Distinguishing 4-top events from background to challenge the Standard Model

Joeran Bosma*
Radboud University Nijmegen, The Netherlands
(Dated: March 30, 2020)

Discriminating signal from background has been an important task for many years. Advances in computational performance have made new methods feasible for this task, for example, ensembles of automatically grown Decision Trees[1] and deep neural networks[2]. Using these classifiers for particle processes can give estimates of the decay rates of the underlying processes. Here, we discriminate between events involving 4 top quarks and events with a top and antitop quark, based on measurements as observed in the Large Hadron Collider. The best performing classifier achieved an accuracy of $95.41\% \pm 0.02\%$, compared to the baseline performance of 83.3% for always predicting a background event. To properly challenge the Standard Model, future research will have to improve this further.

I. INTRODUCTION

The Standard Model theory was used to generate decay processes for events involving either four top quarks (a "4-top event") or a top and antitop quark ("$t\bar{t}$ event"). The quarks in these processes decay to a multitude of products, of which photons, electrons, positrons, jets, b-jets, muons and anti-muons can be measured in the Large Hadron Collider. The number of measured decay products varies between trails, ranging from zero to eighteen.

For each object, the 4-momentum is available and represented in terms of the energy, transverse momentum, azimuth and pseudorapidity (measure for angle relative to the beam axis). This gives five features for each object, counting the type of object as one[1]. The "contribution" of undetected objects (such as neutrinos) are bundled and the resulting "missing" transverse momentum is included in the dataset, as this can also be calculated for actual experiments. This transverse momentum is represented by its magnitude and azimuthal angle. As a result, $2 + n \cdot 5$ features are available for discriminating signal from background, where n is the number of objects.

The physical processes underlying the creation of the measured objects exhibit multiple symmetries. Physical processes are invariant under these symmetries, so the predictions of the classifiers should also be invariant under these symmetries. Rotational symmetry about the beam axis and inversion symmetry in the collision centre are investigated. To attain invariant predictions, training samples can be augmented with random permutations following these symmetries.

* Correspondence email address: j.bosma@student.ru.nl
[1] For deep neural networks this object type is converted to one-hot encoding, giving seven features.

II. METHODS

Multiple strategies are employed to discriminate the 4-top signal events from the $t\bar{t}$ background events. As evaluation metric the accuracy is chosen, measuring the fraction of true predictions. The dataset contains 83.3% background events and 16.7% signal events. This class imbalance is also incorporated in the accuracy, with the signal samples contributing 16.7% and background samples contributing 83.3%.

A. Data Preprocessing

The available data contains a variable number of features, depending on the number of observed objects. At most eighteen objects were measured for a single event. The events with fewer measured objects were zero-padded to fill all $2 + 18 \cdot 5 = 92$ features. This zero padding is necessary for classifiers which expect a fixed number of input features. For the convolutional neural network, the features are transformed into images, which can be done for an arbitrary number of objects, as long as the resolution of the images is kept constant.

Features involving the magnitude of the 4-momentum, i.e. the missing transverse energy (MET), energy of the individual objects and magnitude of the transverse momenta, exhibit long tails in their distribution of values. These distributions can be squashed by taking the logarithm, before feeding the quantities into the classifier. An overview of which transformation is fed to each classifier is given in Appendix A. In many cases, both the original (plain) and transformed (log) features are used jointly.

The dataset has been translated into images in multiple ways. In each configuration, the pseudorapidity (η) and azimuthal angle (ϕ) are rasterized to denote the x and y coordinate of the image, respectively. In total,

five pipelines were constructed to translate an increasing number of features into images. The final pipeline translates all features into images, by creating seven channels for each object type for the energy of the individual objects, and seven channels for the transverse momenta. The MET is set as an additional channel at the respective azimuthal coordinate, with the x coordinate set at the centre of the image. A full overview of the pipelines is provided in Appendix B.

B. Data Augmentation

For the particle processes, rotation symmetry about the beam axis and inversion symmetry about the collision centre seem to be important.

With the conversion of features to images as described in A. Data Preprocessing, a rotation of the whole system about the beam axis corresponds to a translation of the images in y-direction. The translations are done cyclically to account for the cyclical nature of the azimuthal angle. The relative angles are preserved by this operation.

Inversion of the whole system in the collision centre corresponds to horizontally mirroring the produced images (when combined with the rotational symmetry). The channel containing the event-level missing transverse momentum is not flipped, as the pseudorapidity (x coordinate) is not available for this quantity.

C. Conventional Machine Learning Methods

Decision Trees have proven useful in classification tasks within particle physics, especially in ensemble methods[1, 3]. Decision Trees allow for fast inference when working with large datasets. This is in contrast to, for example, a k-Nearest Neighbours classifier, which essentially saves the complete dataset and calculates the distance with all training samples during inference. This difference in computational performance becomes particularly important for ensembles, which can contain hundreds of classifiers.

To assess the performance of the ensemble classifiers, 5-fold cross-validation has been employed. This gives 80.000 training samples for each fold and 20.000 validation samples. Each ensemble consists of Decision Trees (DTs), where the maximal depth of a Tiny, Small, Medium and Large DT is 2, 4, 8 and unlimited, respectively. For all of the ensembles, 100 Decision Trees were grown.

The following classifiers are tested:

- AdaBoost ensemble of Tiny DTs,
- Bagged ensemble of Small DTs with 50% of features and 50% of data,
- Bagged ensemble of Medium DTs with 50% of features and 50% of data,
- Bagged ensemble of Small DTs with 75% of features and 75% of data,
- Bagged ensemble of Medium DTs with 75% of features and 75% of data,
- Random Forest of Large DTs, Entropy criterion,
- Random Forest of Large DTs, Gini criterion.

D. Deep Neural Networks

Conventional classifiers like Decision Trees have some limitations, for example, straight decision boundaries and incompatibility with data augmentation. Ensembles, such as a Random Forest, can overcome some of these limitations, but deep neural networks further alleviate these restrictions.

The deep neural networks are trained on all available features, for which different normalisation methods and transformations are investigated. A baseline model is created which can overfit on the training data, to ensure enough capacity to perform well. The subsequent tests are based on exploratory analysis with single-fold validation, and inspiration from paper [4]. The tests are performed with 5-fold cross-validation.

Each test incorporates the best configuration of the preceding tests. Investigated are:

- Rotational symmetry of azimuthal angles,
- Inversion symmetry of pseudorapidities,
- Dropout, width and depth of architecture,
- Feature transformation to tackle long tails,
- Activation functions,
- Optimisers,
- Batch sizes.

E. Convolutional Neural Networks

Convolutional neural networks are especially competent in extracting spatial relations from its input. The images for these networks are generated as described in A. Data Preprocessing and Appendix B. The azimuthal angles and pseudorapidities are augmented during training as described in B. Data Augmentation.

The convolutional neural network can either be used standalone or in conjunction with, for example, a densely connected neural network. In the latter

case, the features extracted from the images are combined with the features resulting from a (deep) neural network, creating a hybrid model. With the current pipelines, the coupling between features in both branches in broken, possibly resulting in worse performance compared to the individual networks.

The feature extraction performance of the convolutional neural network is investigated by starting with the position of the objects (azimuth and pseudorapidity), and incrementally adding:

- Data augmentation of azimuthal angles,
- Object type,
- Energy of each object,
- Missing transverse energy (including azimuth),
- Transverse momentum of each object,
- Dropout,
- Data augmentation of pseudorapidity.

F. Ensemble of models

An ensemble of convolutional neural networks is created by combining the models trained on the different cross-validation folds. To assess the performance of the ensemble, a sixth of the data is set apart which is not seen by any of the ensemble models. This results in 16.667 test samples, 16.667 validation samples and 66.666 training samples. Evaluating the ensemble performance for each of the test folds, gives in the 'test' accuracy.

Performance of the ensemble is tested for the following voting schemes:

- Majority voting (hard voting),
- Mean voting (soft voting),
- Geometric mean voting (soft voting).

III. RESULTS

Of the conventional machine learning ensemble methods described in C. Conventional Machine Learning Methods, the Random Forest classifier with entropy splitting criterion performed best, with a validation accuracy of $93.90\% \pm 0.13\%$. The relative performance of the other classifiers is shown in Figure 1, with the aforementioned Random Forest classifier as baseline. All shown percentages are percentage points.

Increasing the number of Decision Trees in the Random Forest classifier from 100 to 1000, yields a validation accuracy gain of 0.02%, while decreasing to 30 trees reduces the validation accuracy by 0.26%. This indicates that increasing the number of estimators even further will not result in much performance increase.

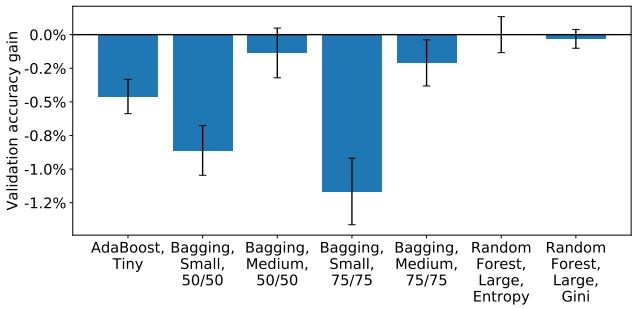

Figure 1. Performance of ensemble classifiers based on DTs, relative to the best Random Forest validation accuracy of $93.90\% \pm 0.13\%$. The classifiers employ four Decision Tree sizes, Tiny, Small, Medium and Large. For these trees the maximal tree depth is 2, 4, 8 and unlimited, respectively. The notions '50/50' and '75/75' specify the percentage of samples and features for each Decision Tree of the Bagged ensemble classifier. Lastly, 'Entropy' and 'Gini' denote the decision criterion for splits in the Random Forest classifiers.

The hyperparameters for the deep neural networks were optimised by performing sweeps and incorporating the best configuration after each sweep. For the best performing options, see Figure 2 and Figure 3. The baseline model, which is the starting point for the first sweep, is chosen to be straightforward and competent enough to at least learn the training data.

The baseline model overfits severely, reaching a training accuracy of $99.67\% \pm 0.05\%$ and validation accuracy of $91.42\% \pm 0.24\%$ after 100 epochs. However, the maximal validation accuracy of this model is $93.50\% \pm 0.09\%$, obtained with early stopping.

Data augmentation of the azimuthal angles and pseudorapidities improves the generalisation of the model significantly.

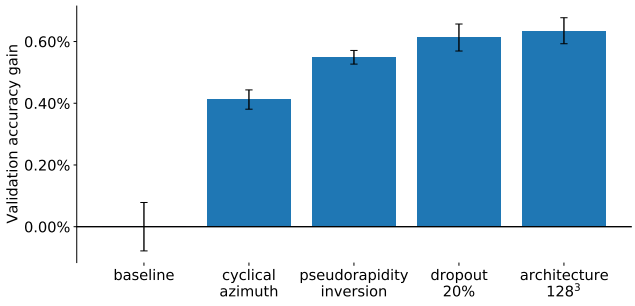

Figure 2. Summary of best performing hyperparameters for the deep neural network across multiple trails. Each trail sweeps across multiple hyperparameters, of which the top performing configuration is shown here. The best performing configuration is incorporated before proceeding to the next trail. For a complete overview, see Appendix C.

Introducing dropout and reducing the number of layers from five to three further improves generalisation. This gives the model shown rightmost in Figure 2 and leftmost in Figure 3, which achieves a maximal validation accuracy of $94.12\% \pm 0.02\%$, an increase of 0.62% from the baseline.

Providing both the original and log-transformed features to the model, switching to LeakyReLU activation functions with negative slope of 0.3, and increasing the batch size to 256 further improves the maximal validation accuracy. These changes give a maximal validation accuracy of $94.26\% \pm 0.02\%$ across folds, an improvement of 0.14%. The Adam optimiser performed best and was retained.

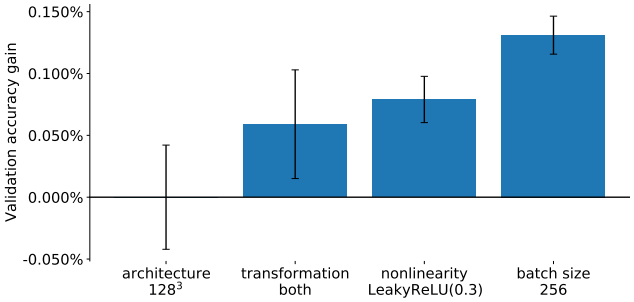

Figure 3. Overview of best performing hyperparameters for the deep neural network across several sweeps. For all results, see Appendix C.

The convolutional neural network was optimised in a similar manner. Providing the locations of all objects (normalised azimuth and pseudorapidity) gave a maximal validation accuracy of 90.79% for the baseline. Cyclical augmentation of the azimuthal angles improved validation accuracy by 0.32% while reducing training accuracy from 99.29% to 91.71%, reducing the generalisation error significantly.

Incrementally including more features improved the maximal validation accuracy to 94.79%. Adding dropout reduces the generalisation error, without impeding validation accuracy. Implementing inversion symmetry of the pseudorapidities improves the maximal validation accuracy of the model by an additional 0.25%. This gives a maximal validation accuracy of $95.08\% \pm 0.04\%$ across folds.

Training six ensembles of five models, as described in F. Ensemble of models, gave a 'test' accuracy of $95.41\% \pm 0.02\%$. For these ensemble predictions, the mean with a threshold of 0.5 was used.

See Figure 4 for a visual depiction of the results.

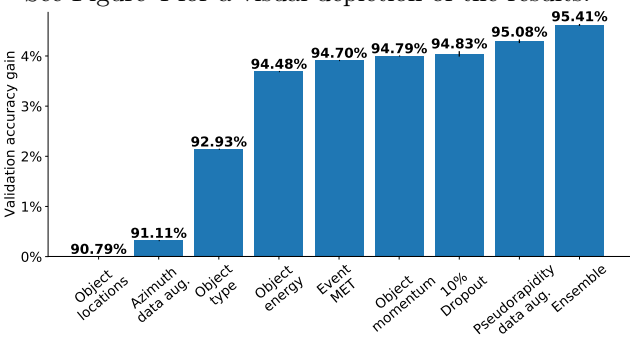

Figure 4. Performance of a convolutional neural network with incrementally more features and data augmentation techniques. Additionally, one test adds dropout and the final step combines multiple models into an ensemble. The bars show the maximal validation accuracy, except for the ensemble performance, which shows the 'test' accuracy.

IV. CONCLUSIONS

Conventional machine learning classifiers perform well in discriminating signal from background, with a Random Forest classifier with 1000 Decision Trees achieving a cross-validation accuracy of $93.92\% \pm 0.12\%$.

Deep neural networks improved upon this, reaching a cross-validation accuracy of $94.13\% \pm 0.05\%$. This required a lot more fine-tuning of the hyperparameters and implementation of data augmentation for the symmetries in the physical processes. Selecting the models based on the validation accuracy gave maximal validation accuracies of $94.26\% \pm 0.02\%$ across folds. However, tests with a separate test fold showed that the final epoch accuracies were a better proxy of the true accuracy.

Converting the tabular dataset to images and employing convolutional neural networks to distinguish between signal and background performed best. These obtain single-model maximal validation accuracies of $95.08\% \pm 0.04\%$ and achieve a 'test' accuracy of $95.41\% \pm 0.02\%$ when combined in an ensemble. This increase in performance is expected to arise from better extraction of spatial relations between the objects.

Combining the dense neural network and the convolutional neural network into a hybrid model did not give promising results, but was not tested to its full extent. Better investigation in this direction could improve the performance of the resulting classifier.

[1] Sascha Caron, Jong Soo Kim, Krzysztof Rolbiecki, Roberto Ruiz de Austri, and Bob Stienen. The BSM-AI project: SUSY-AI-generalizing LHC limits on supersymmetry with machine learning. *Eur. Phys. J.*, C77(4):257, 2017.

[2] Schnake, Simon. Energy Regression with Deep Learning in Particle Collider Physics.

[3] Machine Learning in Particle Physics and Astronomy. Week 2: Ensemble methods. BrightSpace, course material, 2019. [Online; accessed 14-Dec-2019].

[4] Dmytro Mishkin, Nikolay Sergievskiy, and Jiri Matas. Systematic evaluation of convolution neural network advances on the ImageNet. *Computer Vision and Image Understanding*, 161:11–19, Aug 2017.

APPENDICES

A. Transformation of features

The energies and transverse momenta exhibit long tails in their distributions, which can be squashed by the logarithmic function. Experiments showed that only providing these transformed quantities instead of the original quantities did not give good results. However, proving both the original and transformed version did seem to improve performance.

The conventional machine learning classifiers presented in this paper were all trained using both versions of the features. The deep neural networks shown in Figure 2 and Figure 3 were mostly trained using both versions, but the baseline (first model) and second test (dropout) were trained using the original version.

The pipelines creating the images for the convolutional neural network use the logarithmic versions of the energies and transverse momenta.

B. Image creation pipelines

For the convolutional neural networks, images have been created with several features as image channels. These images were created with min/max-normalisation to easily distinguish between features from measured objects and zero-padding. Furthermore, for the energies and transverse momenta, the log-transformed version was used. For all images, the azimuth of each object is binned to the image resolution and gives the y-coordinate. The pseudorapidity is binned to give the x-coordinate. The resolution of all images is set to 32×32 pixels.

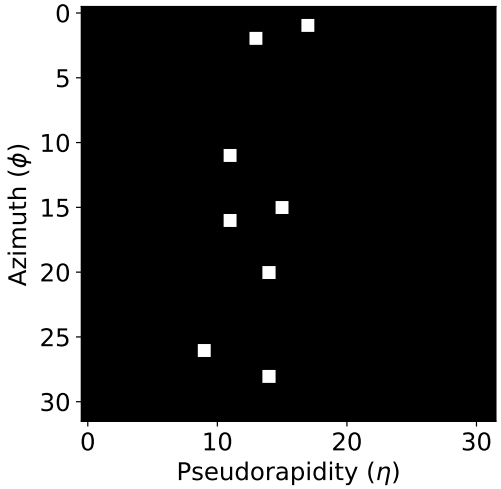

Figure B1. Sample image from pipeline *a*.

a. Object location
This pipeline only contains the location of the objects, and thus does not contain the type of object or any other object properties. The result is an image with one channel (greyscale). A sample image from this pipeline is given in Figure B1.

b. Object type
For this pipeline the image channel denotes the type of object (e^-, e^+, ...), giving images with seven channels. Only the location and object type are provided, so each pixel's channel is either one or zero.

c. Object energy
This pipeline replaces the mere presence of objects at certain locations with the energy of the respective objects. The min/max-normalised log-transformed version of the energies are used, which code for the pixel channel's intensity.

d. Event MET
Here the missing transverse energy of the unobserved objects is added as an additional channel. As x-coordinate the middle of the image is chosen.

e. Object momentum
The final pipeline adds the transverse momentum for each object type as channels, giving seven additional channels.

C. Hyperparameter sweeps

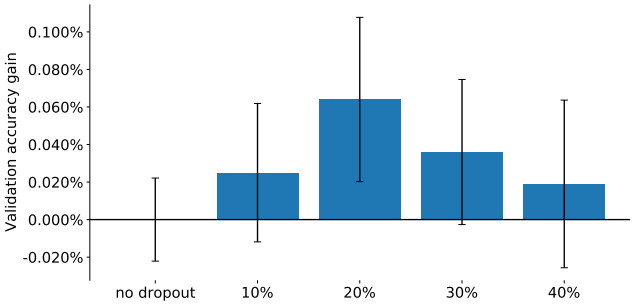

Figure C1. Maximal validation accuracies obtained by the deep neural network with different dropout rates, compared to the baseline maximal validation accuracies of 94.05% ± 0.02%.

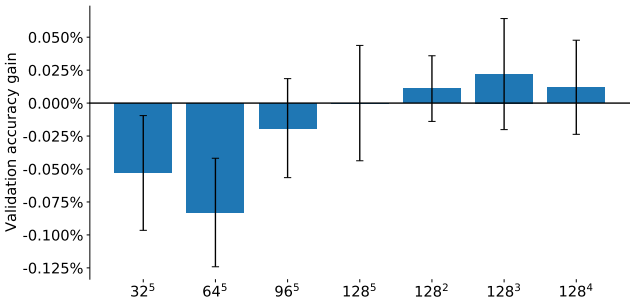

Figure C2. Maximal validation accuracies obtained by the deep neural network with different architectures, compared to the baseline maximal validation accuracies of 94.11% ± 0.05%. The base number represents the number of nodes per layer, with the exponent denoting the number of layers. So 128^5 means five layers of 128 nodes.

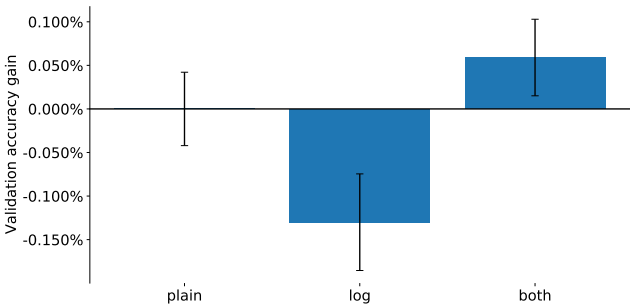

Figure C3. Maximal validation accuracies obtained by the deep neural network with different transformations of the features, compared to the baseline maximal validation accuracies of 94.13% ± 0.05%.

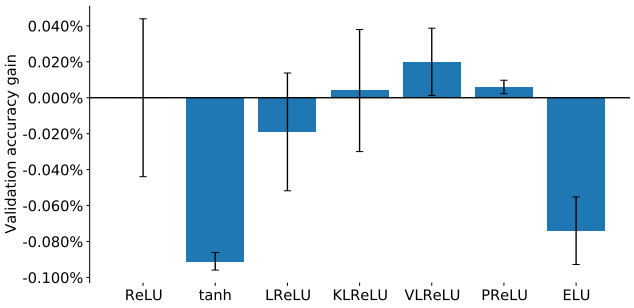

Figure C4. Maximal validation accuracies obtained by the deep neural network with different activation functions, compared to the baseline maximal validation accuracies of 94.19% ± 0.05%.

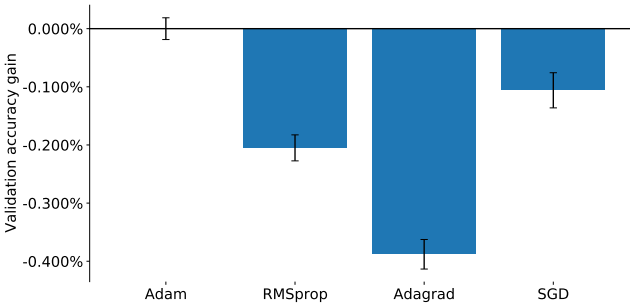

Figure C5. Maximal validation accuracies obtained by the deep neural network with different optimisers, compared to the baseline maximal validation accuracies of 94.21% ± 0.02%.

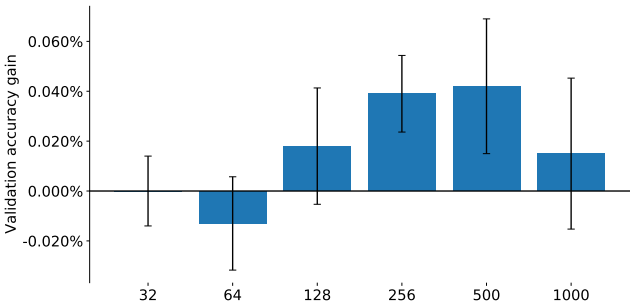

Figure C6. Maximal validation accuracies obtained by the deep neural network with different batch sizes, compared to the baseline maximal validation accuracies of 94.21% ± 0.02%.