AJC
EAM

Adversarial Neural Network on subjetiness substructure observables for mass-decorrelated signal tagging

Manuel A. Toledo Lugo[1] and Rafael Espinosa Castañeda[2]

[1] *Instituto Tecnológico de Monterrey, Campus Monterrey, Nuevo León, México*
[2] *Instituto Tecnológico de Monterrey, Campus Querétaro, Querétaro, México*

Reception date of the manuscript:
Acceptance date of the manuscript:
Publication date:

Abstract— The search for new physics through the hadronic decay of massive particles collision has taken a different approach, as Machine Learning techniques present in the field can be used and optimized for anomaly-detection. However, neural networks may infer non-linear correlations from the invariant mass of jets, causing artificial bumps in the invariant mass spectrum, which decrease the classification performance of neural networks. An adversarial training strategy is implemented to decorrelate the classifier of signal (new physics) events and the invariant mass of dijets, so the performance of the tagger can be compared with previously created techniques on background rejection

Keywords— High energy physics, jets, QCD, neural networks, adversarial, invariant mass, ML, Deep Learning

I. Introduction

Multiple ways of searching for physics Beyond Standard Model (BSM) have been developed, as no new discovery has arisen since 2012 [1]. The Large Hadron Collider's (LHC) technology is powerful enough to attain almost the speed of light on proton-proton (pp) collisions, suggesting that a different approach might be needed to improve the search. As Machine Learning rises on popularity and has proven to be useful in a variety of subjects, Deep Learning methods have gained relevance in the search of physics BSM [2, 3, 4].

The procedure for the search of BSM relies on the hadronic decay of particles created on the pp collisions. The LHC boosts protons with enough energy to 'spark' heavy particles into existance, such as the W, Z and Higgs bosons, or top quarks. These particles decay so rapidly, that its existance after collision can only be confirmed through the resonance or trace they leave. The decays of a boosted object can be reconstruced as jets with a certain radius value, which can be used to distinguish whether the decays are resonant (what shall be known as 'signal'), or usual Quantum ChromoDynamics (QCD) non-resonant background. The jet substructre [5] contains different observables with information about the angular energy distribution, and are detailed in Section II.

However, standard multivariate analysis (MVA) taggers tend to use jet mass as an important feature to determine the existence of signal and misshape the jet mass distribution of QCD background jets into a signal-like distribution, making the tagger less precise.

Similar to other strategies [6], an adversarial trained neural network has the ability to classify events properly by decorrelating the jet substructure observables from the invariant mass of the jet.

In this paper, we elaborate on the unused strategy Adversarial Neural Network (ANN) for the LHC data that punishes its classifier when jet mass is inferred to tag a jet, and compared with previously created models for the same data [7].

Contact data: Manuel A. Toledo Lugo | a00828623@tec.mx, Rafael Espinosa Castañeda | rafael.espinosa.castaneda@tec.mx

II. Simulation Data and Observables

a. MC simulation

LHC Olympics 2020 competition released data containing di-jets hadronic decay information, where signal is given by boosted boson $W' \to XY$ and $X \to q\bar{q}$, $Y \to q\bar{q}$ and the background are common QCD events. Multiple Neural Network strategies (like VAE and CWoLa) have been applied to the dataset, training and evaluating the network's validity, leaving some other strategies off the table.

The Monte Carlo simulated dataset consists of 1M QCD dijet events (what is ought to be considered as Background, BG) and 100k $W' \to X(\to q\bar{q})Y(\to q\bar{q})$ events (the signal, where the dijet is a pair of quark-antiquark), and the masses values for each particle are $m_{W'} = 3.5$ TeV, $m_X = 500$ GeV and $m_Y = 100$ GeV. The events were produced using Pythia8 and Delphes 3.4.1, and selected using a single fat-jet (R=1)

trigger with pT threshold of 1.2 TeV.

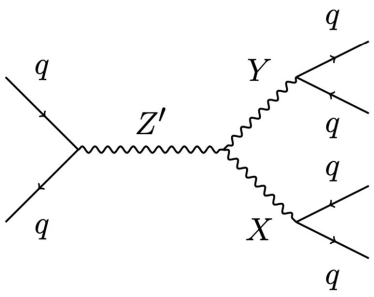

Fig. 1: Feynman Diagram for a boson W decay

b. Observables

The features for each jet are the 3-momenta, invariant masses (p_{xj}, p_{yj}, p_{zj}), and n-jettiness variables τ_1, τ_2, τ_3, which can be used to produce what is known as the observables $\tau_{21} = \frac{\tau_2}{\tau_1}$ and $\tau_{32} = \frac{\tau_3}{\tau_2}$.

τ_N is a jet shape called "N-subjettiness", and it "counts" the number of subjets that a main jet may have. After reconstructing a candidate W jet and a N number of possibles subjets (using jet algorithms like anti-kt), the τ_N observable can be calculated through:

$$\tau_N = \frac{1}{d_0} \sum_k p_{T,k} min\{\Delta R_{1,k}, \Delta R_{2,k} \cdots \Delta R_{N,k}\} \qquad (1)$$

where k : are the constituent particles in a given jet, $p_{T,k}$: is the transverse momenta, and $\Delta R_{J,k} = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$: is the distance in the rapidity-azimuth plane between a candidate subjet "J" and a constituent particle "k" [8].

To identify boosted two-prong objects like the W boson, τ_{21} is the designated observable to use. One advantage of using N-subjettiness is that this measurement is more direct on the desired energy flow properties, making it easier to find jets containing two or more energy lobes. Secondly, this variable allows to adjust the relative degree of signal efficiency and BG rejection without intensive algorithmic adjustments.

For this study, τ_{21} and τ_{32} of each jet shall be used as features to identify which event contains signal data and which are rejected as background events.

III. Method

a. Adversarial Neural Networks

An Adversarial Neural Network is proposed as a possible improvement on actual neural network results [7, 2], due to its decorrelating property that separates results from being influenced by the jet mass and obtaining a cleaner outcome.

This method consists of two network architectures:

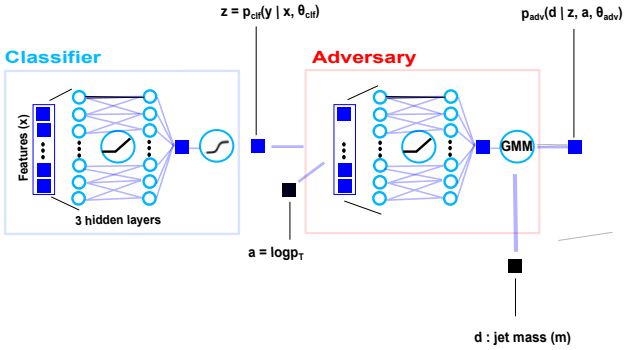

Figura 2: Tagger architecture schematic

where the Classifier Network receives the features as input, and outputs a jet tagging variable $\in [0,1]$ that determines whether a signal (1) or background (0) is tagged. This tagger can work by itself by learning to distinguish between BG and signal and minimizing its classification loss (Called Binary Cross Entropy, given by $L_{clf} = -ylogp_{clf}(y|x,\theta_{clf}) - (1-y)log(1-p_{clf}(y|x,\theta_{clf}))$, which gives a probability a probability of the event being BG or signal in range [0,1] respectively),
and the features are represented as X and the jet labels as Y. However, it is necessary to adjust this loss function in order to consider the adversary part.
The adversary network is tasked with inferring the jet mass from the output and minimize its loss function (L_{adv}), but no linear correlation coefficient is enough to express non-linear correlations between a Multivariate Analysis (MVA) jet tagger and the jet mass. Another function is needed for the adversary model, so the Mean Squared Error (MSE) loss function is implemented ($L_{adv} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$).
which maximizes a conditional probability p_{adv}, dependent of the output of the classifier z, an auxiliar $a \sim A$, and the adversary weights. This posterior probability density function (p.d.f) is explained in the next section.
The loss functions for the entire tagger can be simplified as:

$$\min_{\theta_{clf}} \left(\max_{\theta_{adv}} \left(L_{clf}(\theta_{clf}) - \lambda L_{adv}(\theta_{clf}, \theta_{adv}) \right) \right) \qquad (2)$$

where θ are the weights for its network and λ can be considered as an additional hyperparameter to optimize. It regulates the impact of the adversary network on the tagging task ($\lambda \to 0$ is a normal classifier, $\lambda \to \infty$ get "orthogonal" to mass information), so that the classifier maximizes L_{adv} making it harder to infer mass, but the tagger minimizes the total loss of the network, resulting in an accurate prediction on signals.

b. Configuration

Both neural networks are built separately, and joined together using a gradient reversal layer.
The classifier is composed of 3 fully connected layers with 64 neurons each. This layers are paired with a batch normalization layer meant to accelerate the training, by eliminating

the internal covariate shift (complexity due to change in the internal distribution of each layer's input), and allowing the use of higher learning rates [9]. It receives the number of observables to use in its Input Layer, and ends with a single neuron layer with "sigmoid" activation to determine if the jet is signal or not (Classifier architecture is found in Figure 5). This information is passed to the adversary, along with auxiliar variables that contains jet mass information that the network can use, which is the $logp_T$ of every jet. This network needs the mass as an input too, so it can decorrelate the classifier, but it is used for the Adversary. The adversary and classifier share almost exactly the same structure for the hidden layers, but with a Concatenate layer to put together the pT and $z \sim p(y|x, \theta_{clf})$ inputs, and a Posterior probability distribution function in the jet mass using Gaussian Mixutre Model [10].

All training is done using the *Adam* optimizer, and 'ReLu' non-linear activation function. In order to get the best results, and properly introduce the information needed in the adversary network, a first training session is applied for 200 epochs only for the classifier, so it tags signal without a decorrelation. This is followed by a 10 epoch pre-train only on the adversary, so it is conditioned to the classifier. Finally, both models are combined and work together for another 200 epochs.

This strategy allows the adversary neural network to be preconditioned to the classifier, as the latter is trained by itself first, so the adversary training is done properly by training both networks simultaneously.

c. *Evaluation metrics*

As the tagger efficiency relies on the decorrelated classifier's, a comparison in its signal efficiency ε_{sig}^{rel} versus background rejection $1/\varepsilon_{bg}^{rel}$is executed, where each is defined respectively as:

$$\varepsilon_{sig}^{rel} = \frac{N_{sig}^{tagged}}{N_{sig}^{total}}, \qquad \frac{1}{\varepsilon_{bg}^{rel}} = \frac{N_{bg}^{total}}{N_{bg}^{tagged}} \tag{3}$$

The performance of the model is shown as a Receiver Operating Characteristic (ROC) curve, which compares the True positive rate and the inverse False rate defined.

IV. Results

The accuracy of the model is tested and compared against four different strategies used at LHC Olympics for the same dataset, a weakly supervised model, an autoencoder, along with CWoLa, Optimal CWoLa, SALAD and SA-CWoLa (combination of both) approaches that are developed in [7]; these models have a simple architecture, and their performance may develop better results under different circumstances:

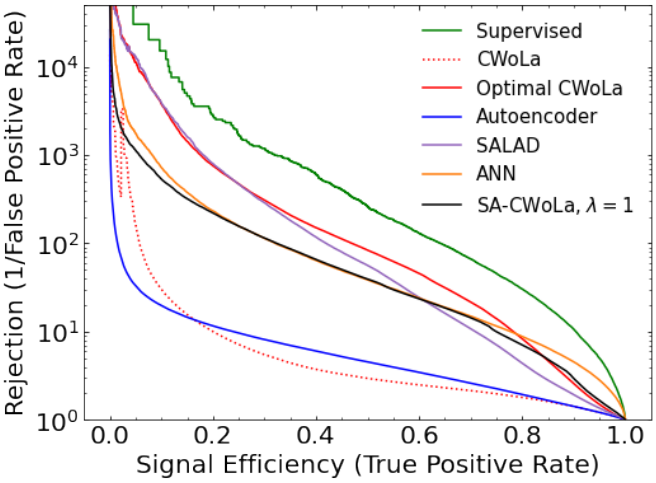

Fig. 3: ROC curves for performance comparison for different model strategies for low dimensionality input

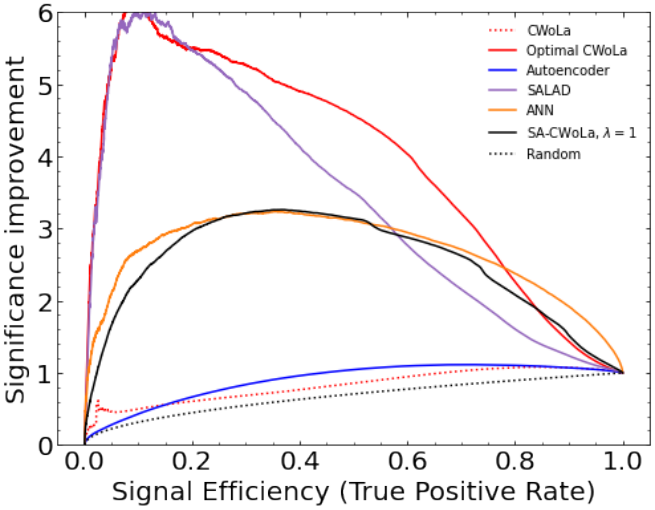

Fig. 4: Significance improvement on different strategies

It is expected to have a rather low performance on the autoencoder, as this model goes through a latent space of 2 dimensions, making the compression of the information very shallow. As for the SALAD and Optimal CWoLa results, the adversarial neural network lies slightly under their performances. However, CWoLa and SALAD can be seen as a strategy that prepares data for a better decorrelated outcome, as proven in [7], opening the possibility to combine said ideas. In addition, the ANN and SA-CWoLa strategies have a similar performance for both the rejection of background and the significance improvement of the model. This result is promising, as the combination of two ideas have shown to take a step forward on signal detection.

V. Conclusions

Having a variety of options to classify events based on the substructure observables is needed based on the properties of the model. Some strategies learn to use different implicit information that can be exploited to improve the search of new physics, and the position and relevance of the ANN performance make a valuable path to follow on tagging-related situations.

The adversarial training method has proven to be sufficient to identify resonance decay through the few observables τ_{21} and τ_{32}. Studies with a larger number of observables introduced as input show better results for decorrelating mass from the model [11], but for a low dimensional input, an ANN is comparable with other strategies, with mass decorrelation as the key feature of the study.

Finally, improvements on the background rejection can be made through different additions. The CWoLa strategy can be thought of as preprocessing data for the analysis, where injections of known signal is applied on previously separated sample groups, letting the model to be even more decorrelated and precise. Additionally, the possibility of using formerly decorrelated observables, rather than data, such as τ'_{21} or τ''_{21} [12] may allow to get even a more sensitive classifier.

VI. Acknowledgments

Manuel Toledo would like to thank Rafael Espinosa for the valuable insights and feedback given throughout the research. Additionally, special thanks to Michelle Calzada for her help and advice with the code. Finally, the researcher Andreas Søgard has been indispensable for the development of the paper throughout his multiple scientific papers on the subject and his advice.

a. Bibliographic citations

References

[1] The ATLAS Collaboration, "Observation of a New Particle in the Search for the Standard Model Higgs Boson with the ATLAS Detector at the LHC," 2012. [Online]. Available: https://doi.org/10.1016/j.physletb.2012.08.020

[2] B. Bortolato, B. M. Dillon, J. F. Kamenik, and A. Smolkovič, "Bump hunting in latent space," 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2103.06595

[3] The ALTAS Collabration, "Identification of hadronically-decaying w bosons and top quarks using high-level features as input to boosted decision trees and deep neural networks in atlas at $\sqrt{s}$ = 13 tev," 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2103.06595

[4] Y. Park, "Concise logarithmic loss function for robust training of anomaly detection model," 2022. [Online]. Available: http://arxiv.org/abs/2201.05748

[5] A. J. Larkoski, I. Moult, and B. Nachman, "Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning," 2017. [Online]. Available: https://doi.org/10.1016/j.physrep.2019.11.001

[6] J. Dolen, P. Harris, S. Marzani, S. Rappoccio, and N. Tran, "Thinking outside the rocs: Designing decorrelated taggers (ddt) for jet substructure," 2016. [Online]. Available: https://doi.org/10.48550/arXiv.1603.00027

[7] K. Benkendorfer, L. L. Pottier, and B. Nachman, "Simulation-assisted decorrelation for resonant anomaly detection," 2020. [Online]. Available: http://arxiv.org/abs/2009.02205

[8] J. Thaler and K. V. Tilburg, "Identifying Boosted Objects with N-subjettiness," 2011. [Online]. Available: https://arxiv.org/abs/1011.2268

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1502.03167

[10] C. M. Bishop, "Mixture density networks," 2017. [Online]. Available: https://research.aston.ac.uk/en/publications/mixture-density-networks

[11] A. Søgaard, "Constructing mass-decorrelated hadronic decay taggers in atlas," 2020. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1525/1/012117/pdf

[12] C. Shimmin, P. Sadowski, P. Baldi, E. Weik, D. Whiteson, E. Goul, and A. Søgaard, "Decorrelated jet substructure tagging using adversarial neural networks," 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1703.03507

VII. Appendix

a. Architecture of models

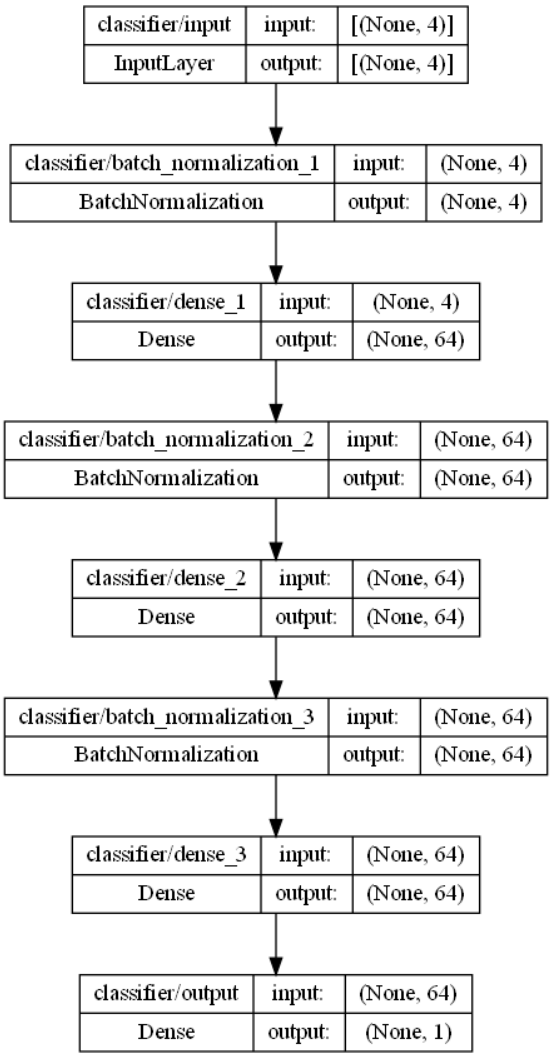

Fig. 5: Classifier architecture

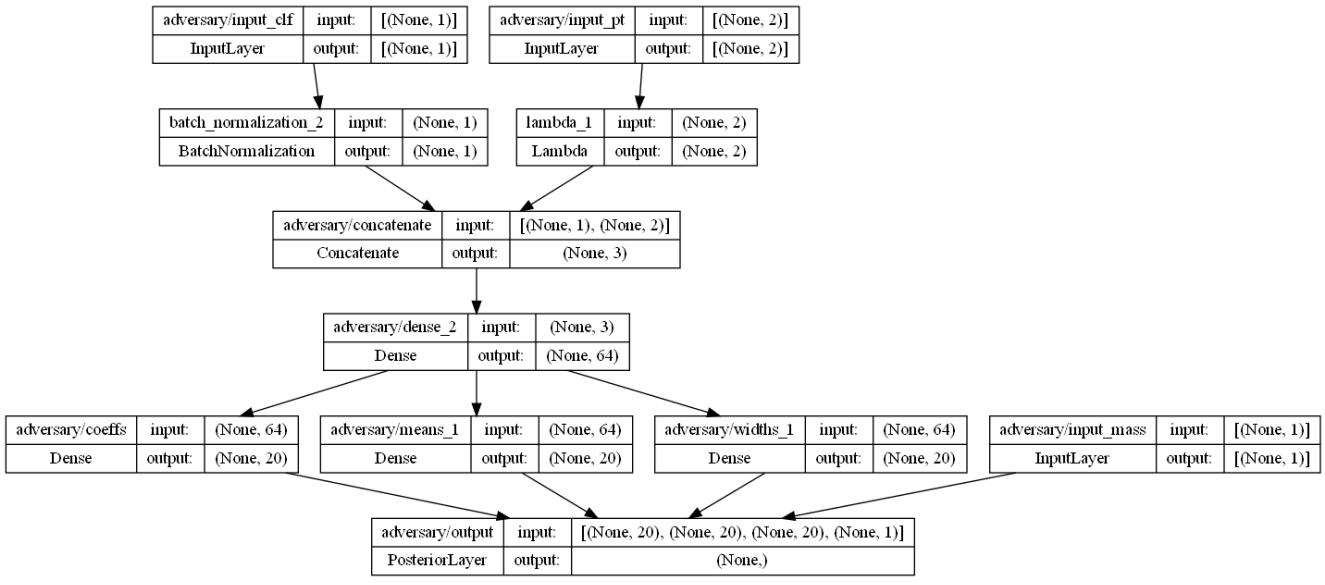

Fig. 6: Adversary architecture

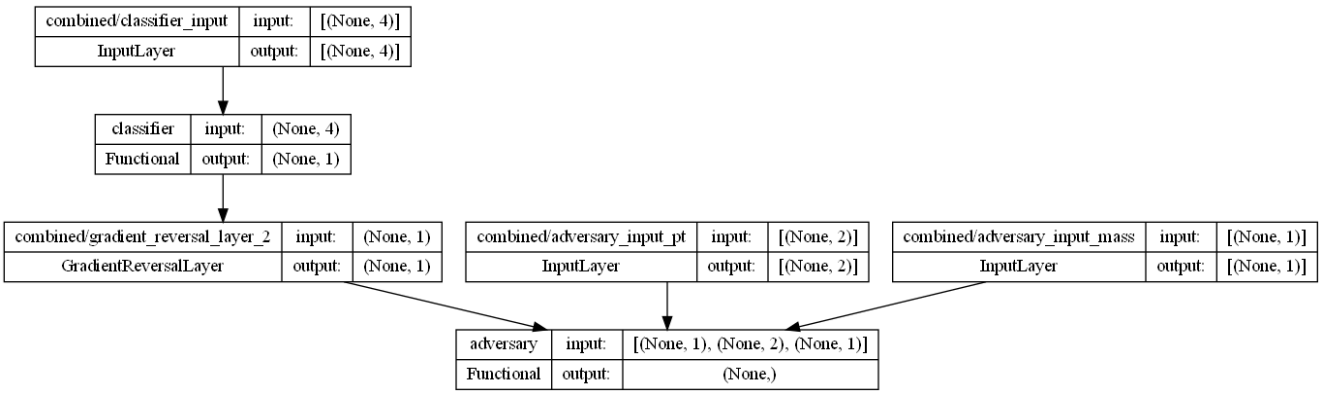

Fig. 7: Combined architecture

b. Github repository

https://github.com/ToledoManuel/Adversarial-NN