SEARCH FOR THE FLAVOR-CHANGING NEUTRAL CURRENT IN TOP PAIR EVENTS WITH AN ASSOCIATED PHOTON USING 13 TEV PROTON-PROTON COLLISION DATA COLLECTED WITH THE ATLAS DETECTOR

by

JASON TYLER BARKELOO

A DISSERTATION

Presented to the Department of Physics
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

March 2020

DISSERTATION APPROVAL PAGE

Student: Jason Tyler Barkeloo

Title: Search for the Flavor-Changing Neutral Current in Top Pair Events With an Associated Photon Using 13 TeV Proton-Proton Collision Data Collected With the ATLAS Detector

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Physics by:

David Strom	Chair
James Brau	Advisor
Spencer Chang	Core Member
Dev Sinha	Institutional Representative

and

Janet Woodruff-Borden	Vice Provost and Dean of the Graduate School

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded March 2020

DISSERTATION ABSTRACT

Jason Tyler Barkeloo

Doctor of Philosophy

Department of Physics

March 2020

Title: Search for the Flavor-Changing Neutral Current in Top Pair Events With an Associated Photon Using 13 TeV Proton-Proton Collision Data Collected With the ATLAS Detector

Abstract for FCNC here.

This dissertation includes previously published and unpublished co-authored material.

CURRICULUM VITAE

NAME OF AUTHOR: Jason Tyler Barkeloo

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:
University of Oregon, Eugene, Oregon
Miami University, Oxford, Ohio
Wittenberg University, Springfield, Ohio

DEGREES AWARDED:
Doctor of Philosophy, Physics, 2020, University of Oregon
Master of Sceience, Physics, 2012, Miami University
Bachelor of Science, Physics, 2010, Wittenberg Univeristy

PROFESSIONAL EXPERIENCE:

University of Oregon, Graduate Research Assistant, ATLAS Experiment, June 2014-Present

University of Oregon, Graduate Teaching Assistant, September 2012-June 2014

Miami University, Teaching and Research Assistant, August 2010 - May 2012

Wittenberg University, Undergraduate Researcher, August 2008 - May 2010

GRANTS, AWARDS AND HONORS:

Weiser Senior Teaching Assistant Award – University of Oregon – 2014

American Association of Physics Teachers Outstanding Teaching Assistant – Miami University – 2012

PUBLICATIONS:

J. Barkeloo, J. Brau, M. Breidenbach, R. Frey, D. Freytag, C. Gallagher, R. Herbst, M. Oriunno, B. Reese, A. Steinhebel, D. Strom. “A silicon-tungsten electromagnetic calorimeter with integrated electronics for the International Linear Collider” J. Phys.: Conf. Ser. **1162** 012016 (2019).

J. Kangara, A. Hachtel, M. C. Gillette, J. Barkeloo, E. Clements, S. Bali. "Design and construction of cost-effective fail-safe tapered amplifier systems for laser cooling and trapping experiments", Am. J. Phys. **82**(8), 805 - 817 (2014).

A. Hachtel, J. Kleykamp, D. Kane, M. Marshall, B. Worth, J. Barkeloo, J. Kangara, J. Camenisch, M. Gillette, S. Bali. "An undergraduate lab on measurement of radiative broadening in atomic vapor", Am. J. Phys. **81**(6), 471 (2013).

Additional ATLAS Collaboration publications can be found:
http://inspirehep.net/search?p=exactauthor%3AJason.Barkeloo

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

CHAPTER I

SEARCH STRATEGY

1.1. Major Backgrounds

1.2. Event Reconstruction

1.3. Neural Network

To help distinguish signal events from the majority of background events a neural network was employeed for event classification. Neural networks are multivariate methods that take a variety of inputs and output a number between 0 and 1. The output value is a discriminating variable that will be used to classify events and determine which events make it into the final Signal Region selection. Signal-like events accumulate towards 1 while background-like events cluster around 0. Two neural networks are trained, one for the electron+jets final state and one for the muon+jets final state. This section will discuss the neural network studies completed and their uses in the search for FCNC events.

1.3.1. Input Variables

A wide variety of input variables to the neural network were studied in detail. Studies were done using only low level variables such as the kinematic variables (p_T, η, ϕ, E) of the physics objects in the signal region. This was done as a complex enough neural network should be able to figure out useful high level/event level variables (i.e. invariant masses, geometric separations) but in practice a combination of some of these low level variables and high level variables used as inputs to the neural network

Variable	Separation e+jets	Separation μ+jets
$p_T(\gamma)$	22.97	24.01
$m_{q\gamma}$	22.65	28.31
γ_{iso}	18.62	41.32
m_{bW}	11.10	11.70
$m_{l\gamma}$	9.00	7.51
$\Delta R_{j\gamma}$	4.59	5.66
ΔR_{bl}	4.99	4.47
m_T^W	3.16	3.37
S_T	3.78	3.32
n_{jets}	1.70	2.03
χ^2_W	1.37	1.91
$p_T(q)$	2.46	2.82
$\Delta R_{l\gamma}$	1.40	1.19
E (lepton)	0.86	0.89
$\not{E}_T$	0.47	0.70
$p_T(b)$	0.51	0.53

TABLE 1.1. Separation of normalized variables between signal and bacground in the e+jets and μ+jets channels for the variables used as input to the final neural network.

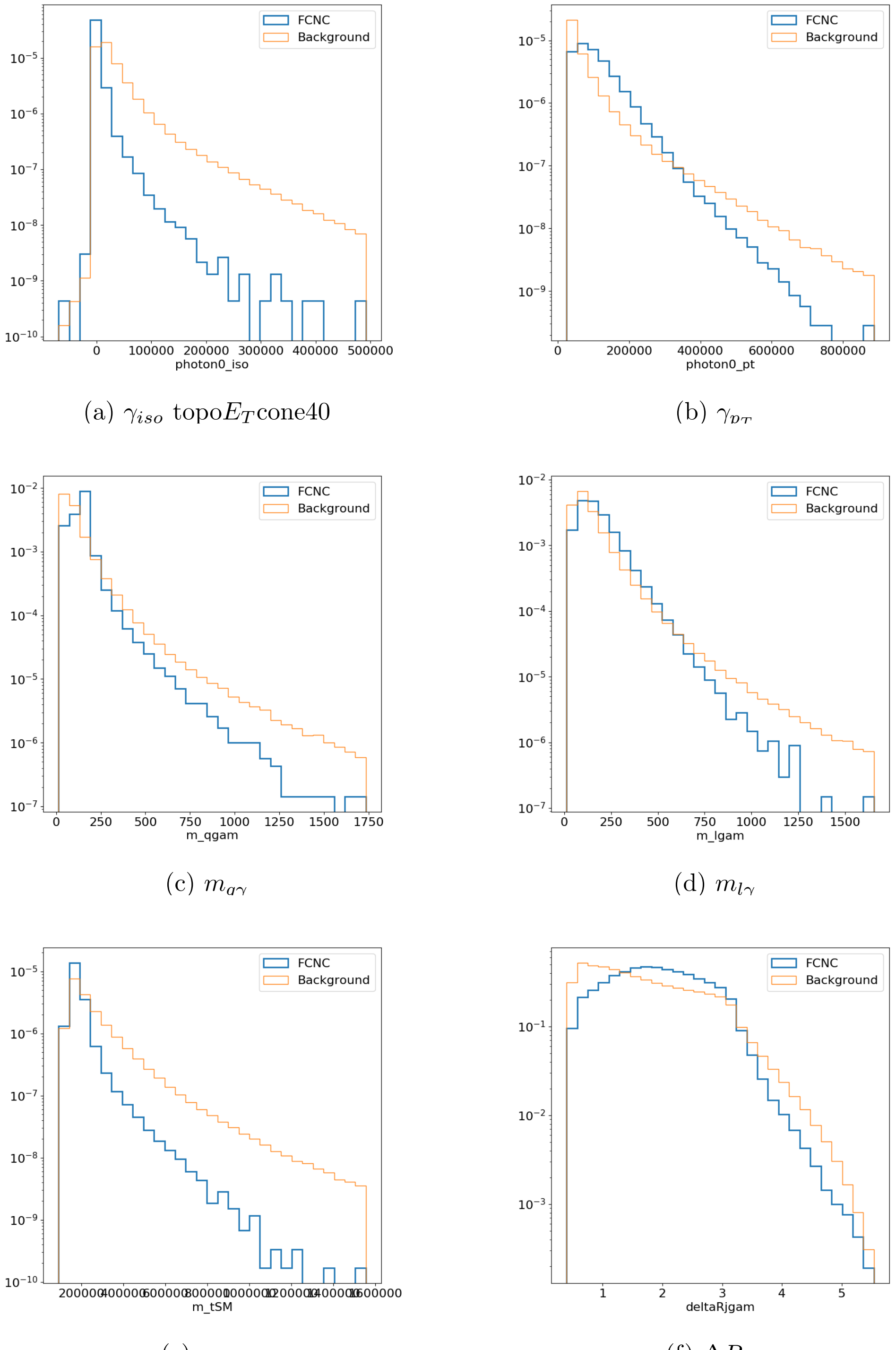
proved to give the best separation and projected limits. Using physical intuition to guide the neural network proved to be a valuable tool.

Combinations of 29 input variables were tested to start with however variables such as η and ϕ tend to not have significant weights in the neural network and are left out in favor the the high level variables that include them (e.g., ΔR values). A measure of how different the variables are between signal and background is the Separation. Table **??** shows the separation values for the variables that are inputs to the final neural network. Comparisons between the shapes of the input variables for the μ+jets channel are shown in Figures **??**, **??**, and **??**

$$\text{Separation} = \sum_{i}^{bins} \frac{n_{si} - n_{bi}}{n_{si} + n_{bi}}$$

Typically the kinematic variables with photon information have the biggest separation values. This is expected because the signal photon comes directly from the decay of a top quark and is much more energetic than background photons. Shape comparison plots for the e+jets channel and additional plots for other investigated variables are shown in Appendix **??**. The largest difference in separation between the e+jets and μ+jets channels is the photon isolation value. This is due to the fact that all backgrounds are included and fake photon contamination from a large Z+jets background are expected. Both networks preform similarly in their separation of signal and background events. The network is able to learn and compensate for this behavior with the help of other variables that include the lepton and photon: $\Delta R_{l\gamma}$ and $m_{l\gamma}$.

The neural networks are trained on MC events that have a chance of being in the signal region after basic event level cuts and optimized for signal significance. Only events with 1 photon ($15GeV$) and 1 bjet (MV2c10 77% working point) are classified by the neural network. The 77% working point was chosen by training the neural network on events with only 1 bjet at each working point: 70%, 77%, and 85% and picking the network and working point with the best estimated significance. The b-tagging neural network study is shown in Section **??**



(a) γ_{iso} topo$E_T$cone40

(b) γ_{p_T}

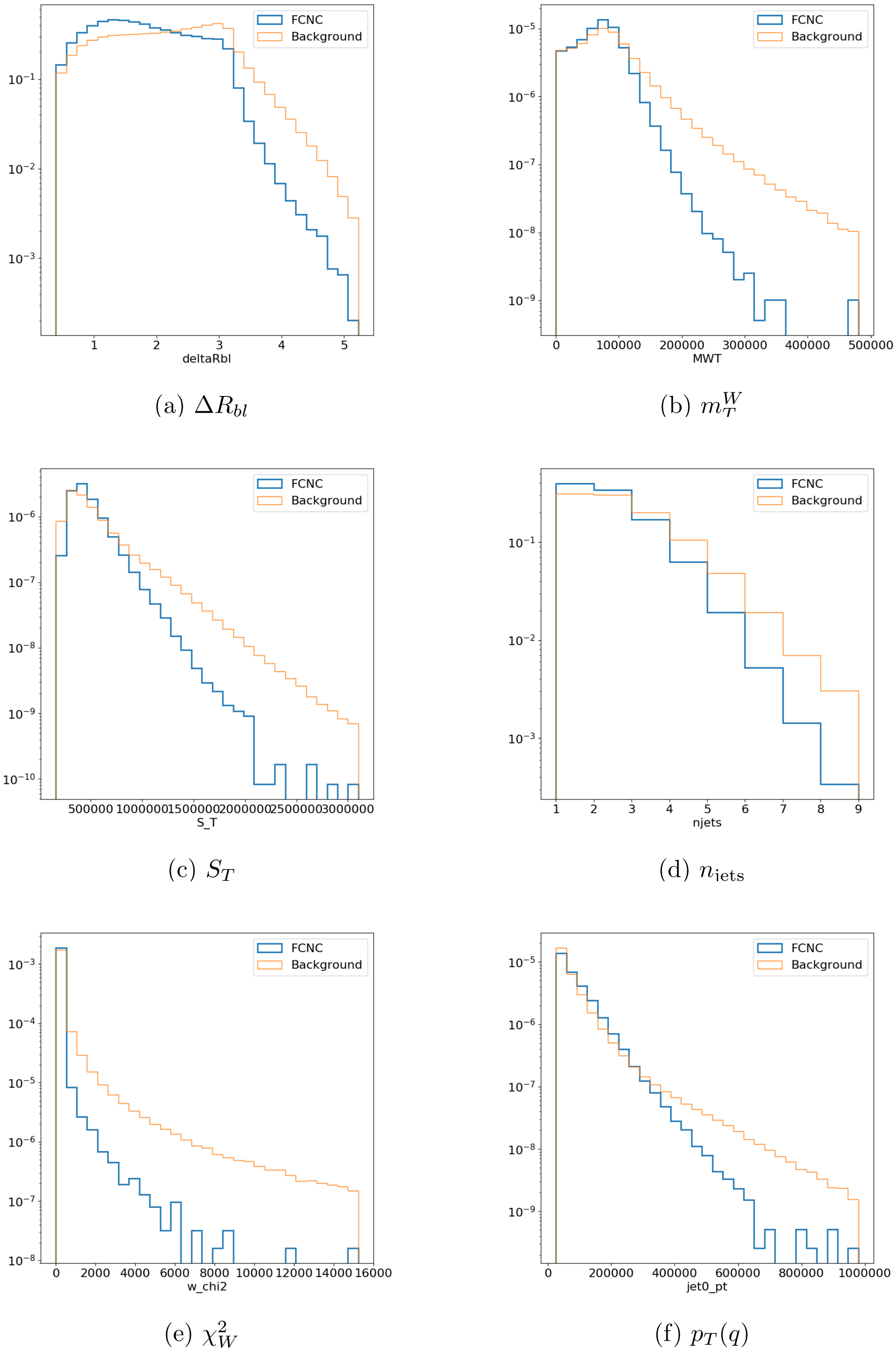
(c) $m_{q\gamma}$

(d) $m_{l\gamma}$

(e) m_{bW}

(f) $\Delta R_{j\gamma}$

FIGURE 1.1. Normalized variables showing the shapes of neural network input variables for the μ+jets channel: γ_{iso} topo$E_T$cone40, γ_{p_T}, $m_{q\gamma}$, $m_{l\gamma}$, m_{bW}, and $\Delta R_{j\gamma}$



(a) ΔR_{bl}

(b) m_T^W

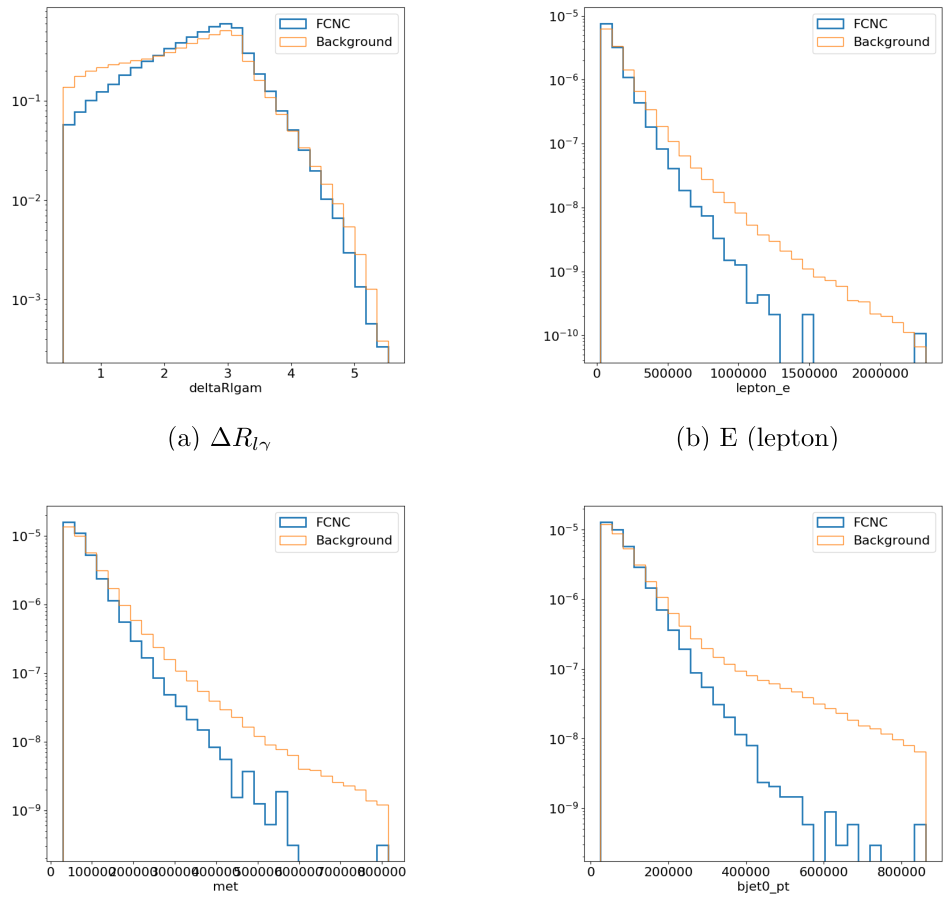
(c) S_T

(d) n_{jets}

(e) χ^2_W

(f) $p_T(q)$

FIGURE 1.2. Normalized variables showing the shapes of neural network input variables for the μ+jets channel: ΔR_{bl}, m_T^W , S_T, n_{jets}, χ^2_W, and $p_T(q)$



(a) $\Delta R_{l\gamma}$

(b) E (lepton)

(c) $\not{E}_T$

(d) $p_T(b)$

FIGURE 1.3. Normalized variables showing the shapes of neural network input variables for the μ+jets channel: $\Delta R_{l\gamma}$, E (lepton), $\not{E}_T$, and $p_T(b)$

1.3.2. Architecture

A variety of architecures of dense neural networks are studied using KERAS[?] on top of the TENSORFLOW backend [?]. Each network has a number of input nodes equal to the number of input variables. Networks with one, two, and three hidden layers are investigated each with 20 nodes. The output layer contains only a single node. Every node in one layer is connected to every node in the next layer and the previous layer. Every connection is assigned a weight that is optimized during the training of the network. For every node in the network a value is computed using the weights and input values of the previous nodes using an activation function. Nodes with the highest output of this function are more important to the fit. The activation function used on the internal nodes in this search is the Rectified Linear Unit activation function.

$$ReLU(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$$

The output layer uses the sigmoid function, $\sigma(x)$, as an activation function. The sigmoid function maps the output smoothly to the range (0,1).

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

Every training step the weights of each node are updated following an optimization algorithm, in this case the ADAM optimizer[?]. This optimizer follows the steepest gradient to reach the minimum of the parameter of interest called the loss function. The loss function used for these classification neural networks is the binary cross entropy:

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i\text{log}(p(y_i)) + (1-y_i)\text{log}(1-p(y_i))$$

where y is a binary indicator (0 or 1) if class label is the correct classification for observation and p is the predicted probability observation is the class label (0 or 1). The logarithmic nature of this loss function means it applys small values to correctly assigned event but more harshly punishes mismatching of events. Therefore having a similar number of signal and background events that get weighted similarly can improve the behavior of the network. In rare decay searches typically the amount of signal events is significantly smaller than the amount of background events in the training sample. Using the weight functionality in keras the total number of signal events can be scaled to be similar to the number of background events.

Weighting the signal events this way allows the network to separate the signal and background events in a way that is significantly less harsh than without the weights by taking advantage of the loss function being used. This improves the estimated significance of the neural network cut after the signal events are rescaled to their proper normalization values.

Various hyperparameters are used as inputs into the neural network as well as the optimizer used. The ADAM optimizer has a default learning rate of 0.001 which was not changed throughout these studies. The learning rate corresponds to the amount that weights are updated during training. A learning rate that is too large can mean the network never settles into a local minima as it is always missing the minima or at the very least it can take much longer to converge into a minima. As the neural network training for this search always converged quickly and to a similar value after being tested multiple different times the learning rate was not adapted.

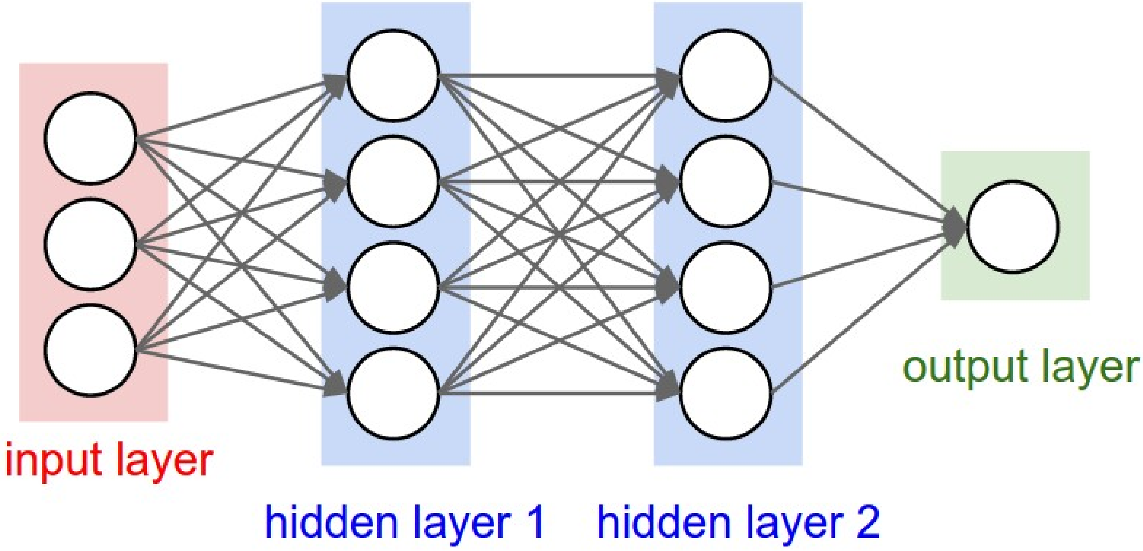


FIGURE 1.4. Pictoral representation of neural network architecture with 3 input variables, 2 hidden layers with 4 nodes each, and 1 output layer[**?**].

Another hyperparameter of note is the batch size which defines the number of samples that are propagated through the network at once. The batch size is of crucial importance in how long the training of the network takes. A set of 1000 training samples with a batch size of 100 will propagate each set of 100 samples through the neural network every epoch, so 10 separate batches. A larger batch size means that each epoch of the training takes a shorter amount of time. However, as the weights are updated after each batch the network can take many more epochs to converge as the weights are being updated less frequently. A batch size of 100 was used while training the networks presented in this chapter. Larger batch sizes were tested with the only difference being the time each epoch took and the total time the network took to converge.

Epochs are the total number of times the network has been trained over the entire training set. All of the networks were allowed up to 200 epochs to converge with a KERAS patience value set to 50. The loss function minimization would be done every batch and after each epoch the best possible value of the loss function is found. If this value is better than any previous epoch the network is allowed to train for 50 more epochs until 50 epochs have passed without finding a new minimum loss function value which then terminates the training. All models converge early and are terminated typically between epoch 80 and 120 meaning the loss function was minimized between epoch 30 and 70.

One method employed to avoid overtraining the network dropout regularization was used on each of the hidden layers. Dropout has the effect of simulating a large number of networks with very different network structures by removing A dropout rate of 20% was used meaning that for every batch 20% of the weights of the hidden layer nodes were set to 0. This forces the network to not become overly dependent on any given node and learning the data 'by heart' as opposed to recognizing the trends in the sample.

1.3.2.1. Training and Validation of Neural Networks

The input variables into the neural network are preprocessed using the RobustScalar method implemented in **scikit-learn**[**?**]. The preprocessing is done so that the input variables exist on a similar scale. As the network is tasked with learning how to combine these inputs through a series of linear combinations and nonlinear activation function values a disparity in the scales of the input values can lead to awkward loss function topology that will focus on certain parameter gradients instead of treating them all similarly. Normalizing the values to a standard scale allows the network to learn the optimal parameters for each input node more quickly and efficiently. This means that less focus can be used on the optimization of the hyperparameters for the network as the scales of the inputs do not need to be learned by the network itself.

Each input variable in the neural network, x, is scaled by the following equation:

$$z = \frac{x - m}{q_3 - q_1}$$

where m is the median of the distribution, q_1 and q_3 are the first and third quartile. This changes the distribution of the input variable distributions to be centered around zero.

A second method to avoid overtraining the neural network is to make use of a train-test split to split the signal and background samples into 3 independent randomized sets before training the neural network. The samples are split into a training set of 64% of the samples, a test set containing 20% of the samples, and the remaing 16% are a validation set. The training and test sets are used during the training of the network while the validation set is used to compute performance of the trained neural network.

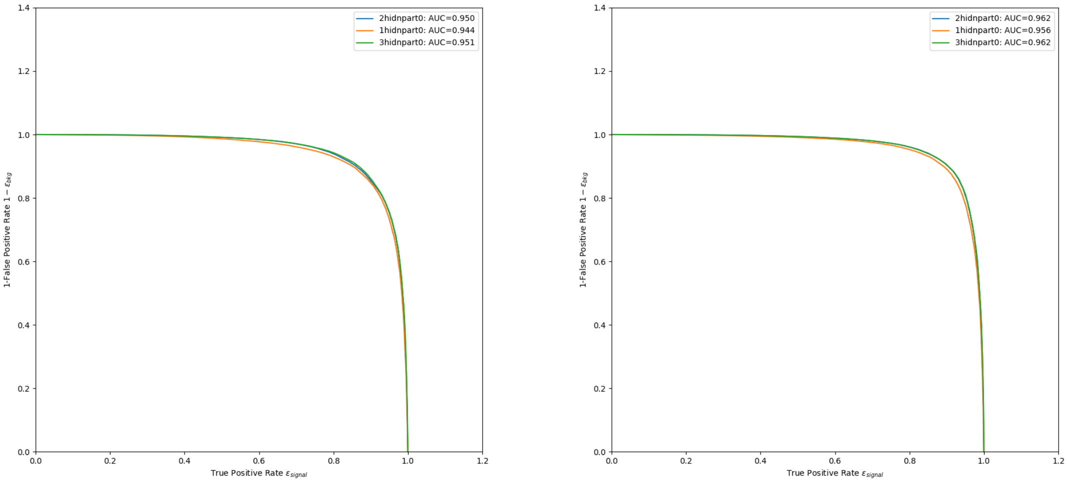
One measure of the performance of the network is the accuracy. The Keras default accuracy measure is defined:

$$\text{accuracy} = \frac{N(\text{event}_{NN} \geq 0.5|\text{signal}) + N(\text{event}_{NN} < 0.5|\text{background})}{N(\text{signal}) + N(\text{background})}$$

where $N(\text{event}_{NN} \geq 0.5|\text{signal})$ ($N(\text{event}_{NN} \geq 0.5|\text{signal})$) is the number of signal (background) events with $P_{\text{signal}} \geq 0.5$ ($P_{\text{signal}} < 0.5$). Essentially the accuracy is a measure of the mean of how often correct prediction values occur assuming a cut on the output of ≥ 0.5.

1.3.3. Hidden Layer Studies

The general performance of the neural network was studied with a varying number of hidden layers (1, 2, and 3) in both the e+jets and μ+jets channels. All of the networks are trained on the same set of variables and with the same train-test split input data. For each of the channels the *Receiver Operating Charactersitic* (ROC) curves are shown in Figure **??**. The ROC curves show the value of $1 - \epsilon_{\text{bkg}}$ as a function of the true positive rate, ϵ_{signal}. The figure of merit is the Area Under the Curve (AUC) which is a measure of how close the resulting values are to the optimal value of unity.

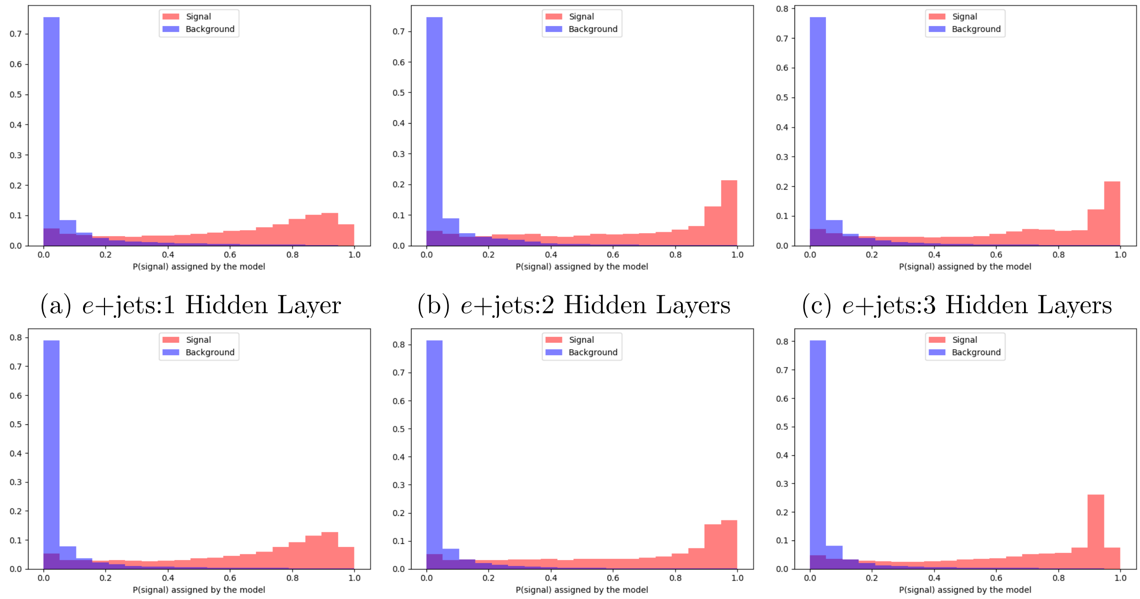


(a) e+jets ROC Curves

(b) μ+jets ROC Curves

FIGURE 1.5. ROC Curves are shown for both search channels for a varying number of hidden layers. Orange lines correspond to one hidden layer, blue to 2 hidden layers and green to 3 hidden layers. The blue and green curves have near identical AUC values.

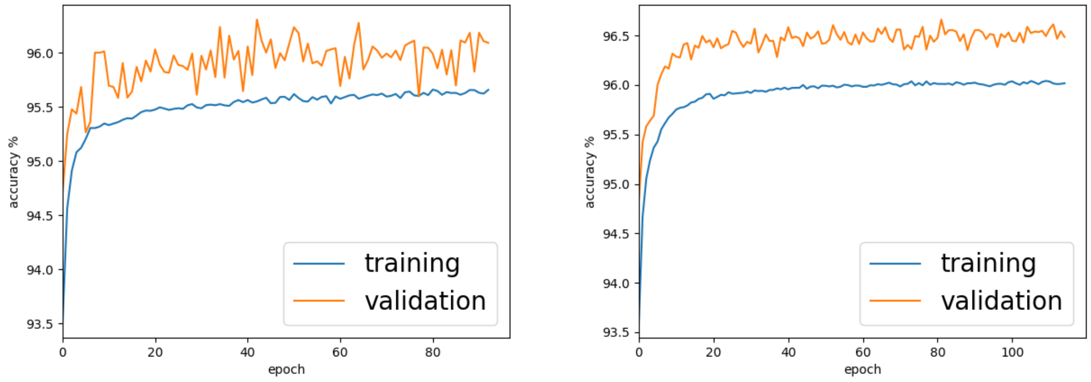
The AUC for 2 hidden layers and 3 hidden layers are identical, to rounding errors, for both channels. As such the network with 2 hidden layers has been chosen as it is computationally simpler. The normalized neural network output values are shown in Figure **??**. Adding a second hidden layer significantly improves the performance of the network but a third layer does not. The output shapes change slightly adding the third hidden layer due to the network learning differently about the same data. However, as the AUC showed the performance of 2 and 3 hidden layers is identical.



(a) e+jets:1 Hidden Layer (b) e+jets:2 Hidden Layers (c) e+jets:3 Hidden Layers

(d) μ+jets:1 Hidden Layer (e) μ+jets:2 Hidden Layers (f) μ+jets:3 Hidden Layers

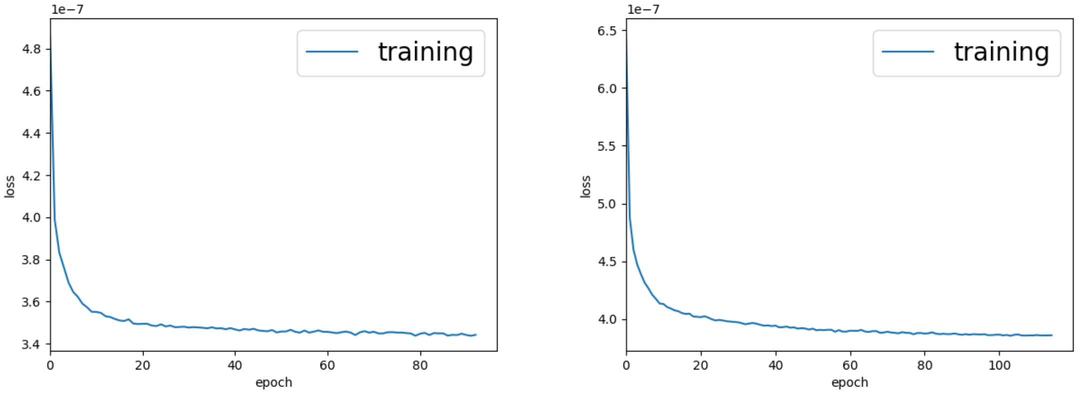
FIGURE 1.6. Normalized neural network output signal and background distribution plots are shown for both search channels for a varying number of hidden layers.



(a) e+jets Accuracy Curves

(b) μ+jets Accuracy Curves

FIGURE 1.7. Accuracy Plots for both channels for the 2 hidden layer neural network



(a) e+jets Loss Curve

(b) μ+jets Loss Curve

FIGURE 1.8. Accuracy Plots for both channels for the 2 hidden layer neural network

1.3.4. B-Tagging Working Point Studies

BTag WP selection

1.3.5. Comparison of FCNC in Decay and Production via the Neural Network

1.4. Data and Simulation Event PreSelection

1.5. Control and Validation Regions

1.6. Signal Region

CHAPTER II

ANALYSIS AND RESULTS

2.1. Uncertainties

2.2. Statistical Treatment of Results

2.3. Limit on Branching Ratio t→qγ

CHAPTER III

COMPLEMENTARY SEARCHES AND OUTLOOK

3.1. Comparison with Complementary Searches

3.2. Future Directions

HL-LHC and Beyond Future prospectives at Linear Colliders? - https://www.sciencedirect.com/

3.3. Conclusion

APPENDIX A

DERIVATION INFORMATION (TOPQ1)

Preselection is applied to both data and MC samples using the derivation framework in order to reduce the xAOD sample size. TOPQ1 derivations were used and further skimmed for the specific n-tuples used in this analysis.

APPENDIX B

COMPLETE LIST OF MONTE CARLO SAMPLES USED

The following MC samples have been used to simulate the signal and various Standard Model backgrounds at center of mass energy of 13 TeV

APPENDIX C

ADDITIONAL PLOTS FROM NN STUDIES

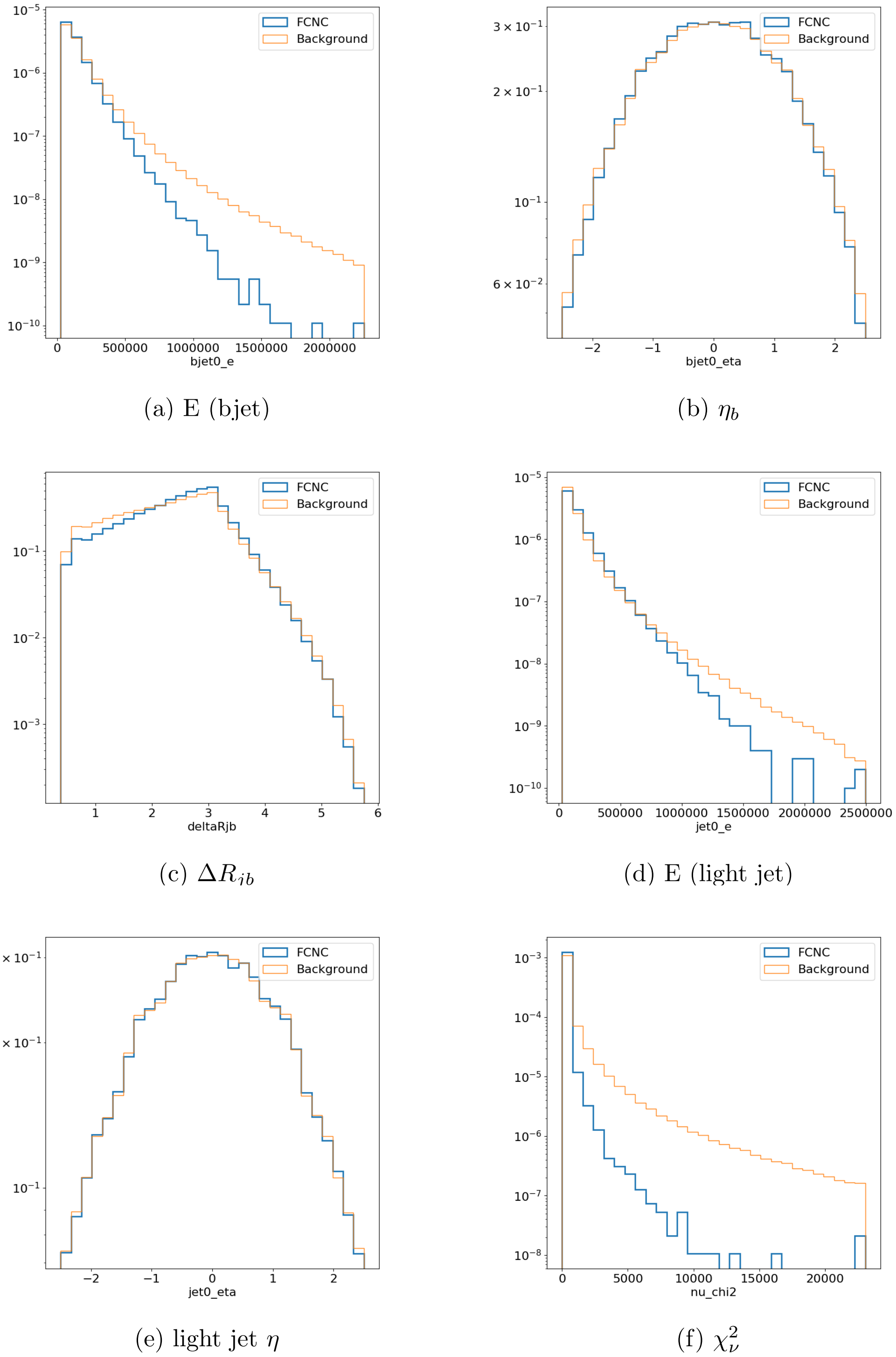
C.1. Additional Shape Comparison Plots: μ+jets channel

Various additional plots are shown in this appendix from the neural network creation and studies. Figure **??** and **??** show additional shape comparisons in variables which are not included in the final neural network model as they do not significantly change the fit values. In the cases of p_T or E variables with the higher separation value were used as there is a large correlation between the two values and the other is shown in this appendix. ΔR_{jb} was not included as the other 3 ΔR values had higher separation values and they are all related to each other as they are the geometrically related.

The neutrino reconstruction is done using a minimization of

$$\chi^2_\nu = \chi^2_{bW} + \chi^2_W$$

. All three were investigated for their separation values and the χ^2_W value had the largest separation.



(a) E (bjet)

(b) η_b

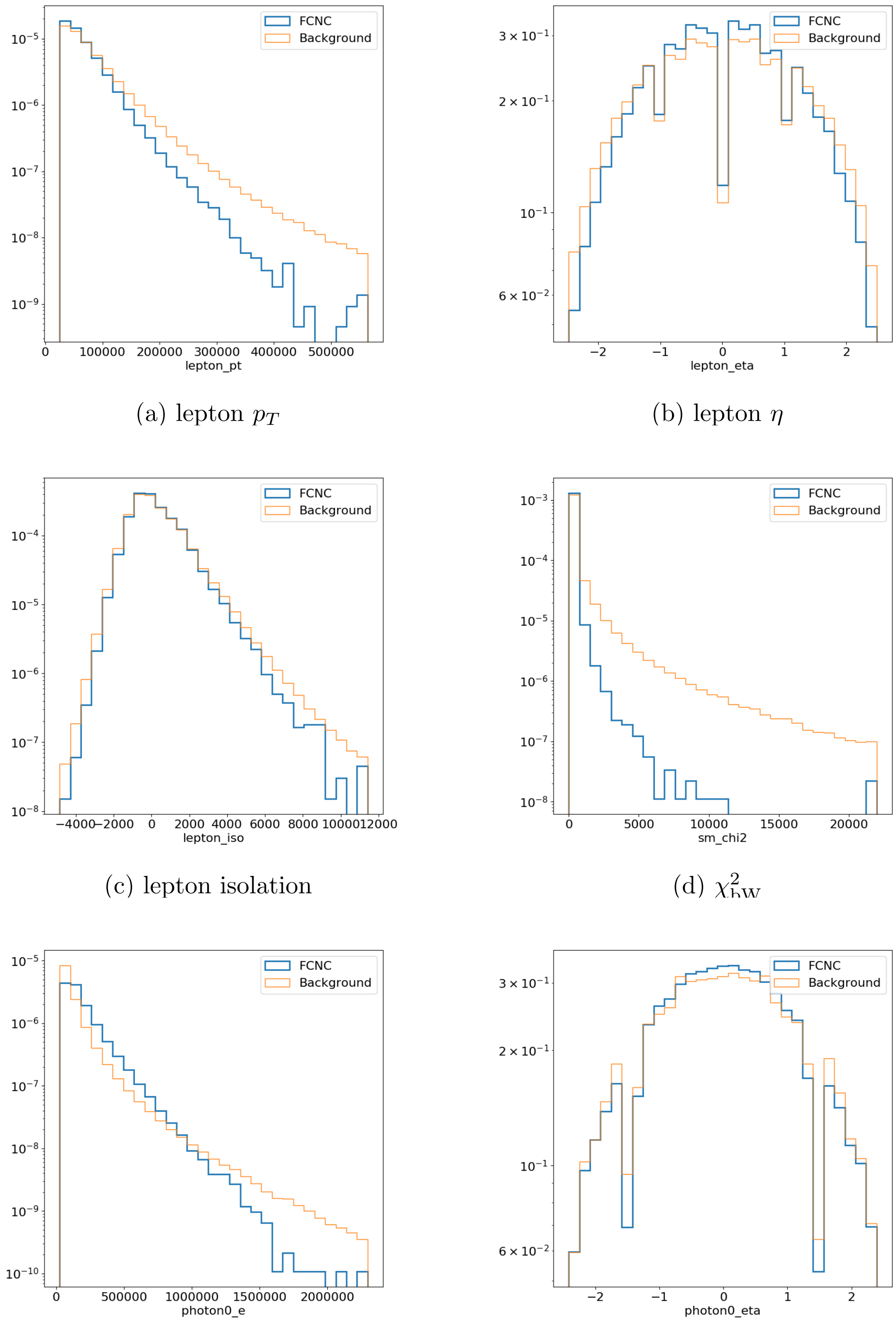
(c) ΔR_{ib}

(d) E (light jet)

(e) light jet η

(f) χ^2_ν

FIGURE C.1. Normalized variables showing the shapes of neural network input variables for the μ+jets channel: [E (bjet), η_b, ΔR_{jb} ,E (light jet), light jet η, and χ^2_ν the total χ^2 fit value



(a) lepton p_T

(b) lepton η

(c) lepton isolation

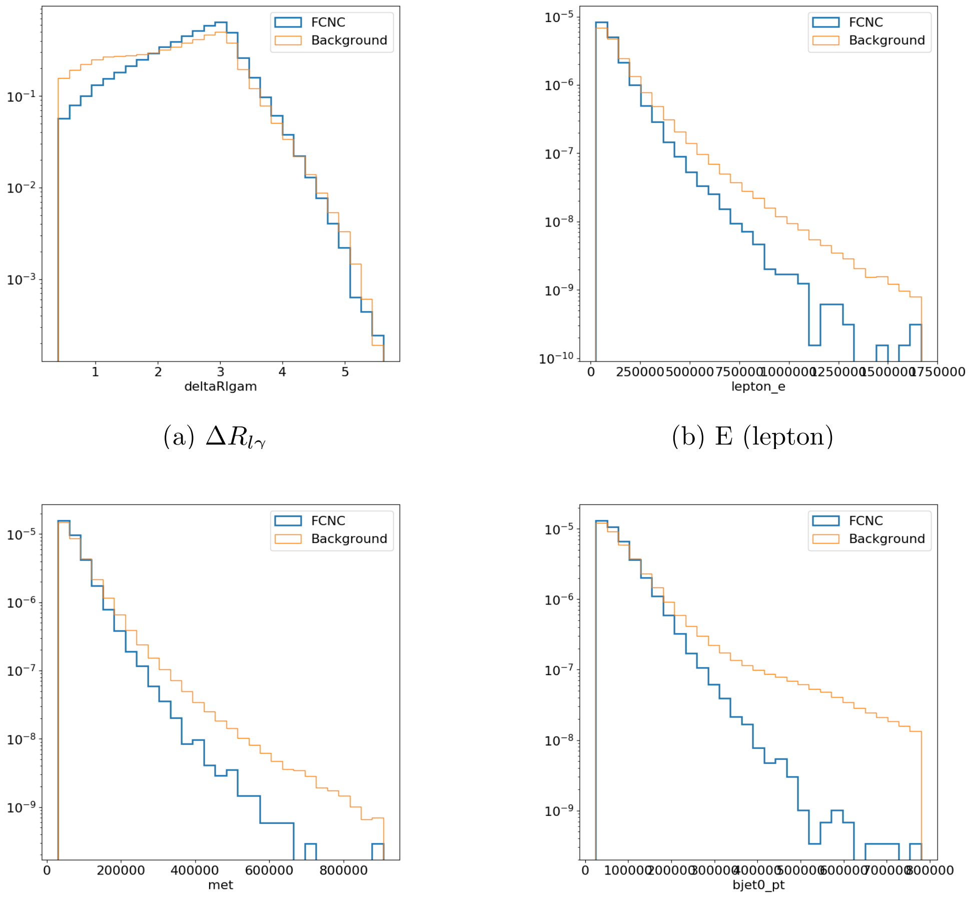
(d) χ^2_{bW}

(e) photon E

(f) photon η

FIGURE C.2. Normalized variables showing the shapes of neural network input variables for the μ+jets channel: [lepton p_T, lepton η, lepton isolation , chi^2_{SM} the bWχ^2 value from neutrino reconstrucion ,photon E, and photon η.

C.2. Shape Comparison Plots: e+jets channel



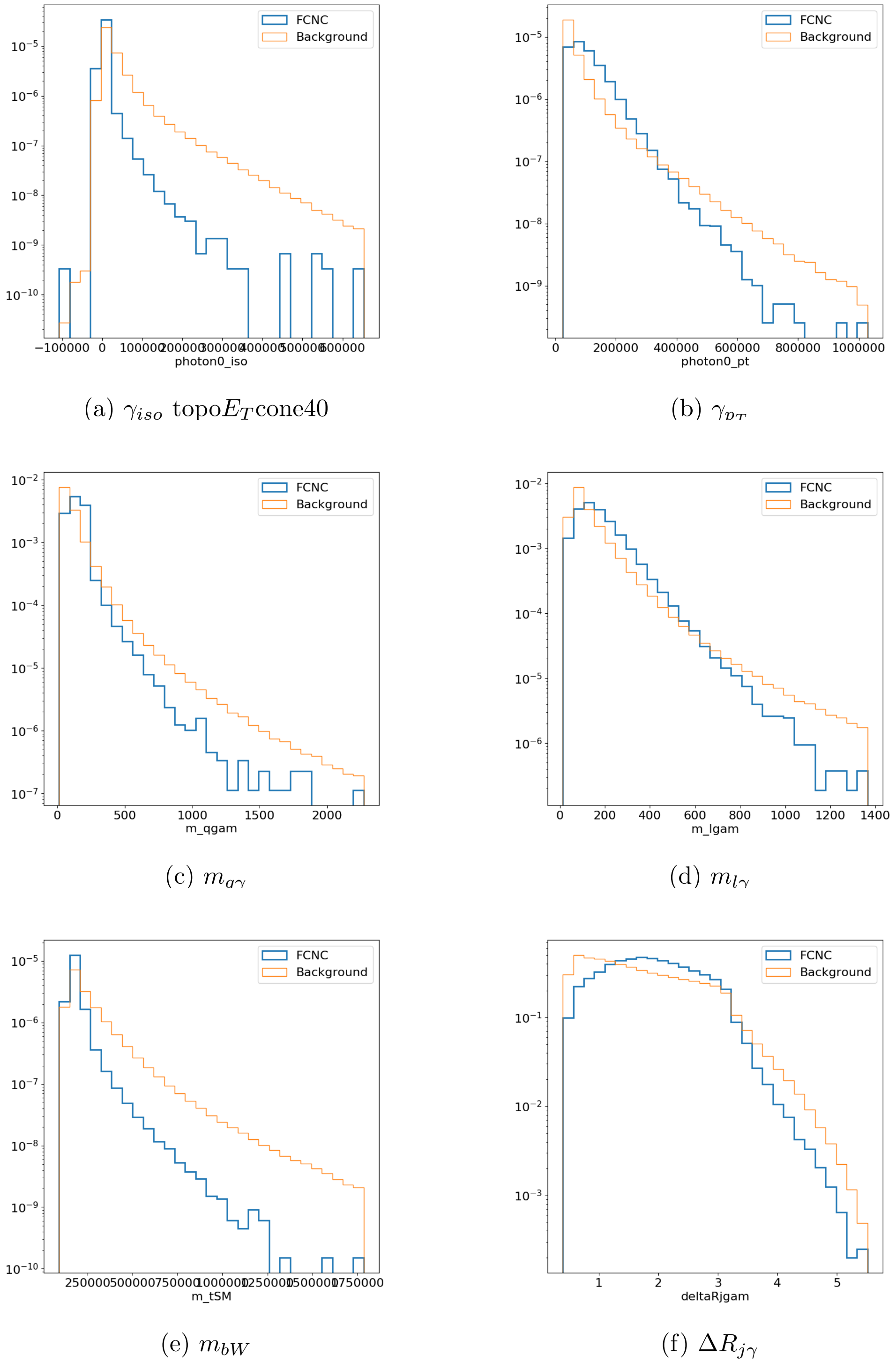
(a) $\Delta R_{l\gamma}$

(b) E (lepton)

(c) $\not{E}_T$

(d) $p_T(b)$

FIGURE C.3. Normalized variables showing the shapes of neural network input variables for the e+jets channel: $\Delta R_{l\gamma}$, E (lepton), $\not{E}_T$, and $p_T(b)$



(a) γ_{iso} topo$E_T$cone40

(b) γ_{p_T}

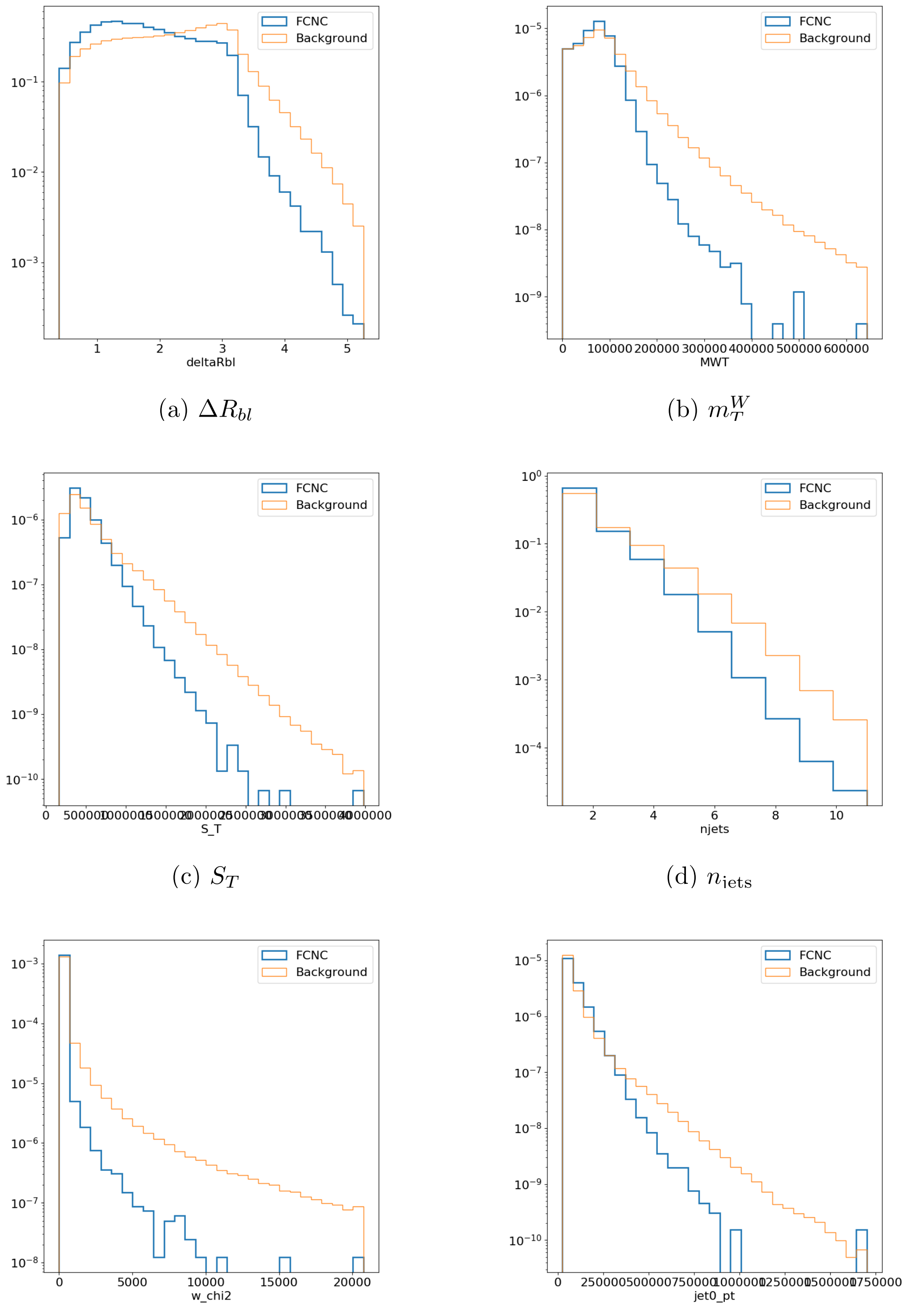
(c) $m_{q\gamma}$

(d) $m_{l\gamma}$

(e) m_{bW}

(f) $\Delta R_{j\gamma}$

FIGURE C.4. Normalized variables showing the shapes of neural network input variables for the e+jets channel: γ_{iso} topo$E_T$cone40, γ_{p_T}, $m_{q\gamma}$, $m_{l\gamma}$, m_{bW}, and $\Delta R_{j\gamma}$



(a) ΔR_{bl}

(b) m_T^W

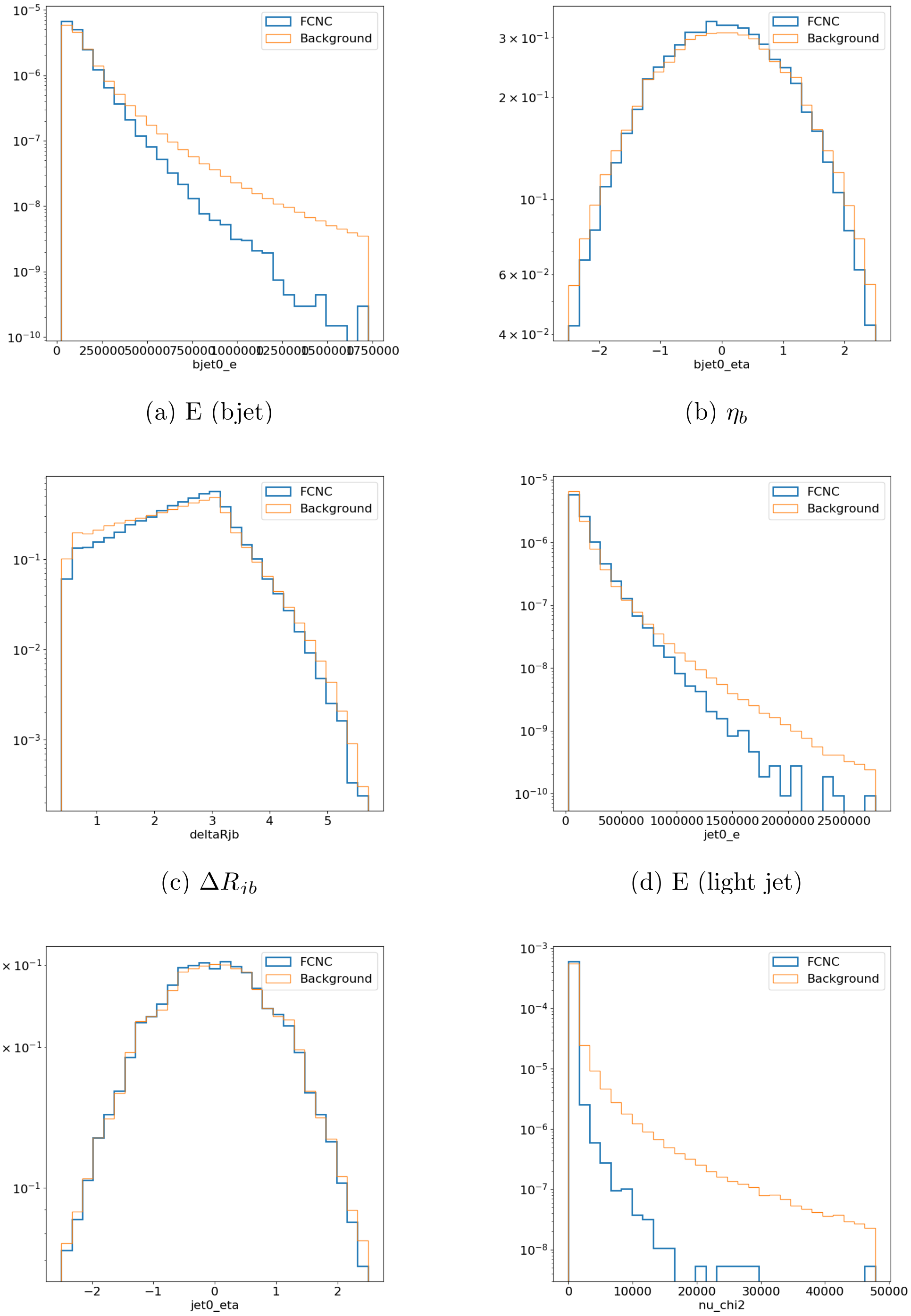
(c) S_T

(d) n_{jets}

(e) χ^2_W

(f) $p_T(q)$

FIGURE C.5. Normalized variables showing the shapes of neural network input variables for the e+jets channel: ΔR_{bl}, m_T^W , S_T, n_{jets}, χ^2_W, and $p_T(q)$



(a) E (bjet)

(b) η_b

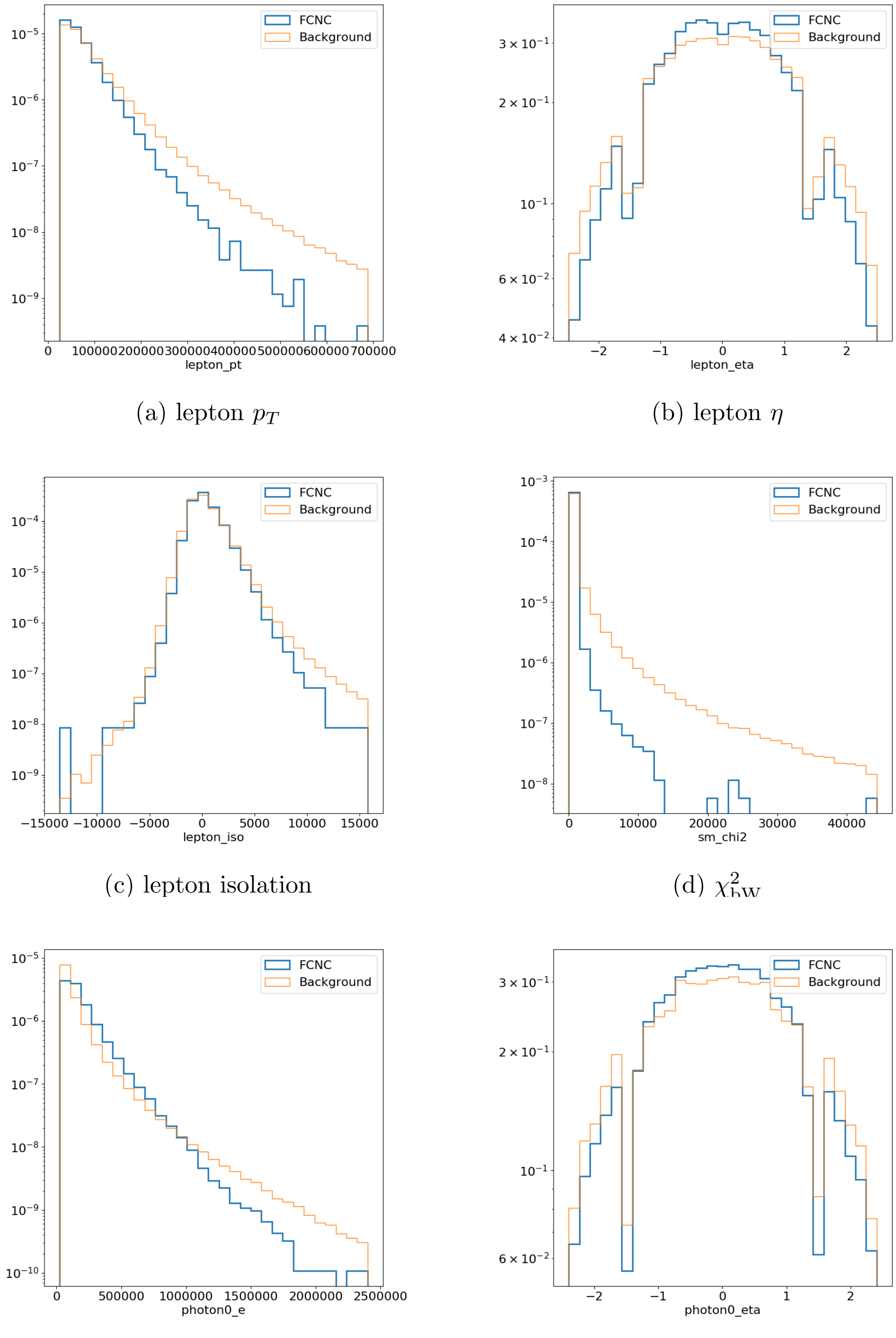
(c) ΔR_{ib}

(d) E (light jet)

(e) light jet η

(f) χ^2_ν

FIGURE C.6. Normalized variables showing the shapes of neural network input variables for the e+jets channel: [E (bjet), η_b, ΔR_{jb} ,E (light jet), light jet η, and χ^2_ν the total χ^2 fit value



(a) lepton p_T

(b) lepton η

(c) lepton isolation

(d) χ^2_{bW}

(e) photon E

(f) photon η

FIGURE C.7. Normalized variables showing the shapes of neural network input variables for the e+jets channel: [lepton p_T, lepton η, lepton isolation , chi^2_{SM} the bWχ^2 value from neutrino reconstrucion ,photon E, and photon η.

REFERENCES CITED

[1] F. Chollet et al. Keras: The python deep learning library, 2019.

[2] Martín Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. 2016.

[3] A. Karpathy. Cs231n convolutional neural networks for visual recognition, 2015.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 2014.

[5] Fabian Pedregosa et al. Scikit-learn: Machine Learning in Python. *J. Machine Learning Res.*, 12:2825–2830, 2011.