the same kind of network architecture is used for the two sets of images. Next, we consider recurrent neural networks based on LSTMs and GRUs, trained directly on the lists of 801 particles. Finally, we consider a classifier taking both the high-level features and the list of 801 particles as inputs, using a combination of recurrent neural networks and fully connected neural networks.

The CNNs are implemented in `PyTorch` [12]. The recurrent neural networks and feed-forward neural networks are implemented in `Keras` [13] and trained using `Theano` [14] as a back-end. The Adam optimizer [15] is used to adapt the learning rate. The training is capped at 50 epochs, and can be stopped early if there is no improvement in terms of validation loss after 8 epochs. Categorical cross entropy is used as the loss function. All trainings are performed on a cluster of GeForce GTX 1080 GPUs. In an early stage of this work, experiments on the recurrent models were performed on the CSCS Piz Daint super computer, using the `mpi-learn` library [16] for multiple-GPU training.

High-Level-Feature Classifier

A fully connected feed-forward DNN based on a set of high-level features (*HLF classifier*) is the closest approach to the currently used rule-based trigger algorithms. We train a model of this kind taking as input the 14 features contained in the HLF dataset (see section "Data Set"). The 14 features are normalized to take values between 0 and 1.

The final network configuration is the result of an optimization process performed using the `scikit-learn` optimizer [17], which performs an exhaustive cross-validated grid search over a set of hyperparameters related to the network architecture and the training setup. The number of layers, the number of nodes in each layer, and the choice of optimizer have been considered in the scan. For a given number of layers, discrimination performances were found to be constant over the considered range of number of nodes per layer. We believe that this is a direct consequence of the simple problem at hand: even a relatively small networks achieve good classification performances. We then took the smallest network as the best compromise between performance and architecture minimality.

The chosen architecture consists of three hidden layers with 50, 20, and 10 nodes, activated by rectified linear units (ReLU) [18]. The output layer consists of three nodes, activated by a softmax activation function.

Raw-Image Classifier

To classify events represented as raw calorimeter images (*raw-image classifier*), we use DenseNet-121, a model based on the Densely Connected Convolutional Network [19]. The DenseNet-121 architecture includes 4 dense blocks, each of which contains 6, 12, 24, and 16 dense layers, respectively. Each dense layer contains two 2D convolutional layers preceded by batch normalization layers. A dropout rate of 0.5 is applied after each dense layer. Between two subsequent dense blocks is a transition layer consisting of a batch normalization layer, a 2D convolutional layer, and an average pooling layer.

Abstract-Image Classifier

We use the same DenseNet-121 architecture above to classify the abstract-image representation. We refer to this model as *abstract-image classifier*.

Particle-Sequence Classifier

A *particle-sequence classifier* is trained using a recurrent network, taking as input the 801 candidates. To feed these particles into a recurrent network, particles are ordered according to their increasing or decreasing distance from the isolated lepton. Different physics-inspired metrics are considered to quantify the distance (ΔR, $\Delta\phi$, $\Delta\eta$, k_T [10], or anti-k_T [20]). The best results are obtained using the ΔR decreasing distance ordering.

We use gated recurrent units (GRU) to aggregate the input sequence of particle flow candidate features into a fixed size encoding. The fixed encoding is fed into a fully connected layer with three softmax activated nodes. Input data are standardized, so that each feature has zero mean and unit standard deviation. The zero-padded entries in the particle sequence are skipped with the Masking layer. The best internal width of the recurrent layers was found to be 50, determined by k-fold cross validation on a training set of 210,000 events. We also considered using long–short-term memory networks (LSTM) to replace the GRU, but we found that the GRU architecture outperformed the LSTM architecture for the same number of internal cells.

Inclusive Classifier

To inject some domain knowledge in the GRU classifier, we consider a modification of its architecture in which the 14 features of the HLF data set are concatenated to the output of the GRU layer after some dropout (see Fig. 4). As for the other classifiers, the final output layer consists of three nodes, activated by a softmax activation function. We refer to this model as *inclusive classifier*.

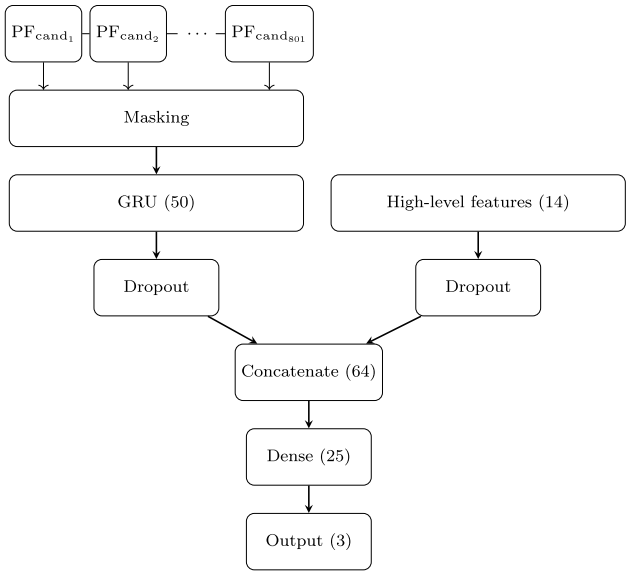

Fig. 4 Network architecture of the inclusive classifier

Results

Each of the models presented in the previous section returns the probability of each event to be associated to a given topology: y_{QCD}, y_W, and $y_{t\bar{t}}$. By applying a threshold requirement on y_W or $y_{t\bar{t}}$, one can define a W or a $t\bar{t}$ classifier, respectively. By changing the threshold value, one can build the corresponding receiver-operating characteristic (ROC) curve. Figure 5 shows the comparison of the ROC curves for five classifiers: the DenseNets based on raw images and abstract images, the GRU using the list of particles, the DNN using the HLFs, and the inclusive classifier using both the HLFs and the list of particles. Results for both a $t\bar{t}$ and W selectors are shown.

Acceptable results are obtained already with the raw-image classifier. On the other hand, the use of abstract images allows us to reach better performances. A further improvement is observed for those models not using an image-based representation of the event. The fact that the HLF selectors perform so well does not come as a surprise, given a considerable amount of physics knowledge implicitly provided by the choice of the relevant features. On the other hand, the fact that the particle-sequence classifier reaches better performances compared to the HLF selector is remarkable, as is the further improvement observed by merging the two approaches in the inclusive classifier. In some sense, the GRU layer is gaining a good part of the physics intuition that motivated the choice of the

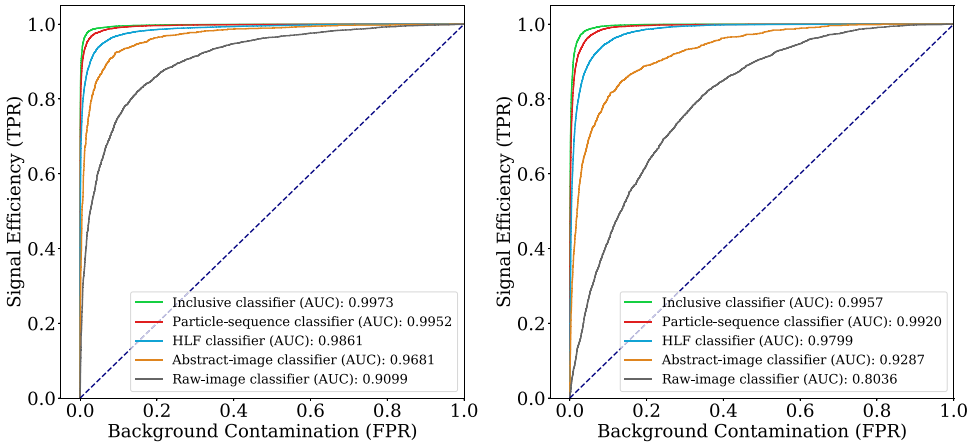

Fig. 5 ROC curves for the $t\bar{t}$ (left) and W (right) selectors described in the paper

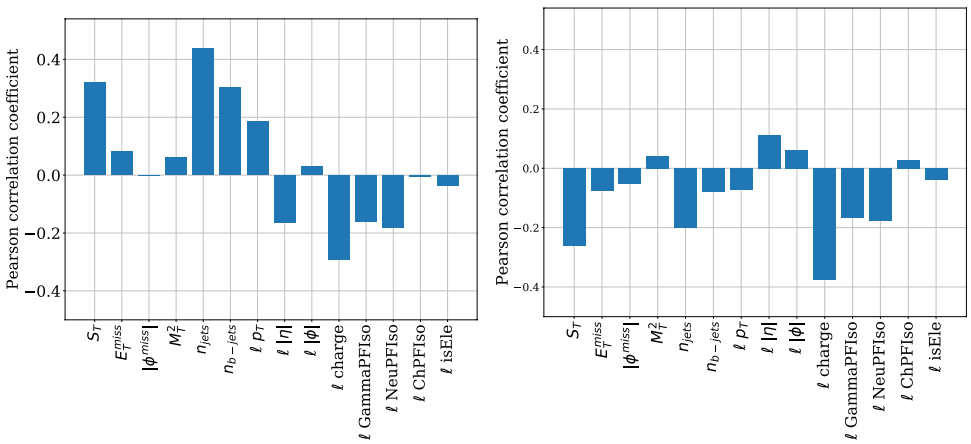

Fig. 6 Pearson correlation coefficients between the $y_{t\bar{t}}$ (left) and y_W (right) scores of the particle-sequence classifier and the 14 quantities of the HLF data set

HLF quantities, but not entirely. Figure 6 shows the Pearson correlation coefficients between the GRU scores ($y_{t\bar{t}}$ and y_W) and the HLF quantities. As one would expect, $y_{t\bar{t}}$ exhibits a stronger correlation with those features that quantify jet activity (n_{jets} in Fig. 6), as well as with the b-jet multiplicity ($n_{\text{b-jets}}$). On the contrary, W events shows an anti-correlation with respect to jet quantities, since the production of associated jets in W events is much more penalized than for $t\bar{t}$ events. As expected, both scores are anti-correlated to the isolation quantities, which takes larger values for non-isolated leptons.

The performance of each of the five classifiers is summarized in Table 1 in terms of false-positive rate (FPR) and trigger rate (TR) as a function of the true-positive rate (TPR). The best QCD rejection is obtained by the inclusive classifier, which can retain 99% of the $t\bar{t}$ or W events with a false-positive rate of $\sim 5.2\%$.

The trigger baseline selection which we use in this study, looser than what is used nowadays in CMS, gives an overall trigger rate (i.e., summing electron and muon events) of ~ 690 Hz, more than a factor two larger than what is currently allocated. Using the 99% working points of the two classifiers, one would reduce the overall rate to ~ 270 Hz (counting the overlap between the two triggers). This would be comparable to what is currently allocated for these triggers, but with a looser selection, i.e., with a less severe bias on the offline analysis. In addition, the trigger efficiency (the TPR) is so high that the bias imposed on offline quantities is quite minimal. This is illustrated in Fig. 7, where the dependence of the TPR on the most relevant HLF quantities is shown. In our experience, any rule-based algorithm with the same target trigger rate would result in larger inefficiencies at small values of at least some of these quantities, e.g., the lepton p_{T}. One should also consider that the principle of a topology classifier could be generalized to other physics cases, as well as to other uses (e.g., labels for fast reprocessing or access to specific subsets of the triggered samples).

Figure 8 shows the TPR and FPR of the inclusive $t\bar{t}$ selector when applying the 99% TPR working-point threshold, as a function of the number of vertices in the event, which quantifies the amount of pileup. The TPR is fairly insensitive to PU until $PU \sim 35$, (the average PU recorded by the LHC in 2018), where the TPR drops to 97%. At the same time, the FPR increases mildly, resulting in a rate increase

Table 1 False-positive rate (FPR) and trigger rate (TR) at different values of the true-positive rate (TPR), for a $t\bar{t}$ (top) and W selector

	Raw-image (DenseNet)	Abstract-image (DenseNet)	HLF (DNN)	Particle-sequence (GRU)	Inclusive (DNN+GRU)
$t\bar{t}$ selector					
FPR @99% TPR	$76.5 \pm 0.2\%$	$50.1 \pm 0.2\%$	$28.6 \pm 0.2\%$	$9.2 \pm 0.1\%$	$5.2 \pm 0.1\%$
FPR @95% TPR	$41.3 \pm 0.2\%$	$15.7 \pm 0.1\%$	$6.1 \pm 0.1\%$	$1.7 \pm 0.1\%$	$0.7 \pm 0.0\%$
FPR @90% TPR	$26.5 \pm 0.2\%$	$7.4 \pm 0.1\%$	$2.7 \pm 0.1\%$	$0.6 \pm 0.0\%$	$0.2 \pm 0.0\%$
TR @99% TPR	382.0 ± 0.9 Hz	250.9 ± 1.0 Hz	143.9 ± 0.9 Hz	48.1 ± 0.6 Hz	28.4 ± 0.4 Hz
TR @95% TPR	207.8 ± 1.0 Hz	80.3 ± 0.7 Hz	32.4 ± 0.5 Hz	11.0 ± 0.3 Hz	6.0 ± 0.2 Hz
TR @90% TPR	134.2 ± 0.9 Hz	39.0 ± 0.5 Hz	15.5 ± 0.3 Hz	5.2 ± 0.2 Hz	3.5 ± 0.1 Hz
W selector					
FPR @99% TPR	$79.0 \pm 0.2\%$	$61.8 \pm 0.2\%$	$23.5 \pm 0.2\%$	$10.2 \pm 0.1\%$	$6.3 \pm 0.1\%$
FPR @95% TPR	$60.5 \pm 0.2\%$	$36.0 \pm 0.2\%$	$9.7 \pm 0.1\%$	$3.7 \pm 0.1\%$	$1.8 \pm 0.1\%$
FPR @90% TPR	$48.1 \pm 0.2\%$	$22.8 \pm 0.2\%$	$5.1 \pm 0.1\%$	$1.8 \pm 0.1\%$	$0.9 \pm 0.0\%$
TR @99% TPR	488.9 ± 0.3 Hz	462.3 ± 0.5 Hz	301.9 ± 0.6 Hz	268.2 ± 0.5 Hz	259.7 ± 0.4 Hz
TR @95% TPR	454.5 ± 0.6 Hz	365.1 ± 0.8 Hz	259.2 ± 0.5 Hz	242.6 ± 0.4 Hz	238.0 ± 0.4 Hz
TR @90% TPR	408.2 ± 0.8 Hz	301.8 ± 0.8 Hz	235.0 ± 0.5 Hz	225.4 ± 0.5 Hz	223.3 ± 0.5 Hz

Rate values are estimated scaling the TPR and process-dependent FPR values by the acceptance and efficiency, assuming a leading-order (LO) production cross section and luminosity of 2×10^{34} cm^{-2} s^{-1}. TR values should be taken only as suggestions of the actual rates, since the accuracy is limited by the use of LO cross sections and a parametric detector simulation

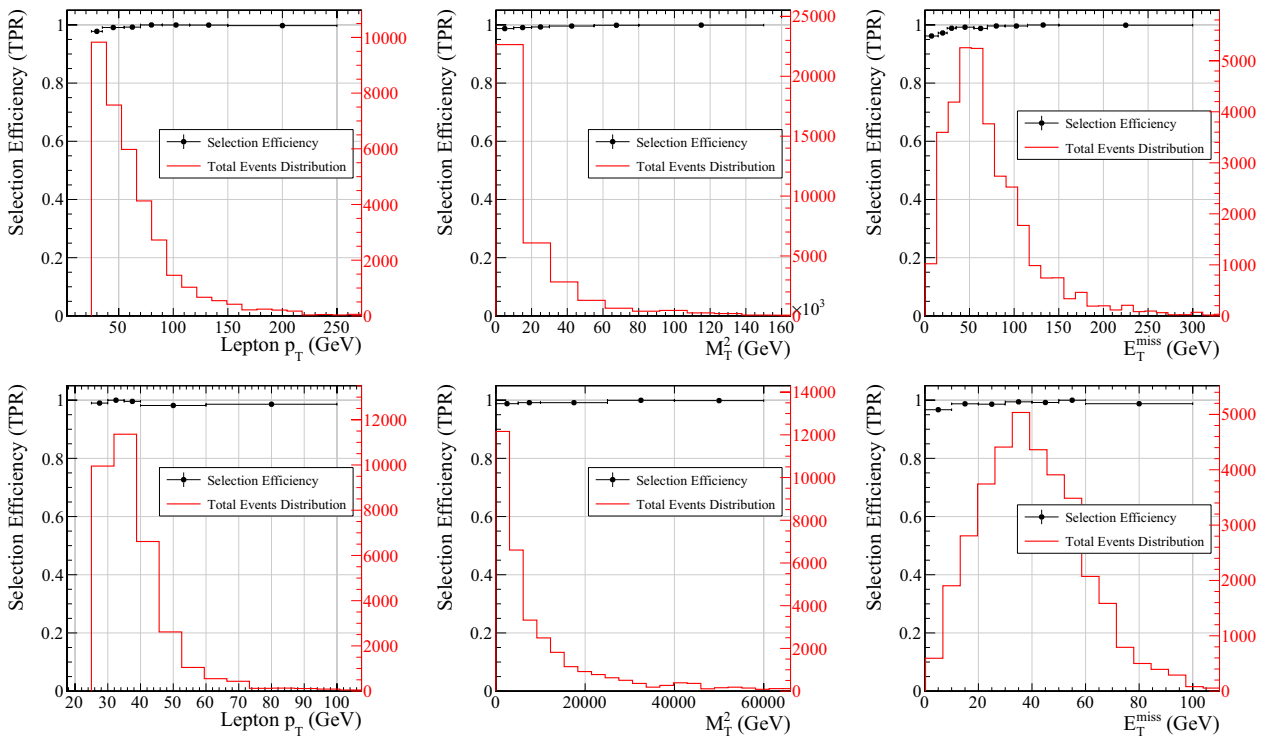

Fig. 7 Selection efficiency using 99% TPR working point as functions of lepton p_T, M_T^2, and E_T^{miss} for the $t\bar{t}$ selector on $t\bar{t}$ events (top) and the W selector on W events (bottom)

from ~ 34 Hz (at the average PU value ~ 20) to ~ 48 Hz at $PU \sim 35$. In other words, the algorithm trained on 2016 conditions would have been sustainable until 2018 with $\sim 15\%$ rate increase (with respect to the average value) or it

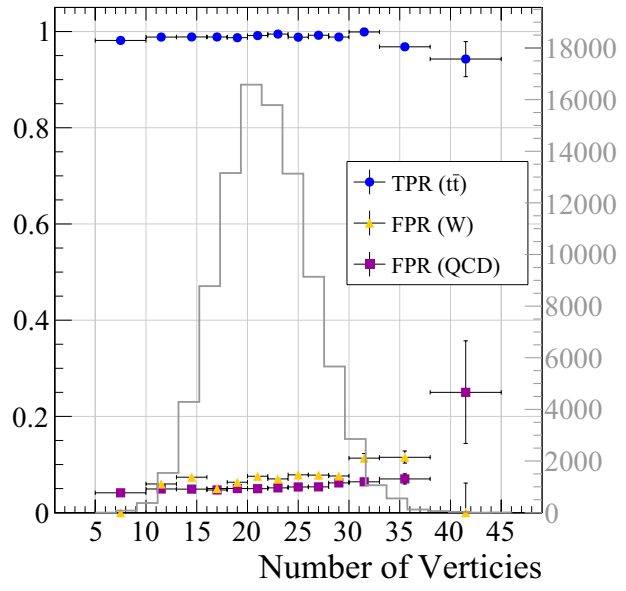

Fig. 8 Dependence of TPR and FPR on the amount of pileup in the event (estimated through the number of vertices) for the inclusive $t\bar{t}$ selector when applying the 99% TPR working-point threshold. The gray histogram shows the distribution of the number of vertices in the training data set, covering a wide range from ∼ 10 to ∼ 40 following a Poisson distribution with mean value of 20

would have required a threshold adjustment along the way, a pretty standard operation when designing a trigger menu at the beginning of the year. We believe that, in view of these facts, the proposed algorithm would be as robust as many state-of-the-art algorithms operated at the LHC experiments.

Impact on Other Topologies

While reducing the resource consumption of standard physics analyses is the main motivation behind this study, it is important to evaluate the impact of the proposed classifiers on other kinds of topologies. For this purpose, we consider a handful of beyond-the-standard-model (BSM) scenarios, and we compute the TPR as a function of the most relevant kinematic quantities, similar to what was done in Fig. 7 for the standard topologies.

We consider the following BSM processes:

- $A \to H^+W$: a heavy Higgs boson A with mass 425 GeV decaying to a charged Higgs boson H^+ of mass 325 GeV and a W^- boson. The H^+ then decays to a W^+H^0 final state, where H^0 is the 125 GeV Higgs boson, which we force to decay to a bottom quark–antiquark pair. This model, introduced in Ref. [21], generates a $2b2W$ topology similar to that given by $t\bar{t}$ events.
- High-mass $A \to H^+W$: a high-mass variation of the previous model, in which the A and H^+ masses are set to 1025 GeV and 625 GeV, respectively.
- $A \to 4\ell$: a light neutral scalar particle A with mass 20 GeV, decaying to two neutral scalars of 5 GeV each, both decaying to muon pairs, for a total of four muons in the final state.
- W' resonance with mass 300 GeV, decaying inclusively with W-like couplings.
- Z' resonance with mass 600 GeV, decaying to a pair of electrons or muons.

These events are filtered with the baseline selection described in section "Data Set".

For each of these models, we consider the inclusive classifier and apply the 99%-TPR thresholds on $y_{t\bar{t}}$ and y_W. We then consider the fraction of events passing at least one of the two selectors. Results are shown in Fig. 9 for the most relevant kinematic quantities. While the individual selectors might show local inefficiencies, the combination of the two trigger paths is perfectly capable of retaining any event with features different from that of a QCD multijet event. In this respect, the logical OR of our two exclusive topology classifiers is robust enough to also select a large spectrum of BSM topologies. On the other hand, one cannot guarantee that QCD-like topologies (e.g., a dark photon produced in jet showers and decaying to lepton pairs) would not be rejected, a limitation which also affects traditional inclusive trigger strategies.

Robustness Study

As the classifier is trained on Monte Carlo simulation samples, one needs to consider the discrepancy between Monte Carlo and real data when deploying the classifier in the trigger. We investigate the robustness of our topology classifiers against this discrepancy by creating a pseudo-data sample, which attempts to emulate real data by adding a Gaussian noise to the particles' momenta in the simulation samples. The Gaussian noise has mean of zero and standard deviation of 10% of the variable's values being applied. Figure 10 shows some comparisons between the Monte Carlo samples and the pseudo-data with this Gaussian noise added.

We evaluate the performance of our fully trained inclusive classifier on the new pseudo-data. Table 2 shows a slight reduction of signal efficiency: at the same background contamination rate of 5.2%, the signal efficiency reduces by only 1.4%. This demonstrates that our classifiers can be robust against some augmentation that mimics the discrepancy between data and Monte Carlo simulation. A comprehensive study on full simulation and data in proper control

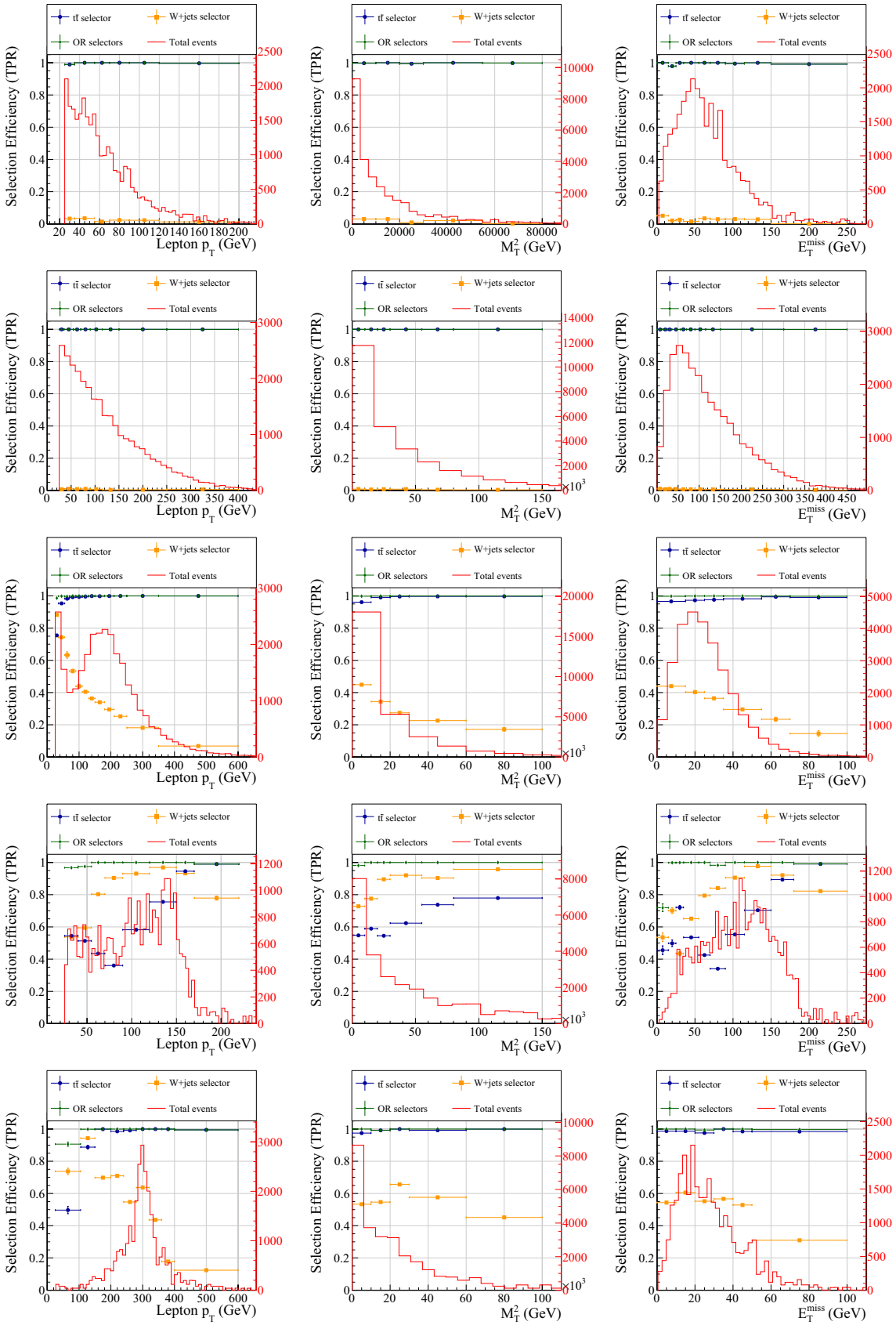

Fig. 9 Selection efficiencies of different BSM models using 99% TPR working point as functions of lepton p_{T}, M_{T}^2, and $E_{\mathrm{T}}^{\mathrm{miss}}$. From top to bottom, $A \to H^+W^-$, high-mass $A \to H^+W^-$, $A \to 4\ell$, W', and Z'

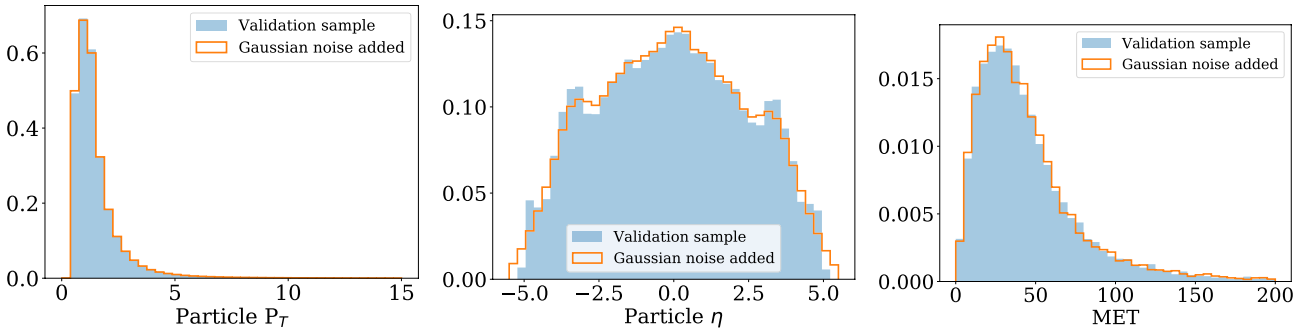

Fig. 10 Distributions of the validation sample and pseudo-data. The pseudo-data are created by adding a Gaussian noise of mean zero and standard deviation of 10% to the validation sample's particle momenta. The high-level features are then recomputed with the new list of particles

Table 2 Signal efficiency (TPR) at different values of the false-positive rate (FPR) for the *inclusive classifier* selecting $t\bar{t}$ evaluated on the validation sample and the pseudo-data

FPR	TPR on validation sample	TPR on pseudo-data
5.2%	$99.0 \pm 0.1\%$	$97.6 \pm 0.1\%$
0.7%	$95.0 \pm 0.1\%$	$90.9 \pm 0.2\%$
0.2%	$90.0 \pm 0.2\%$	$83.5 \pm 0.2\%$

regions would be needed when deploying this classifier into production.

Related Works

Machine learning is traditionally used in high-energy physics as part of data analysis, and was an important ingredient to the discovery of the Higgs boson, as discussed in [22]. Several classification algorithms have been studied in the context of LHC physics application, notably for jet tagging [23–30] and event topology identification [11, 21, 31] using feed-forward neural networks, convolutional neural networks, or physics-inspired architectures. Lists of particles have been used to define jet and event classifiers starting from a list of reconstructed particle momenta [32–34]. These studies typically consider data analysis as the main use case, focusing on small FPR selections. This is the main difference with respect to this study, which focuses on the optimization of real-time data-taking procedure.

In parallel, machine learning techniques have also been used in online event selection. For example, the LHCb experiment used a decision-tree based approach for the high-level trigger in the first LHC run [35] and re-optimized it with MatrixNet algorithm for Run II [36]; ATLAS uses BDT in its multi-step tau trigger for Run II [37]; a BDT was also deployed on FPGA cards of the hardware-level trigger of the CMS experiment [38]. These triggers are mainly based on high-level features related to specific parts of a collision event. We propose instead to define an algorithm that is based on a raw-event representation and considers the full-event collision at once. To our knowledge, this is the first demonstration of how a recurrent neural network could perform a successful inference on a full event and improve topology identification based on object-specific features.

In addition, traditional triggers based on machine learning run in *tagging mode*, i.e., are used to identify certain types of particles. Instead, we propose to use our topology classifier in *veto mode*: the trigger algorithm running downstream would be a classic trigger with loose selection, which would normally be unsustainable due to high throughput. The topology classifier would subsequently remove a majority of background events, sustaining the trigger rate and saving downstream computing resources.

Note. After submitting this paper for review, the study presented in Ref. [39] showed how a topology classification based on full-event information can boost tagging efficiency or purity of a single-object trigger, or both, in the context of an offline analysis.

Conclusions

We show how deep neural networks can be used to train topology classifiers for LHC collision events, which could be used as a clean-up filter to select or reject specific event topologies in a trigger system. We consider several network architectures, applied to different representations of the same collision datasets.

The best results are obtained by combining a set of physics-motivated high-level features with the output of a GRU unit applied to a list of particle-level features. For the most difficult case, i.e., selecting rare $t\bar{t}$ events, we show how a trigger based on this concept would retain 99% of the $t\bar{t}$ events while reducing the FPR by more than ~ 10 times.

The information given as input to the GRU, the abstract-image CNN, and the raw-image CNN is the same, but coded differently. The difference in performance is then a combination of two effects: the encoding of this information in the input event representation and the way the network architecture exploits it. The DNN case is different. The DNN uses in principle less information. On the other hand, the list of HLFs given as input to the DNN is based on domain knowledge that the other networks have to learn by themselves. This is why the DNN model is very competitive despite using less information and why the inclusive classifier (GRU+DNN) improves on the GRU-based particle-sequence classifier. Nevertheless, it is remarkable that the score of the particle-sequence classifier learns interesting correlation patterns with the HLF features, showing that (to some extent) the GRU is learning some of this domain knowledge.

We show that such a trigger would have a minimal impact on the main kinematic features of the event topologies under consideration. The effect of operating this topology classifier as a final filter of a given single-lepton trigger would result in small decrease of trigger efficiency by few percentage (depending on the TPR of the chosen working point). On the other hand, such a filter would allow for a looser selection, efficiently including non-isolated leptons with low p_{T} without downstream consequences in terms of computational power and storage. In addition, the logic `OR` of the $t\bar{t}$ and W selections would also catch a broad class of new-physics topologies, on which the classifiers were not trained.

The advantages of running these types of algorithms come at the cost of computational resources to train the models. In our case, a single training of the *inclusive classifier* took 4 h on a cluster consisting of 6 GeForce GTX 1080 GPUs. Building a cluster of a few tens of GPUs of this kind, to be used as a training facility, is well within the budget of big-experiment computing projects. For this reason, dedicated studies are ongoing to integrate train-on-demand services in the computing infrastructures of LHC experiments [16] [40]. In view of the challenging trigger environment foreseen for the High-Luminosity LHC, it would be important to test this trigger strategy as a way to preserve a good experimental reach with a substantial reduction of computational resources. In this respect, we look forward to the LHC Run III as an opportunity to experiment with this technique using full simulation and study its impacts on real-time event selection.

Acknowledgements This work is supported by Grants from the Swiss National Supercomputing Center (CSCS) under project ID d59, the United States Department of Energy, Office of High Energy Physics Research under Caltech Contract No. DE-SC0011925, and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant agreement no. 772369). T.N. would like to thank Duc Le for valuable discussions during the earlier stage of this project. We thank CERN OpenLab for supporting D.W. during his internship at CERN. We are grateful to Caltech and the Kavli Foundation for their support of undergraduate student research in cross-cutting areas of machine learning and domain sciences. Part of this work was conducted at "*iBanks*", the AI GPU cluster at Caltech. We acknowledge NVIDIA, SuperMicro and the Kavli Foundation for their support of "*iBanks*".

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Appendix A An Alternative Use Case

In this paper, we showed how one could use a topology classifier to keep the overall trigger rate under control while operating triggers with otherwise unsustainable loose selections. In this appendix, we discuss how topology classifiers could be used to save resources for a pre-defined baseline trigger selection by rejecting events associated to unwanted topologies. In this case, the main goal is not to reduce the impact of the online selection. Instead, we focus

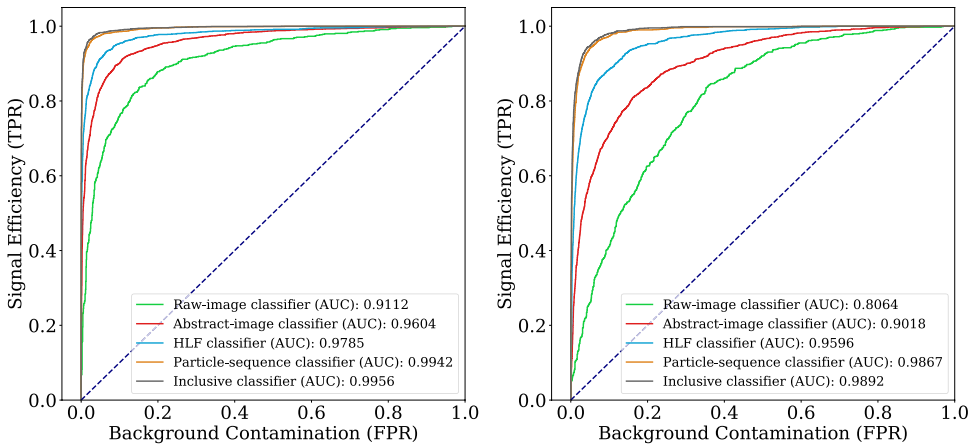

Fig. 11 ROC curves for the $t\bar{t}$ (left) and W (right) selectors described in the paper, trained on a data set defined by a tighter baseline selection

Table 3 False-positive rate (FPR) and trigger rate (TR) corresponding to different values of the true-positive rate (TPR), for a $t\bar{t}$ (top) and W selector

	Raw-image (DenseNet)	Abstract-image(DenseNet)	HLF (DNN)	Particle-sequence (GRU)	Inclusive (DNN+GRU)
$t\bar{t}$ selector					
FPR @99% TPR	76.7 ± 0.2%	55.5 ± 0.3%	44.3 ± 0.3%	13.4 ± 0.2%	10.2 ± 0.2%
FPR @95% TPR	43.5 ± 0.3%	20.2 ± 0.2%	9.1 ± 0.2%	2.1 ± 0.1%	1.5 ± 0.1%
FPR @90% TPR	24.8 ± 0.3%	9.9 ± 0.2%	4.2 ± 0.1%	0.6 ± 0.0%	0.5 ± 0.0%
TR @99% TPR	285.8 ± 0.9 Hz	230.4 ± 1.0 Hz	219.6 ± 1.0 Hz	56.7 ± 0.7 Hz	42.4 ± 0.6 Hz
TR @95% TPR	148.9 ± 1.0 Hz	84.6 ± 0.9 Hz	37.2 ± 0.6 Hz	9.9 ± 0.3 Hz	8.3 ± 0.3 Hz
TR @90% TPR	72.9 ± 0.8 Hz	41.6 ± 0.6 Hz	18.6 ± 0.4 Hz	3.9 ± 0.2 Hz	3.8 ± 0.2 Hz
W selector					
FPR @99% TPR	81.3 ± 0.2%	68.9 ± 0.3%	45.7 ± 0.3%	17.3 ± 0.2%	14.9 ± 0.2%
FPR @95% TPR	58.4 ± 0.3%	43.9 ± 0.3%	19.6 ± 0.2%	6.1 ± 0.1%	5.2 ± 0.1%
FPR @90% TPR	46.9 ± 0.3%	30.2 ± 0.3%	11.7 ± 0.2%	3.0 ± 0.1%	2.5 ± 0.1%
TR @99% TPR	385.9 ± 0.2 Hz	384.3 ± 0.2 Hz	376.3 ± 0.2 Hz	363.1 ± 0.2 Hz	362.8 ± 0.2 Hz
TR @95% TPR	367.5 ± 0.5 Hz	360.8 ± 0.5 Hz	349.7 ± 0.5 Hz	344.2 ± 0.4 Hz	343.9 ± 0.5 Hz
TR @90% TPR	343.6 ± 0.6 Hz	336.6 ± 0.6 Hz	323.8 ± 0.6 Hz	325.0 ± 0.6 Hz	324.7 ± 0.6 Hz

Rate values are estimated scaling the TPR and process-dependent FPR values by the acceptance and efficiency, assuming a leading-order (LO) production cross section and luminosity of 2×10^{34} cm^{-2} s^{-1}. TR values should be taken only as a loose indication of the actual rates, since the accuracy is limited by the use of LO cross sections and a parametric detector simulation

on reducing resource consumption downstream for a given trigger selection.

To this purpose, we consider a copy of the data set described in section "Data Set", obtained tightening the p_{T} threshold from 23 to 25 GeV and the isolation requirement from ISO < 0.45 toISO < 0.20. Doing so, the sample composition changes as follow: 7.5% QCD; 92% W; 0.5% $t\bar{t}$. With such selections, the trigger acceptance rate would decrease from 690 Hz to 390 Hz, closer to what is currently allocated for these triggers in the CMS experiment.

Following the procedure described in sections "Model description" and "Results", we train the same topology classifiers on this data set. The corresponding ROC curves are presented in Fig. 11 for a $t\bar{t}$ and a W selector.

We then define a set of trigger filters applying a lower threshold to the normalized score of the classifier, choosing the threshold value that corresponds to a certain TPR value. The result is presented in Table 3, in terms of the FPR and the trigger rate.

The trigger baseline selection we use in this study, close to what is used nowadays in CMS for muons, gives an overall trigger rate (i.e., summing electron and muon events) of ∼ 390 Hz (i.e., 190 Hz per lepton flavor). If one was willing to take (as an example) half the W events and all the $t\bar{t}$ events, this number could be reduced to ∼ 200 Hz using the inclusive selectors presented in this study (taking into account the partial overlap between the two triggers). A more classic approach would consist in prescaling the isolated-lepton triggers, i.e., randomly accepting half of the events. The effect on W events would be the same, but one would lose half of the $t\bar{t}$ events while still writing 15 times more QCD than $t\bar{t}$ events. In this respect, the strategy we propose would allow a more flexible and cost-effective strategy.

References

1. Aaboud M et al (2017) Performance of the ATLAS trigger system in 2015. Eur Phys J C 77(5):317
2. Adam W et al (2006) The CMS high level trigger. Eur Phys J C 46:605–667
3. LeCun Y et al (1990) Handwritten digit recognition with a back-propagation network. In: Touretzky DS (ed) Advances in neural information processing systems 2. Morgan-Kaufmann, Burlington, pp 396–404
4. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
5. Cho K et al (2014) On the properties of neural machine translation: encoder-decoder approaches. In: Proceedings of the 8th workshop on syntax, semantics and structure in statistical translation (SSST-8)
6. Sjöstrand T et al (2015) An introduction to PYTHIA 8.2. Comput Phys Co 191:159–177
7. de Favereau J et al (2014) DELPHES 3, a modular framework for fast simulation of a generic collider experiment. JHEP 02:057
8. Contardo D et al (2015) Technical proposal for the Phase-II upgrade of the CMS detector. CERN-LHCC-2015-010. LHCC-P-008. CMS-TDR-15-02
9. Cacciari M, Salam GP, Soyez G (2012) FastJet user manual. Eur Phys. J C72:1896
10. Cacciari M, Salam GP, Soyez G (2008) The anti-k_t jet clustering algorithm. JHEP 04:063
11. Madrazo CF et al (2017) Application of a convolutional neural network for image classification to the analysis of collisions in High Energy Physics. Preprint at https://arxiv.org/abs/1708.07034
12. Paszke A et al (2017) Automatic differentiation in PyTorch. NIPS Autodiff Workshop. Preprint at https://openreview.net/pdf?id=BJJsrmfCZ

13. Chollet F et al (2015) Keras. GitHub. https://github.com/fchollet/keras
14. Al-Rfou R et al (2016) Theano: a Python framework for fast computation of mathematical expressions. Preprint at https://arxiv.org/abs/1605.02688
15. Kingma DP, Adam JB (2014) A method for stochastic optimization. In: Proceedings of the 3rd international conference on learning representations (ICLR 2015)
16. Anderson D, Spiropulu M, Vlimant JR (2017) An MPI-based Python framework for distributed training with Keras. Preprint at https://arxiv.org/abs/1712.05878
17. Pedregosa F et al (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
18. Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. Proc ICML 27:807–814
19. Huang G et al (2017) Densely connected convolutional networks. In: Proceedings of the 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Honolulu, HI, pp 2261–2269
20. Catani S et al (1993) Longitudinally invariant K_t clustering algorithms for hadron hadron collisions. Nucl Phys B 406:187–224
21. Baldi P, Sadowski P, Whiteson D (2014) Searching for exotic particles in high-energy physics with deep learning. Nat Commun 5(07):4308
22. Radovic A et al (2018) Machine learning at the energy and intensity frontiers of particle physics. Nature 560(7716):41–48
23. de Oliveira L et al (2016) Jet-images—deep learning edition. JHEP 07:069
24. Guest D et al (2016) Jet flavor classification in high-energy physics with deep neural networks. Phys Rev D 94(11):112002
25. Macaluso S, Shih D (2018) Pulling out all the tops with computer vision and deep learning. JHEP 10:121
26. Datta K, Larkoski AJ (2018) Novel jet observables from machine learning. JHEP 03:086
27. Butter A et al (2018) Deep-learned top tagging with a Lorentz layer. Sci Post Phys 5(3):028
28. Kasieczka G et al (2017) Deep-learning top taggers or the end of QCD? JHEP 05:006
29. Komiske PT, Metodiev EM, Schwartz MD (2017) Deep learning in color: towards automated quark/gluon jet discrimination. JHEP 01:110
30. Schwartzman A et al (2016) Image processing, computer vision, and deep learning: new approaches to the analysis and physics interpretation of LHC events. J Phys Conf Ser 762(1):012035
31. Bhimji W et al (2018) Deep neural networks for physics analysis on low-level whole-detector data at the LHC. J Phys Conf Ser 1085:042034. https://doi.org/10.1088/1742-6596/1085/4/042034
32. Louppe G et al (2018) QCD-aware recursive neural networks for jet physics. J Phys Conf Ser 1085:042034
33. Egan S et al (2019) Long short-term memory (LSTM) networks with jet constituents for boosted top tagging at the LHC. JHEP 1901:057
34. Cheng T (2018) Recursive neural networks in quark/gluon tagging. Comput Softw Big Sci 2(1):3
35. Gligorov VV, Williams M (2013) Efficient, reliable and fast high-level triggering using a bonsai boosted decision tree. J Instrum 8(02):P02013
36. Likhomanenko T et al (2015) LHCb topological trigger reoptimization. J Phys Conf Ser 664(8):082025
37. Beauchemin P-H (2018) Real time data analysis with the ATLAS Trigger at the LHC in Run-2. In: Proceedings of 21st IEEE real time conference (RT2018)
38. Acosta DE et al (2017) Boosted decision trees in the level-1 muon endcap trigger at CMS. J Phys Conf Ser 1085:042042
39. Lin J et al (2018) Boosting $H \to b\bar{b}$ with machine learning. JHEP 10:101
40. Kuznetsov V (2018) Tensorflow as a service (TFaaS). GitHub. https://github.com/vkuznet/TFaaS