Searching for exotic long-lived particle states at the LHC using a deep neural network

R. Bainbridge, O. Buchmüller, V. Čepaitis, M. Komm, A. Tapper
on behalf of the CMS Collaboration

Blackett Laboratory, Department of Physics
Imperial College London London, SW7 2AZ, UK
r.bainbridge@imperial.ac.uk

Abstract

We present a tagging algorithm to identify displaced jets arising from the decays of new long-lived particle (LLP) states in data recorded by the CMS detector at the CERN LHC. The tagger is a multiclass classifier based on a deep neural network. Information from individual particles and secondary vertices within jets are refined through the use of convolutional networks, before being combined with high-level engineered variables via a dense network. The LLP decay length, $c\tau_0$, is used as an external parameter to the neural network, which allows for hypothesis testing over several orders of magnitude in $c\tau_0$. We define a method based on truth information from Monte Carlo simulation to reliably label jets originating from an LLP decay for supervised training. The training is performed by streaming ROOT trees containing O(100 M) jets directly into the TensorFlow queue and threading system. This custom workflow allows a flexible selection of input features and the asynchronous preprocessing of data, such as the resampling and shuffling of batches on the CPU, in parallel to training on the GPU. Domain adaptation is performed with control samples of pp collision data to ensure good agreement between data and Monte Carlo simulation. The tagger provides a rejection factor of 10 000 for jets from standard model processes while maintaining an LLP jet tagging efficiency of 30–80% for LLPs with $1\,\mathrm{mm} \leq c\tau_0 \leq 10\,\mathrm{m}$. We describe the novel application of several machine learning techniques to LLP searches.

1 Introduction

Machine-learned algorithms are routinely deployed to perform event reconstruction, particle identification, event classification, and other tasks [1] when analysing data samples of proton-proton (pp) collisions recorded by experiments at the CERN LHC. A jet is a collimated spray of a few tens of final-state particles that originate from the hadronization of a quark or gluon. Jets are copiously produced at the LHC. Machine learning techniques have been widely adopted to classify jets according to the underlying flavour of the original parton [2]. Heavy b hadrons with lifetimes of $\mathcal{O}(10^{-12}\,\mathrm{s})$ can typically travel distances of approximately 1–10 mm, depending on their momenta, before decaying. Several algorithms have been developed to identify jets containing b hadrons (b jets) [3, 4, 5, 6].

Various extensions to the standard model (SM) [7, 8, 9] predict the existence of long-lived particles (LLPs) with a proper lifetime τ_0 that can be very different from those of known SM particle states. Consequently, the production and decay of LLPs at the LHC could give rise to atypical experimental signatures. A comprehensive review of current LHC searches for LLPs can be found in Ref. [10]. In this paper, a novel LLP jet tagger, inspired by the DeepJet algorithm [5, 6], is presented and its performance is benchmarked using simplified models of split supersymmetry (SUSY) [7] that yield a

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

long-lived gluino. The gluino decays to a weakly interacting and massive neutralino, which is a dark matter candidate, as well as jets that can be significantly displaced from the luminous region of the proton beams. The decay length $c\tau_0$ of the gluino is a free parameter of the split SUSY model.

A detailed description of the CMS detector can be found in Ref. [11]. The central feature of the CMS apparatus is a superconducting solenoid of $6\,\mathrm{m}$ internal diameter, providing a magnetic field of $3.8\,\mathrm{T}$. Within the solenoid volume is a silicon pixel and strip tracker. Calorimeter systems are also located within the solenoid and beyond. Muons are detected in gas-ionization chambers outside the solenoid. Events of interest are selected using a two-tiered trigger system [12]. The particle-flow (PF) algorithm [13] aims to reconstruct and identify each individual particle in an event, with an optimized combination of information from the various elements of the CMS detector. Jets are clustered from the PF particle candidates using the anti-k_T algorithm [14].

2 The LLP jet algorithm

The training of the deep neural network (DNN) is performed with simulated events produced with various Monte Carlo generator programs. Reliable labelling schemes are typically based on truth information from Monte Carlo generators. A standard procedure known as "ghost" labelling [15] determines the jet flavour by clustering not only the reconstructed final-state particles into jets, but also the generator-level particles. The ghost labelling scheme is adopted by the LLP jet tagger for jets from SM background processes. For the LLP jets, the presence of strong interactions from QCD between the quark-antiquark pairs produced in the gluino decay prevents a reliable and unique association between each reconstructed jet and its originating quark. An alternative scheme labels a jet as originating from an LLP if the majority of its momentum stems from a vertex formed by the daughter particles from a gluino decay.

The DNN architecture, shown in Figure 1, is based on the DEEPJET tagger [5, 6]. Up to $\approx$700 input features are used, comprising the kinematical properties of: up to 25 charged and 25 neutral PF particle candidates, ordered by impact parameter significance or transverse momentum, respectively; up to four secondary vertices; and 14 global features associated with the jet. One-dimensional convolutional layers are applied in four sequential layers, each with up to 64 nodes depending on the type of input features, to extract the most useful latent features. After each layer, a LeakyReLU activation function is used [16]. Dropout layers are interleaved throughout the DNN with a 10% dropout rate [17]. After the final convolutional layer, comprising 4 or 8 nodes, the compressed feature vectors are flattened and concatenated with the global jet features. The resulting feature vector is fed into a series of dense layers for predicting the jet label using softmax as activation for the last layer and categorical cross entropy as the loss function. The multiclass classifier predicts labels for LLP jets and four SM background classes: b and c hadrons, light-flavour (uds) quarks, and gluons (g).

The $c\tau_0$ of the LLP is introduced as an external parameter at the dense network stage. The experimental signature for a displaced jet depends strongly on $c\tau_0$: the DNN is able to exploit information from all CMS detector systems if the decay occurs promptly, in the vicinity of the luminous region of the beams, while information can be limited if the decay occurs in the outermost detector systems. The parameterised approach allows for hypothesis testing with a single DNN. Values of $c\tau_0$ that span six orders of magnitude are used.

Domain adaptation (DA) by backpropagation [19] is employed to obtain a similar jet classification performance when applied to jets in control samples of pp collision data or simulated events. This ensures the DNN is insensitive to differences in the input feature distributions for the two domains, which may arise due to limitations in the simulation. To achieve this, the DNN is extended to predict the jet domain (simulation or data). This is done by adding a branch after the first dense layer, the feature layer. At the end of the domain prediction branch, the sigmoid activation function is used while the loss function is binary cross entropy. A gradient reversal layer is inserted in the domain branch directly after the feature layer. This special layer is only active during backpropagation and reverses the gradients of the domain loss with respect to the weights in preceding layers. During the DNN training, the combined loss is minimized.

Supervised training of the DNN is performed to predict the jet class. The ADAM optimizer [20] is used to minimize the loss function with respect to the parameters. The DNN training relies on simulated events of split SUSY models and various SM background processes. Approximately 20 million jets, are a few tens of epochs, are used to train the DNN. For the domain prediction,

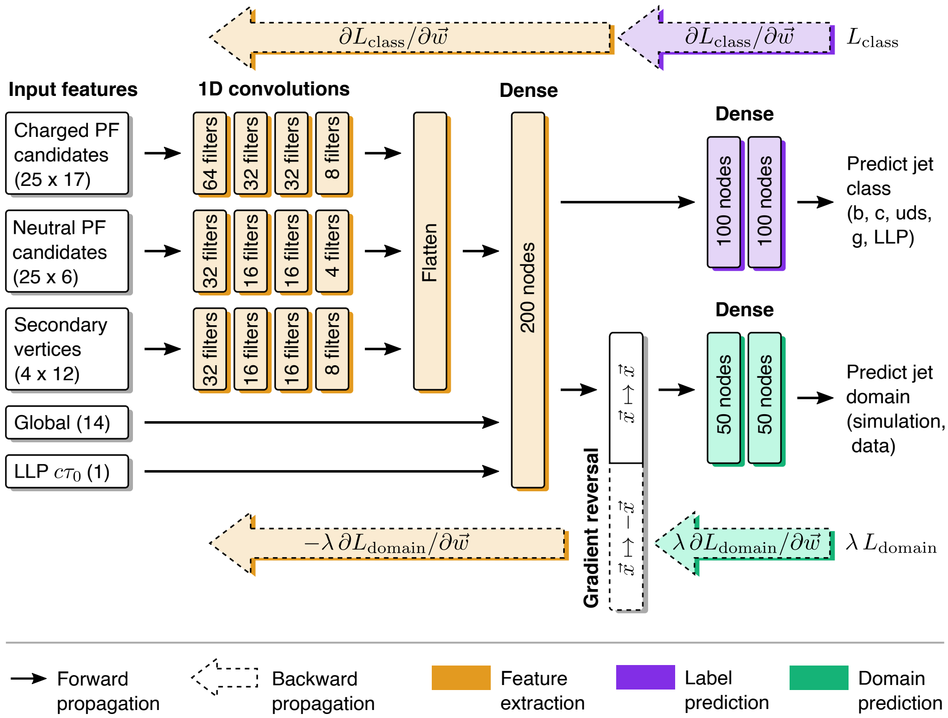

Figure 1: An overview of the DNN architecture, which comprises convolutional and dense layers; the number of filters and nodes, respectively, is indicated. Dropout layers and activation functions are not shown. The input features are grouped by object type and $(m \times n)$ indicates the maximum number of objects (m) and the number of features per object (n). The gradients of the class (L_{class}) and domain (L_{domain}) losses with respect to the weights $\vec{w}$, used during backpropagation, are shown. Figure taken from Ref. [18].

1.2 million jets from control samples of pp collision data, and simulated samples of the expected SM processes, are used. The learning rate is scheduled to decay, with an initial learning rate of 0.01.

The TENSORFLOW v1.6 [21] queue system is used to read and preprocess files for the DNN training. The KERAS v2.1.5 [22] software package is used to implement the DNN architecture. At the beginning of each epoch, a queue holding a randomized list of the input files is initialized. Files names are dequeued asynchronously in multiple threads. For each thread, ROOT v6.18.00 [23] trees contained in the files are read from disk to memory in batches using a TENSORFLOW operation kernel, developed in the context of this paper. The resulting batches are resampled to achieve the same distributions in p_{T} and η for all jet classes and are enqueued asynchronously into a second queue, which caches a list of tensors. The DNN training commences by dequeuing a randomized batch of tensors and generating $c\tau_0$ values for all SM jets within the batch. The advantages of this system lay in its flexibility to adapt to new input features or samples on-the-fly. The (p_{T}, η) resampling and the generation of $c\tau_0$ values for the SM jets proceeds asynchronously in multiple threads, managed by TENSORFLOW, on the CPU while the network is being trained.

3 Performance and summary

The application of DA significantly improves the agreement between the distributions of LLP jet class probability, $P(\text{LLP}|c\tau_0)$, obtained from simulation and pp collision data. The maximum value of $P(\text{LLP}|c\tau_0)$ obtained from all selected jets in a given event is shown in Figure 2. This approach allows a more robust treatment of related experimental systematic uncertainties in estimates of SM backgrounds for a new-physics search.

The performance of the LLP jet tagger is studied with simulated events for three different benchmark SUSY models with $c\tau_0 = 1\,\text{mm}$, as shown in Figure 3 (left). Given a mistag rate of 0.01% for

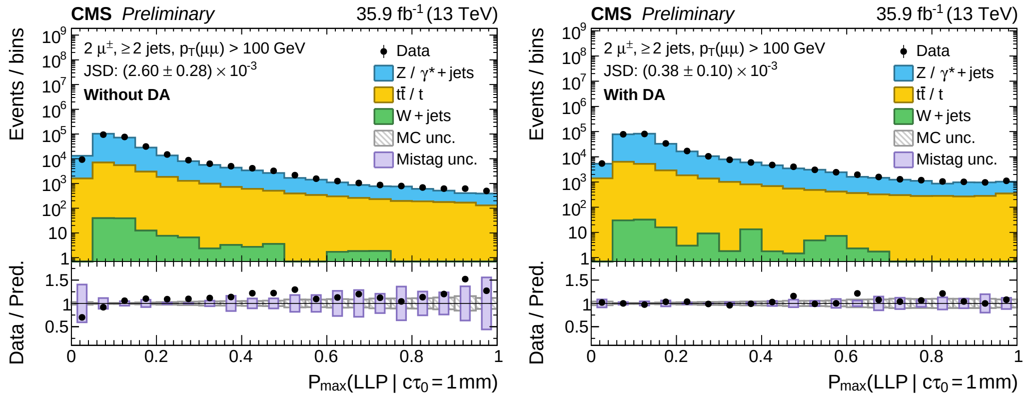

Figure 2: Distributions of the maximum probability for the LLP jet class obtained from all selected jets in a given event, $P_{\mathrm{max}}(\mathrm{LLP}|c\tau_0 = 1\,\mathrm{mm})$. The distributions from pp collision data (circular marker) and simulation (histograms) are compared for a sample of $\mu\mu$+jets events, using a DNN trained without (left) and with (right) DA. The Jensen–Shannon divergence (JSD) [24] is reduced by an order of magnitude following the application of DA. Figure taken from Ref. [18].

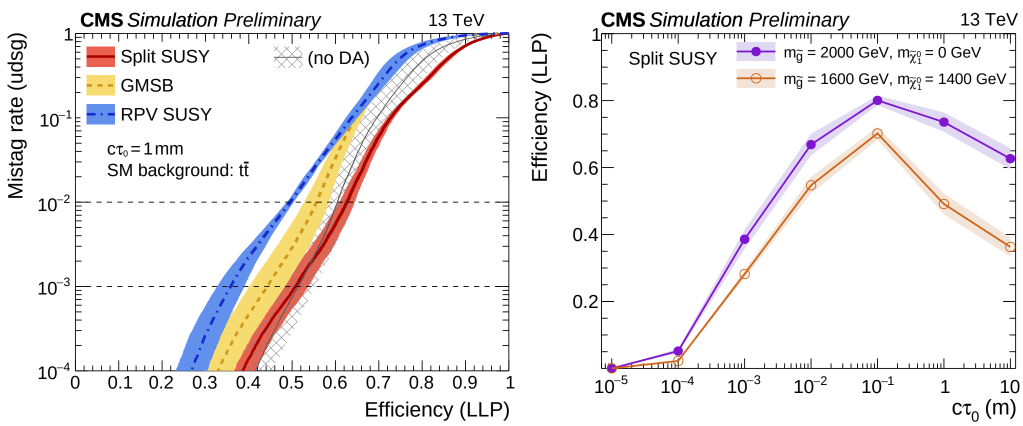

Figure 3: (Left) ROC curves illustrating the tagger performance for a split (solid line), GMSB [8] (dashed), and RPV [25] (dot-dashed) SUSY benchmark model. The thick and thin solid curves indicate the performance using the DNN trained with and without DA, respectively. (Right) The LLP jet tagging efficiency, using a working point that yields a mistag rate of 0.01% for the light-flavour jet class, as a function of the model parameter value $c\tau_0$ for split SUSY models characterized by a large and small difference in gluino ($\tilde{g}$) and neutralino ($\tilde{\chi}^0_1$) mass, as indicated in the legend. Figure taken from Ref. [18].

light-flavour jets, equivalent to a background rejection factor of 10 000, efficiencies of 25–40% are achieved for LLP jets. Figure 3 also demonstrates that training the DNN with pp collision data does not significantly degrade the tagger classification performance.

The LLP jet tagging efficiency as a function of $c\tau_0$, obtained with a working point that yields a mistag rate of 0.01% for the light-flavour jet class, is shown in Figure 3 (right) for two split SUSY scenarios characterized by different mass spectra. Further studies demonstrate that the parameterization according to $c\tau_0$ does not significantly impact the tagger performance with respect to the training of multiple DNNs, one per $c\tau_0$ value.

Finally, the potential performance of the tagger is demonstrated through its application in a search for split SUSY in final states containing jets and significant transverse missing momentum. The expected performance offers excellent prospects for the discovery of new physics at the "lifetime frontier" of collider-based experiments. Further details can be found in Ref. [18].

References

[1] Kim Albertsson et al. Machine Learning in High Energy Physics Community White Paper. *J. Phys. Conf. Ser.*, 1085:022008, 2018.

[2] Andrew J. Larkoski, Ian Moult, and Benjamin Nachman. Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning. 2017. arXiv:1709.04464.

[3] Georges Aad et al. ATLAS b-jet identification performance and efficiency measurement with $\mathrm{t\bar{t}}$ events in pp collisions at $\sqrt{s} = 13$ TeV. 2019. Submitted to *Eur. Phys. J. C*.

[4] A. M. Sirunyan et al. Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV. *JINST*, 13:P05011, 2018.

[5] Markus Stoye. Deep learning in jet reconstruction at CMS. *Journal of Physics: Conference Series*, 1085:042029, 2018.

[6] Performance of the DeepJet b tagging algorithm using 41.9/fb of data from proton-proton collisions at 13TeV with Phase 1 CMS detector. CMS Detector Performance Note CMS-DP-2018-058, 2018.

[7] G. F. Giudice and A. Romanino. Split supersymmetry. *Nucl. Phys. B*, 699:65–89, 2004. [Erratum: Nucl. Phys. B 706 (2005) 487].

[8] G. F. Giudice and R. Rattazzi. Theories with gauge mediated supersymmetry breaking. *Phys. Rept.*, 322:419–499, 1999.

[9] Matthew J. Strassler and Kathryn M. Zurek. Discovering the Higgs through highly-displaced vertices. *Phys. Lett. B*, 661:263–267, 2008.

[10] Juliette Alimena et al. Searching for Long-Lived Particles beyond the Standard Model at the Large Hadron Collider. 2019. arXiv:1903.04497.

[11] S. Chatrchyan et al. The CMS experiment at the CERN LHC. *JINST*, 3:S08004, 2008.

[12] Vardan Khachatryan et al. The CMS trigger system. *JINST*, 12:P01020, 2017.

[13] A. M. Sirunyan et al. Particle-flow reconstruction and global event description with the CMS detector. *JINST*, 12:P10003, 2017.

[14] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. The anti-k_t jet clustering algorithm. *JHEP*, 04:063, 2008.

[15] Matteo Cacciari and Gavin P. Salam. Pileup subtraction using jet areas. *Phys. Lett. B*, 659:119–126, 2008.

[16] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Fourteenth International Conference on Artificial Intelligence and Statistics, Proceedings, Fort Lauderdale, USA , April 11–13, 2011*, volume 15, pages 315–323, 2011.

[17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15:1929–1958, 2014.

[18] CMS Collaboration. A deep neural network-based tagger to search for new long-lived particle states decaying to jets. CMS Physics Analysis Summary CMS-PAS-EXO-19-011, 2019.

[19] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML, pages 1180–1189. JMLR.org, 2015.

[20] Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. arXiv:1412.6980, 2014.

[21] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems. arViv:1603.04467, 2015.

[22] François Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[23] R. Brun and F. Rademakers. ROOT: An object oriented data analysis framework. *Nucl. Instrum. Meth. A*, 389:81–86, 1997.

[24] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theor.*, 37:145, 2006.

[25] R. Barbier et al. R-parity violating supersymmetry. *Phys. Rept.*, 420:1–202, 2005.