1. There has been another study proposing to improve the search for top-quark FCNCs using neural-nets from Aguilar-Saavedra and Branco in 2000 (hep-ph/0004190). This paper only considers scalar contributions and does not consider the tri-lepton OS final state, but should still be mentioned.

   This is added: An early study by Aguilar-Saavedra and Branco [hep-ph/0004190] used neural networks to search for top-quark FCNCs via scalar interactions, highlighting the potential of machine learning despite focusing on different final states.

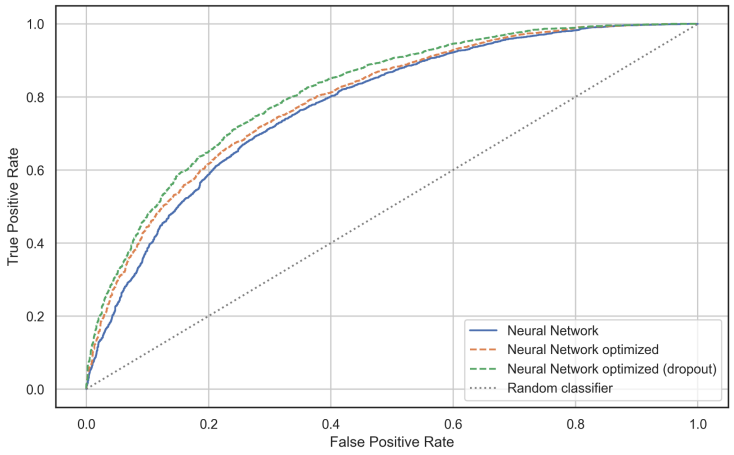
2. In section 4.1, there needs to be much more quantitative detail on the neural network architecture (number of of layers, number of neurons in each layer), the loss function used (included as an equation maybe), the optimiser chosen, the learning rate, the dropout layers and dropout rate, activation functions, etc. They should also include any data preprocessing here, e.g. re-scalings. The results should be in principle reproducible by the reader.

   This is added: The neural network used in this analysis consists of an input layer followed by three hidden layers, each containing 128 neurons. The scaled exponential linear unit (SELU) activation function is employed in all hidden layers to promote self-normalizing behavior. To mitigate overfitting, dropout regularization is applied after each hidden layer with a dropout rate of 10%. The output layer uses a sigmoid activation function to produce a binary classification score. The network is trained using the Adam optimizer with the learning rate provided by the Keras library (0.001), and the loss function is binary cross-entropy, defined as:

   \begin{equation}
   \mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \left[ y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right]
   \end{equation}

   where yiy_i and y^i\hat{y}_i denote the true and predicted labels, respectively. Prior to training, all input features are standardized using a StandardScaler transformation to zero mean and unit variance.

3. On the ROC plot, you should consider replacing 'Luck' with something like 'Random classifier', or explain what it means in the text.
   Done!




4. The accuracy and the AUC for these neural-net classifiers are not presented, they should be either in the text or in a table. The accuracy of one of the neural nets is mentioned in 4.2, but only in the discussion of the other models.
   Done in comment 5.

5. The ROC curves show good performance, but robustness is mentioned later in the paper, and one way to demonstrate robustness in neural-net performance is to train more than one classifier per task and compute the average performance +- std dev. I would recommend doing this here for the AUC and accuracy since the performances of these networks are quite close.

   ```
   \begin{table}[h]
   \centering
   \begin{tabular}{lcc}
   \hline
   \textbf{Model} & \textbf{AUC} & \textbf{Accuracy} \\
   \hline
   Baseline DNN & $0.7805 \pm 0.0069$ & $0.7310 \pm 0.0040$ \\
   DNN with Dropout & $0.8102 \pm 0.0064$ & $0.7537 \pm 0.0019$ \\
   DNN with Optimizer Tuning & $0.7958 \pm 0.0025$ & $0.7427 \pm 0.0009$ \\
   \hline
   \end{tabular}
   \caption{Performance comparison of different neural network architectures in terms of AUC and accuracy (mean $\pm$ standard deviation) across multiple runs.}
   \label{tab:nn_performance}
   \end{table}
   ```

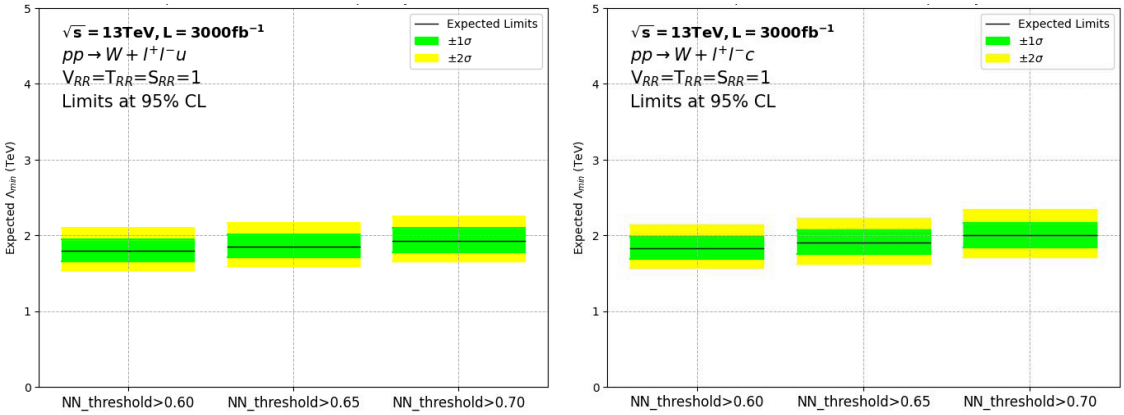
| Model | AUC | Accuracy |
|---|---|---|
| Baseline DNN | $0.7805 \pm 0.0069$ | $0.7310 \pm 0.0040$ |
| DNN with Dropout | $0.8102 \pm 0.0064$ | $0.7537 \pm 0.0019$ |
| DNN with Optimizer Tuning | $0.7958 \pm 0.0025$ | $0.7427 \pm 0.0009$ |

6. What is the motivation for using the AMS (eq. 4) to choose the working point? A well-motivated choice of metric for choosing the working-point is usually the significance improvement, as discussed in 1909.03081. You should consider this, or motivate why using AMS is more suitable.

   We acknowledge the referee's point regarding the choice of metric for optimizing the working point and the common utility of significance improvement metrics, such as those discussed in arXiv:1909.03081. In our analysis, which employs machine learning classifiers to distinguish signal from background, we selected the Approximate Median Significance (AMS) to determine the optimal threshold on the classifier's output. The AMS metric, AMS=2((TPR+FPR+br)ln(1+FPR+brTPR)−TPR), is designed to balance signal efficiency (represented by the True Positive Rate, TPR) against background contamination (related to the False Positive Rate, FPR). A key feature of the AMS metric is the inclusion of a regularization term, , which ensures stability and prevents the selection of overly narrow regions, particularly in scenarios with low background statistics that can arise after applying the stringent selections often achieved with powerful classifiers. This regularization is crucial for robustly optimizing the working point. The AMS metric was notably utilized in the context of the Higgs boson discovery analyses and provides a well-established and practical approach for maximizing the expected sensitivity in searches for new phenomena by optimizing the trade-off between retaining signal events and rejecting background. While direct optimization of significance improvement is a valid approach, AMS offers a robust and widely adopted method for defining the signal region in analyses like ours, particularly when dealing with the continuous output of a machine learning model.

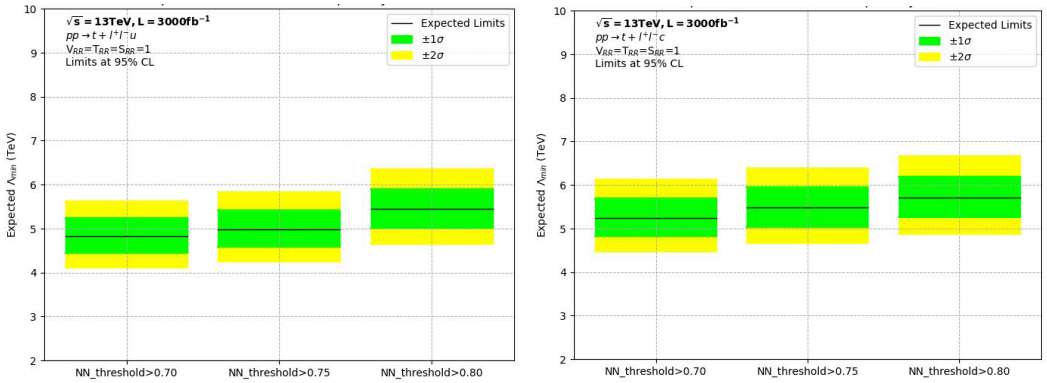
7. In addition to the previous point, different terminology for the classifier working point is used at different points in the paper, e.g. threshold and NN_weight. NN_weight (p. 12) is misleading because the network weights are what are optimised during the network training. Something consistent should be used throughout.

   the All NN_weight terms are changed to NN_threshold like the plot below for tW channe.




8. On p. 12 you say that the improvement in signal and background separation as you increase the classifier threshold/working-point gives greater sensitivity in distinguishing signal and background. But in fig 10 you show that the constraints on the NP scale for tW is relatively unaffected by moderate changes in this threshold, you should show the same plot and have a similar discussion for ttbar.

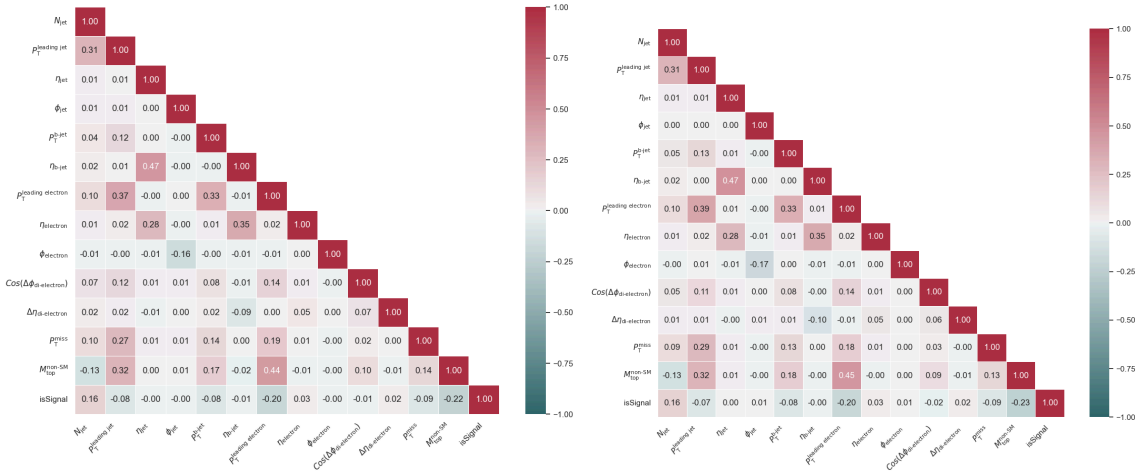
   As shown below, for both the tuee and tcee operators in the &t\bar{t}& channel, increasing the neural network (NN) threshold from 0.70 to 0.80 yields a modest but consistent improvement in the expected limits on the NP scale Λmin. While the central values increase slightly across the thresholds, the associated ±1σ and ±2σ bands remain stable, indicating improved signal-to-background separation without a significant increase in statistical uncertainty. This trend aligns with expectations: higher NN thresholds enhance purity, reducing background contamination and improving sensitivity.




9. The results for the two different NP couplings tuee and tcee are quite similar, but not exactly the same. Where do these differences arise?

   Although both u and c quark masses are significantly smaller than the top quark mass, the non-zero mass difference (m_c>m_u) leads to variations in the phase space

available for the t→qe+e− decay (where q=u,c). This can slightly alter the momentum distributions and angular correlations of the final state particles. These slight kinematic variations directly impact the input features fed into our machine learning models, as observed in the correlation heatmaps for the `tuee` and `tcee` channels below. Since the DNN models are trained independently on datasets reflecting these slightly different kinematic signatures for `tuee` and `tcee` signals, their learned decision boundaries and overall performance metrics, such as the Area Under the ROC Curve (AUC) or NP scale, can exhibit small discrepancies.



10. The benefits of the neural-net approach as opposed to the 'crude' traditional approach should be demonstrated by providing an example of the sensitivity to the NP scales obtained using the much simpler cut and count based approach on high-level observables.

    The Signal region for simple cut and count approach is defined using the most powerful variables in signal-background discrimination:

    150 GeV < M_top_non-SM < 200 GeV

    `Njet` >= 3

    And the signal and background yields are listed in the table below:

```
\begin{table}[h!]
\centering
\renewcommand{\arraystretch}{1.4}
\begin{tabular}{c||c|c|c}
\hline\hline
Process & $\mathcal{N}_S$ & $\mathcal{N}_B$ & $\mathcal{N}_S/\mathcal{N}_B$ \\
\hline\hline
```

$pp \to t\bar{t} \to ue^- e^+ \bar{t}$ & 105.3 & 98.5 & 1.07 \\
\hline
$pp \to t\bar{t} \to ce^- e^+ \bar{t}$ & 98.3  & 98.5 & 1.00 \\
\hline\hline
\end{tabular}
\caption{Cut-and-count results showing the number of signal ($\mathcal{N}_S$) and background ($\mathcal{N}_B$) events, and the signal-to-background ratio for up and charm signals.}
\end{table}

| Process | $\mathcal{N}_S$ | $\mathcal{N}_B$ | $\mathcal{N}_S/\mathcal{N}_B$ |
|---|---|---|---|
| $pp \to t\bar{t} \to ue^- e^+ \bar{t}$ | 105.3 | 98.5 | 1.07 |
| $pp \to t\bar{t} \to ce^- e^+ \bar{t}$ | 98.3 | 98.5 | 1.00 |

To benchmark the performance of our machine learning strategy, we performed a traditional cut-and-count analysis using high-level observables and a manually defined signal region. The resulting signal yields were 98.27 for the tcee operator and 105.30 for tuee, with 98.53 background events. This yields an NP scale sensitivity of approximately 3.6 TeV at 95% CL for both channels. While this demonstrates that even simple selection criteria can yield meaningful results, the deep neural network approach improves background suppression and signal discrimination, ultimately leading to stronger limits on the NP scale—surpassing the cut-and-count baseline.