We thank all the reviewers for their constructive and detailed reviews. We outlined the comments and discussed them below:

1. **Abbreviations (R1)**

We added the full name of LSTM and CNN at their first appearance in the introduction.

2. **Limitations of the work (R1, R2)**

We made it clear in the introduction that we were using previous data to evaluate the models as R1 suggested. We changed 5.3 to *performance and limitations of the LSTM-CNN model* and added a paragraph indicating that our model is designed for people who can type on a smartphone, and how learning disabilities other than PD affect deep learning models is not clear and is worth conducting future research.

3. **Related work: what is beyond previous research (R1)**

We added an explanation that Tian et al. performed a comprehensive feature analysis before applying the SVM model and deep learning eliminates the need for heavy feature engineering.

4. **How did we acquire the results and the reliability (R1, R3)**

We added details in 4.1.2 and 4.2.2 about how we performed leave-one-out cross-validation (LOOCV): for each participant *i*, we used the other 101 (in Tian et al.'s 102-participant dataset) participants' data excluding *i* to train the models and predicted the probability scores for participant *i*. In this way, the results obtained are user-independent. To address R1's comments about section 4.3, we added an explanation in 4.3.1 that data generated from gestural interactions with touchscreen also contains time series and can be transferred to a binary classification problem on time series data. We also clarified that we performed LOOCV for four common gestures individually.

5. ***tanh* as the activation function in convolutional layers (R1)**

We kept it consistent with the default activation function used in LSTM (keras API). Because we did not suffer the vanishing gradient problem in these tasks, we are able to use tanh as the activation in CNN layers without problems in training.

6. **Authors claim that LSTM-CNN outperforms the rest of the models (R1, R2)**

In sections 4.2 and 4.3, we claimed the LSTM-CNN model performed better only in terms of F1-score and specificity, but not AUC. We rephrased section 5.1 of discussing the results as R1 commented. We thank R1 and R3 for pointing out this and we agree other experiments need to be conducted in order to make further claims.

7. **Why the use of CNN generates better results (R1)**

We explained in section 3.2 that CNN is capable of capturing distinctive features from input that has spatial relations. Stacked layers of CNN can extract discriminative feature maps in a hierarchical manner. One possible reason that using CNN in our model leads to better results is that it further extracts distinctive features from LSTM outputs.

8. **The motivation behind the LSTM-CNN model and differences between the LSTM-CNN model and the CNN-LSTM model (R1, R2)**

We defined the other three models in sections 3.3-3.5 in the text and explained the differences between each of them and the LSTM-CNN model. The main difference between the LSTM-CNN model and the CNN-LSTM model is that the CNN-LSTM model applied time-distributed CNN on the input data, which requires the inputs to have a spatial structure in their dimensions. However, it isn't clear that typing sequences have this feature. We agree that making the motivation clear is important and we explained the motivation in section 3.2.

9. **We fixed typos and incorrect numerical values in the text and tables.**