Separating between background and noise in search for a new Higgs-like particle

A. Kalechman

Raymond and Beverly Sackler School of Physics & Astronomy, Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel

November 2024

1. Introduction

The standard model is the most comprehensive and accurate model for particle physics today, which describes almost all known physics. It divides our world into Fermions, the elementary particles that make up all matter in the universe (excluding dark matter), and gauge bosons, the mediators of the fundamental forces (excluding gravity, as the corresponding particle, the Graviton, has not yet been detected).

A unique particle in the SM is the Higgs boson, which is an elementary particle produced by excitations in the Higgs field. According to the Standard Model, the Higgs field 'gives' rest mass to all massive elementary particles, including the Higgs itself. The Higgs particle is a boson with no electric and color charge, zero spin and rest mass of $125 \pm 0.11 \left[\frac{GeV}{c^2}\right]$. The Higgs was first predicted by Peter Higgs in 1964 according to the Standard model, and its existence was confirmed in CERN in 2012.

Many indications suggest that the Standard model is incomplete and needs further development, particularly in the search for Higgs-like particles. One significant example is that, according to the Standard Model Neutrinos are massless, but Neutrinos have very small mass (this discovery is known as the solution to the Solar Neutrino problem) [2].

As Run 3 is taking place at CERN, there are promising opportunities to discover Higgs-like particles. However, one of the primary challenges in identifying new particles is distinguishing the signature of a signal from that of backgrounds originating in Standard Model processes. The group's goal is to explore the possible existence of a lighter Higgs-like particle. The predicted Higgs-like particle (which we will refer to as X particle) can be created via many processes. In this paper we will focus on the ttX channel which is produced in association with top anti-top ($\bar{t}t$) quarks. We will also focus mainly on the semi-leptonic decay of $\bar{t}t$, where one quark decays hadronically and the second one decays leptonically (see Fig. 1 for full Feynman diagram). The predicted Higgs particle is expected to be unstable and has a few decay channels. We will focus on decay channel involving pairs of tau leptons as this is an unexplored field. As the tau-leptons have small angular separation we will refer to them as a new object called Ditau). The existence of the X particles will be inferred from the kinematic properties of the Ditaus and the other byproducts of $\bar{t}t$ decay.

In this work, we introduce a neural network-based classifier designed to distinguish between standard events and signatures indicative of a new particle signal. The dataset used for training and evaluation is generated through Monte Carlo simulations.

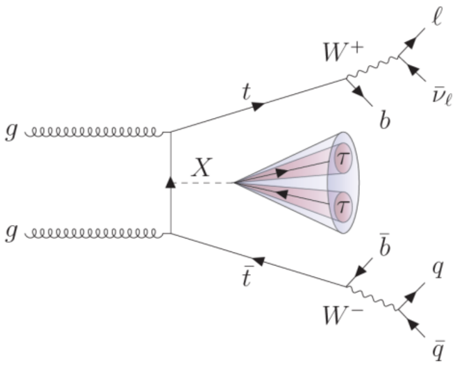


Fig 1. Feynman diagram of X particle through ttX channel

2. Data processing

2.1 Filtering signal and background

Two datasets are considered, one simulates the signal containing X particles and the other contains $\bar{t}t$ interactions within the framework of the Standard model with final states similar to those of the X particle. We apply the following conditions to both signal and background datasets to target only interactions which can produce the X particle:

1. Exactly one lepton in each interaction – in our case either electron or muon
2. At least 3 jets with at least one of them being bottom quark jet (will be referred do as bjet).
3. At least one Ditau that isn't overlapping a lepton or a jet and with appropriate kinematics.

Additionally, a kinematic filter was applied to each particle type to select only those within a specific range of kinematic properties (see kinematic filters per particle in Table 1).

Particle type	p_T cutoff (GeV)	η cutoff	JVT cutoff	BDT cutoff	Number of particles
Lepton	$p_T > 27$	$\eta < 2.7$			$n = 1$
Jet	$p_T > 20$	$\eta < 2.8$	If $p_T < 60$ and $\|\eta\| \leq 2.4, JVT > 0.5$		$n \geq 3$
Bjet	$p_T > 20$	$\eta < 2.8$	If $p_T < 60$ and $\|\eta\| \leq 2.4, JVT > 0.5$		$n \geq 1$
Ditau	$p_T > 50$	$\eta < 2.5$		$BDT > 0.2$	$n \geq 1$

Table 1: Applied filters per particle type. Empty cells mean no filter was applied.

Another test performed was the overlap removal, which involved checking the spatial positions of different particles to ensure they were not overlapping, as overlapping particles could lead to mis-readings in the detector.

To evaluate the effectiveness of our pre-selection, we define the following metric:

$$(1) \quad z = \frac{s}{\sqrt{b}}$$

Here, s denotes the number of selected signal events, and $\sqrt{b}$ represents the uncertainty in the background count, assuming the background follows a Poisson distribution. An efficient preselection cutoff will be for $z \gg 1$, meaning that the number of signal events is much bigger than the uncertainties in the background preselection.

As previously mentioned, the number of events obtained does not account for the varying cross-sectional areas. In general, the background cross-section is significantly larger than that of the signal. Consequently, when cross-sections are taken into consideration, the final number of events will result in s << b and thus $z \ll 1$.

Building a neural network will enhance efficiency by allowing us to analyze each event individually, rather than relying on statistics for the entire dataset.

2.2 Input variables

After applying the cutoff described in the previous section, histograms for each variable were plotted. This allows us to assess data integrity and identify the variables most relevant for distinguishing between background and signal. For a complete list of input variables, refer to Table 3.

Input Variable	Meaning
Ditau pt	Transverse momentum of Ditau
Ditau eta	Eta coordinate of Ditau
Ditau BDT	BDT score of Ditau – indicating probability of real a Ditau present
Num Ditau	Number of Ditaus in interaction
Ditau subjet subl pt	Transverse momentum of less energetic tau
Ditau subjet subl eta	Eta of less energetic tau
Ditau subject subl phi	Phi of less energetic tau
Ditau subjet lead pt	Transverse momentum of more energetic tau
Ditau subjet lead eta	Eta of more energetic tau
Ditau subject lead phi	Phi of more energetic tau
MetTST met	Energy detected in the full simulation
MetTST sumet	
MetTrack met	Energy detected in the inner simulation
MetTrack sumet	
Average Interaction Per Crossing	Average Interaction Per Crossing
Jet pt 1,2	Transverse momentum of the two most energetic regular jets
Jet eta 1,2	Eta coordinates of the two most energetic regular jets
Jet phi 1,2	Phi coordinates of the two most energetic regular jets
Jet JVT 1,2	Indicates if a jet comes from a pile-up or a hard interaction
Num bjet	Number of bjets in interaction
Num jet	Number of regular jets in interaction
Bjet pt	Transverse momentum of bjet
Bjet phi	Phi coordinate of the bjet
Bjet eta	Eta coordinate of the bjet
Bjet JVT	JVT value of the bjet
Lepton pt	Transverse momentum of the electron / muon
Lepton eta	Eta coordinate of the electron / muon
Lepton phi	Phi coordinate of the electron / muon

Table 3. List of input values and their meaning.

For example, the most distinct differences between background and signal distributions were observed in the ditau variables, particularly η and the BDT. In contrast, the signal and background distributions for leptons and jets appeared nearly identical. In total our dataset contained 34 variables, including a boolean column determining whether an event is background and signal (this will be used as y variable).

The dataset comprises a total of 753,388 events, evenly split between signal and background to eliminate bias arising from size disparities. Each event is characterized by 33 input variables, as detailed in Table 3. In each event only the two most energetic regular jets and the most energetic bjet were saved. The data was split into train – validation – test sets of sizes 70%, 15%, 15% of the original size, respectively.

Constructing the Neural Network

2.3 Introduction to Neural networks

Neural networks rely on a combination of layers and techniques to effectively learn patterns from data. Some of the main components are activation layers, dropout layers, dense layers, loss functions, each with its own purpose.

Neural networks consist of n layers, each with a different dimensionality, where the number of neurons in a layer defines its dimension. The network described in this paper is a fully connected network (FCN), meaning each neuron is connected to every neuron in the neighboring layers. During each iteration of the training phase, the weights of the neurons are adjusted to minimize the loss function, enabling the network to learn from the data.

The two most fundamental types of layers are dense layers and activation layers:

1. Dense layer – receives output from the previous layer, executes a linear transformation of the form: $y = wx + b$ where w is the weight and b is the bias [3].
2. Activation layer - To enable the network to learn non-linear patterns, the activation layer applies a non-linear transformation to the output of the dense layer. In this network, the activation function used is PReLU, which is defined as:

$$(2) \quad f(y_i) = \begin{cases} a_i y_i, & y_i < 0 \\ y_i, & y_i \geq 0 \end{cases}$$

Where a_i is a learnable hyperparameter which can be adjusted during training and y_i is the output of the i^{th} dense layer [4].

To transform the output into a probability, the output layer is an activation layer with the Sigmoid function, which takes the values $-\infty < x < \infty$ and transforms them to a value between 0 and 1:

$$(3) \; \sigma(x) = \frac{1}{1 + e^{-x}}$$

The method by which a network 'learns' is minimizing the loss function, which quantifies the error between the real data and the network's prediction. One of the most common loss functions for classification problems is the Binary cross entropy function, given by the following equation:

$$(4) \; L = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i)(1 - \log(1 - \hat{y}_i)]$$

Where y_i is the i^{th} sample, $\hat{y}_i$ is the i^{th}predicted probability for the positive class and N is the total number of samples.

To prevent the model from overfitting and enhance its ability to generalize to new data, a dropout layer was introduced. During training, the dropout layer randomly deactivates a determined percentage of neurons in each layer, which helps the model avoid relying too heavily on specific neurons and promotes better generalization [5].

2.4 Network structure

The purpose of the Neural network was to act as a classifier – for a given event with 33 input variable it would calculate a probability for the event being a signal event. Our network was built using Keras library in python.

For that purpose, the network type chosen is a fully connected neural network (also known as FCN), using the loss function 'Binary Cross Entropy' with 'Adam' solver and PReLU as the activation function. To transform the hidden layers output to a probability between zero and one the output layer was an activation layer with the sigmoid function (as defined in Eq. 3).

Our final network consisted of six dense layers, each with 64 units. Each dense layer was followed by a PReLU activation function and a dropout layer with a rate of 0.2, except for the last dense layer. This design was chosen to upscale the network dimensions beyond the 33 input variables in the dataset, enhancing its capacity for representation and learning.

A simpler version of the network was initially created, with six hidden layers and smaller dimensions. This version included one hidden layer with 16 units, four with 32 units each, and a final hidden layer with 16 units, all followed by PReLU activation functions. The output layer used a sigmoid activation function. See Appendix 1 for visualization of the networks structure. Both networks were trained in 100 epochs with a learning rate of 0.001 (see full results in Fig 5).

2.5 Results

Both networks provided good results with slightly better metrics for the final network, as expected, with validation accuracy reaching 0.9 and validation loss reaching 0.1 (see Fig. 5). The loss function plot shows that the training converged to a constant value, and the close match between training and validation loss suggests good generalization. The lack of a noticeable gap between the two losses indicates that overfitting was not an issue.

It is important to note that we set the 'default' probability threshold to 0.5, meaning any event with a probability greater than 0.5 is classified as signal. Our probability threshold, T, may not be optimal to accurately reflect the differences in physical cross-sections between signal and background processes. This can lead to bias in the result metric, especially the accuracy metric. For this reason, the ROC curve was used to compare the performance of both models with different probabilistic threshold (T).

The ROC curve is a plot of two quantities, True Positive Rate (TPR) and False Positive Rate (FPR). In our case we will define Signal efficiency ϵ_s and Background efficiency ϵ_b which are analogue to TPR and FPR, and are defined the following way:

$$(5) \; \epsilon_s = \frac{\int_T^{\infty} f_s(x)dx}{\int_{-\infty}^{\infty} f_s(x)dx}$$

$$\epsilon_b = \frac{\int_T^{\infty} f_b(x)dx}{\int_{-\infty}^{\infty} f_b(x)dx}$$

Where f_s, f_b are the probability density functions of the signal and background, respectively, and T is the probabilistic threshold selected. Each of these quantities reflect the ratio between 'good predictions' of a class to the total predictions of this class. Both these values range from zero to one. X axis will show the ϵ_b and y axis the ϵ_s. A good model will have a low ϵ_b and a high ϵ_s. A model which 'guesses' will be presented as a diagonal line from (0,0) to (1,1). A test metric property that quantifies the performance of the model is the Area Under the Curve (AUC) of the ROC curve, meaning a good model will have AUC close to one.

Determining the signal efficiency (ϵ_s) enables us to estimate the accuracy of labeling signal events as we adjust the threshold (T). For example, when ϵ_s= 0.9, we obtain T=0.91, and the preselection efficiency (Z) is 216 (as described in Eq.1). Similarly, for ϵ_s =0.7 T=0.98, resulting in Z=146. This outcome is reasonable, as a lower ϵ_s corresponds to fewer events being labeled as signal, thereby reducing Z. Comparing the Z value obtained from the neural network's results with the Z value from the preselection is irrelevant because many background events were filtered out before applying the neural network to prevent bias due to different sizes

Conclusions

The main goal of the project, creating a classifier between background and signal, was achieved with great success. Additional steps could be made to improve the network – as implementing weights to train on datasets which have different number of signal and background events. Additionally, data from different simulations could be added to better generalize the model.

References

[1] https://www.iop.org/explore-physics/big-ideas-physics/standard-model

[2] https://www.aps.org/archives/publications/apsnews/200206/sno.cfm

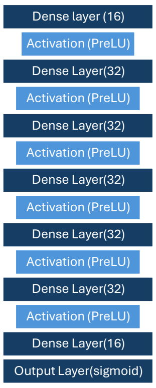
[3] https://keras.io/api/layers/core_layers/dense/

[4] https://en.wikipedia.org/wiki/Activation_function

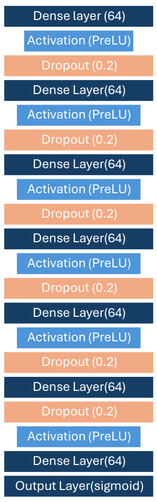
[5] https://keras.io/api/layers/regularization_layers/dropout/

Appendix

1. Networks structure:



Appendix 1.a: initial network structure



Appendix 1.b: Final network structure