PATTER, Volume *1*

Supplemental Information

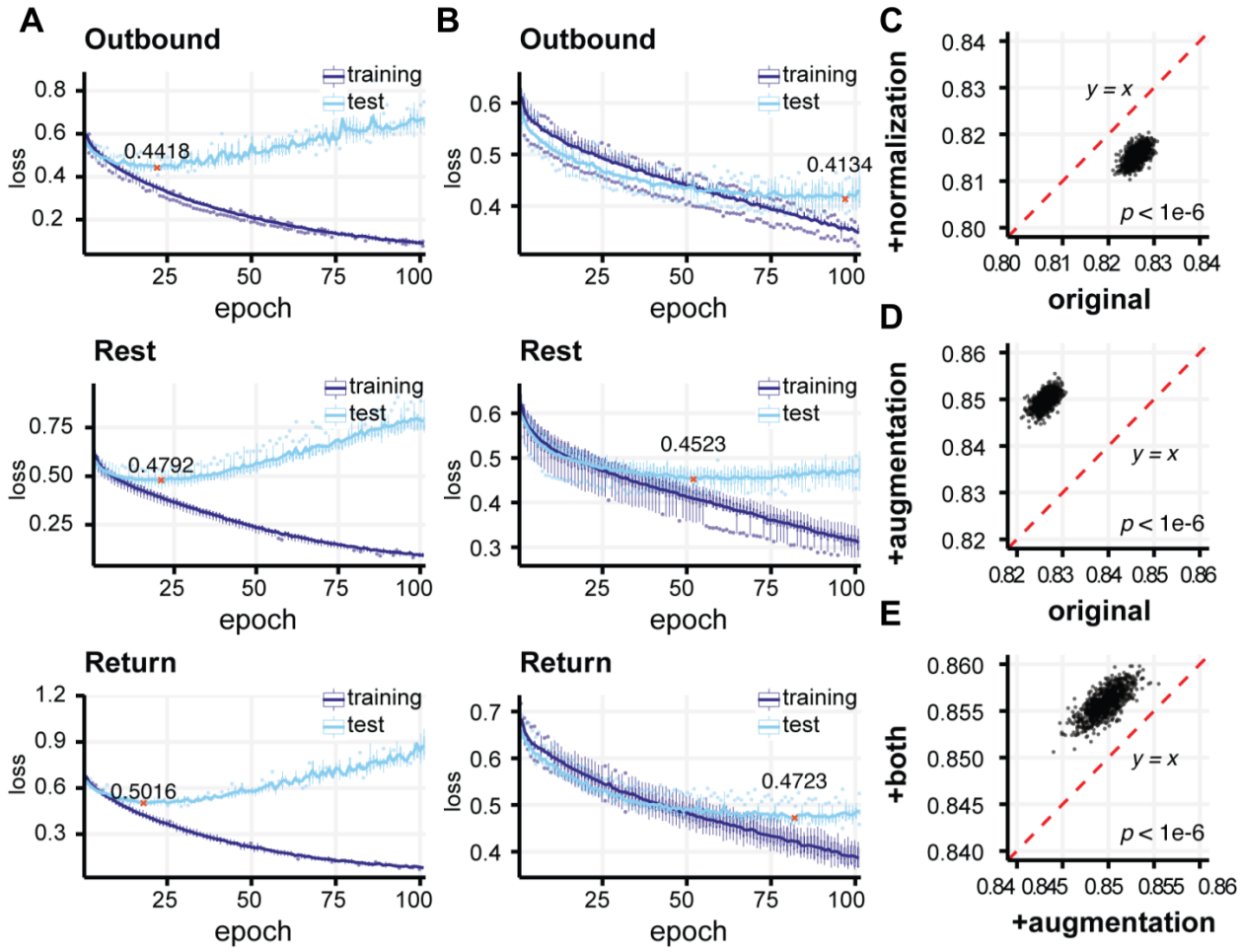
Deep Learning Identifies Digital Biomarkers for Self-Reported Parkinson's Disease

Hanrui Zhang, Kaiwen Deng, Hongyang Li, Roger L. Albin, and Yuanfang Guan

Supplementary Information

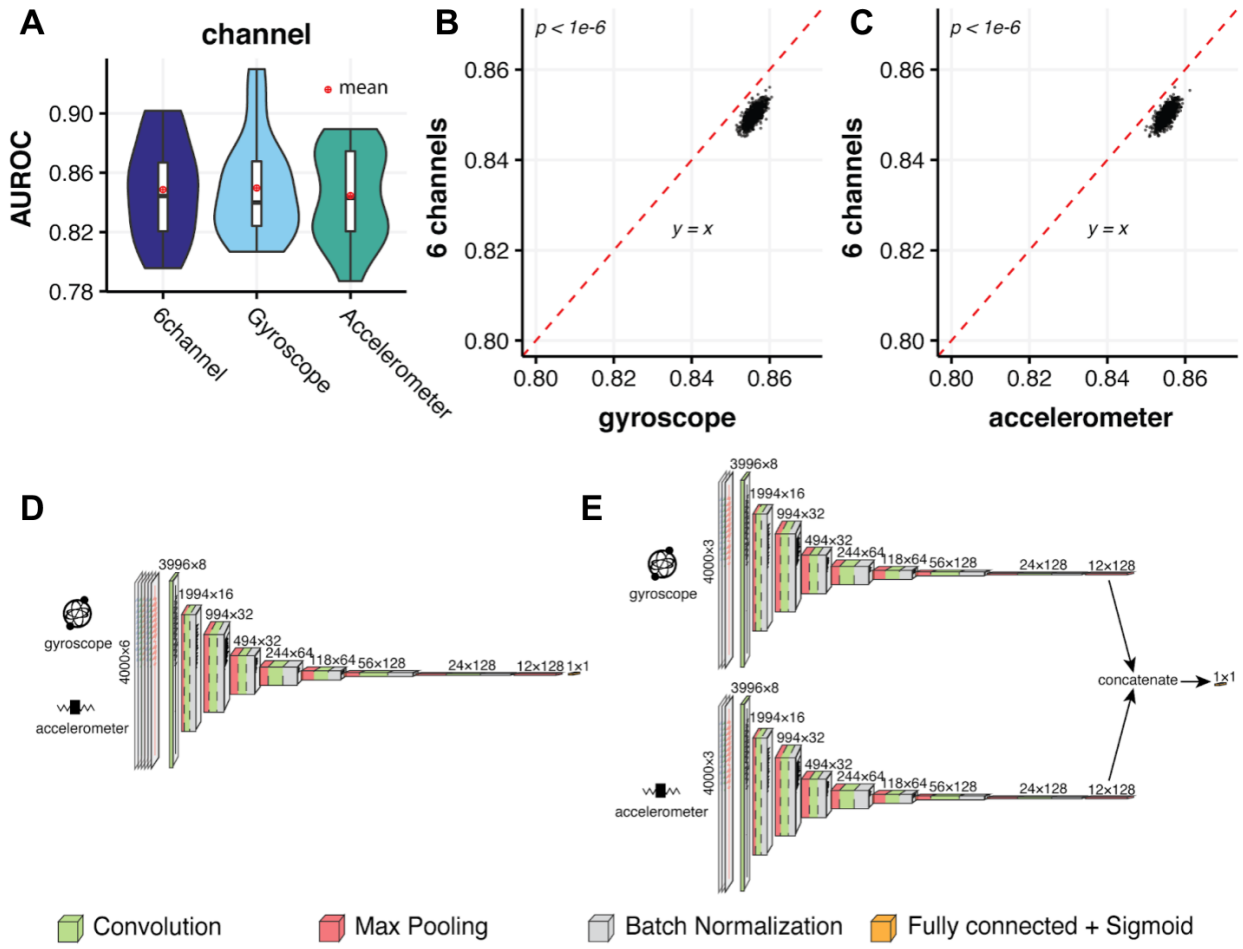
1. **Supplemental Figures:**

Figure S1. Training with only normalization and augmentation and comparison of model performance with normalization and augmentation than original.



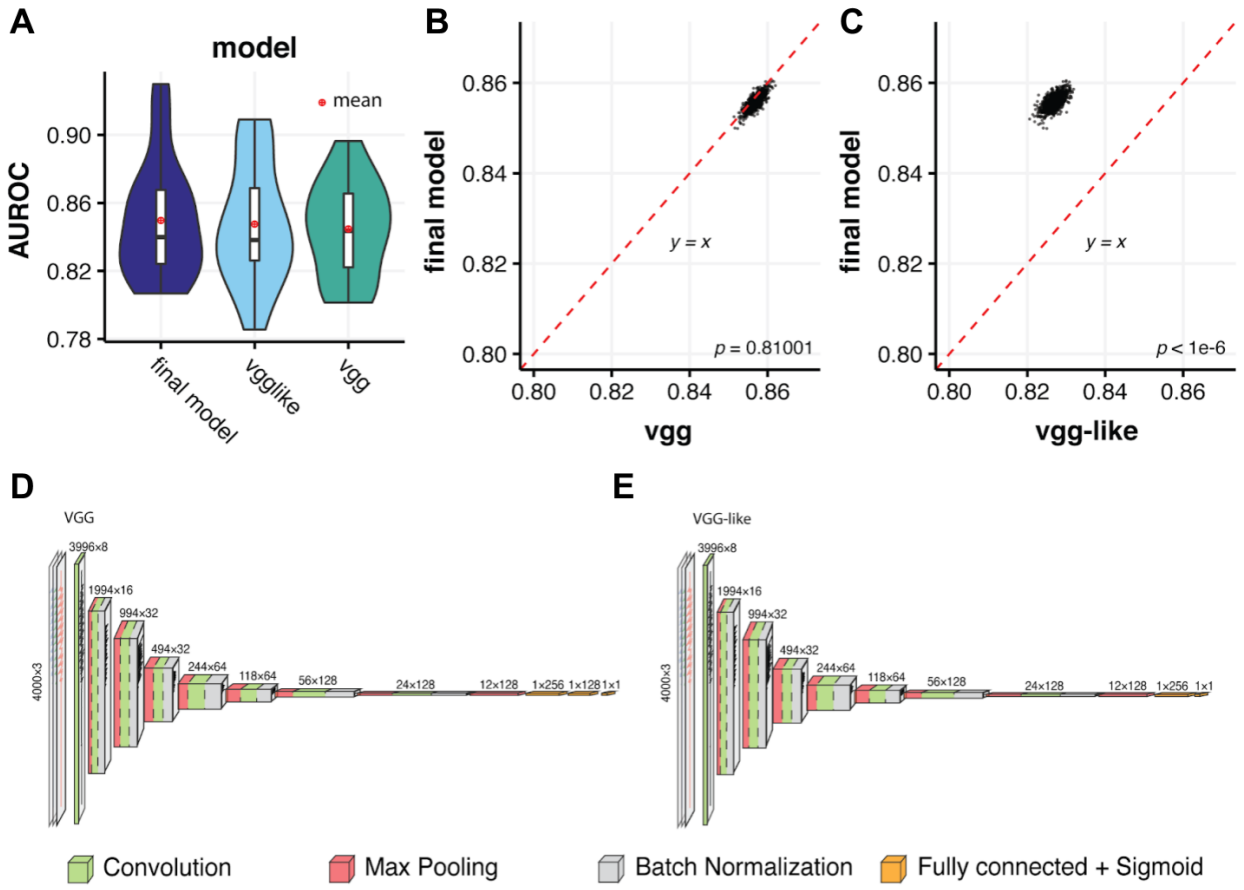
(A) and (**B)** show the dynamics of training and testing loss during 100 epochs of training process for models applying only normalization and data augmentation. Models achieved the lowest test loss were obtained to avoid underfitting and overfitting to the training set. Lowest test loss achieved were denoted and marked by red crosses. **(C)** and **(D)** show the pairwise comparison of AUROCs between models using raw signal (neither normalization and augmentation) and with only normalization/only augmentation during bootstrapping. **(E)** shows the bootstrapped AUROCs between applying only augmentation and applying both augmentation and normalization. **(A).** Training and test loss within 100 epochs during training of models with only normalization. **(B).** Training and test loss within 100 epochs during training of models with only augmentation. **(C).** Pairwise comparison between models using raw (original) and normalized walking records (+normalization). **(D).** Pairwise comparison between models using raw (original) and augmented walking records (+augmentation). **(E).** Pairwise comparison between models using augmented walking records (+augmentation) and using both normalized and augmented records (+both).

Figure S2. Comparison of training with 6 channels (with both gyroscope and accelerometer) and either gyroscope/accelerometer alone.



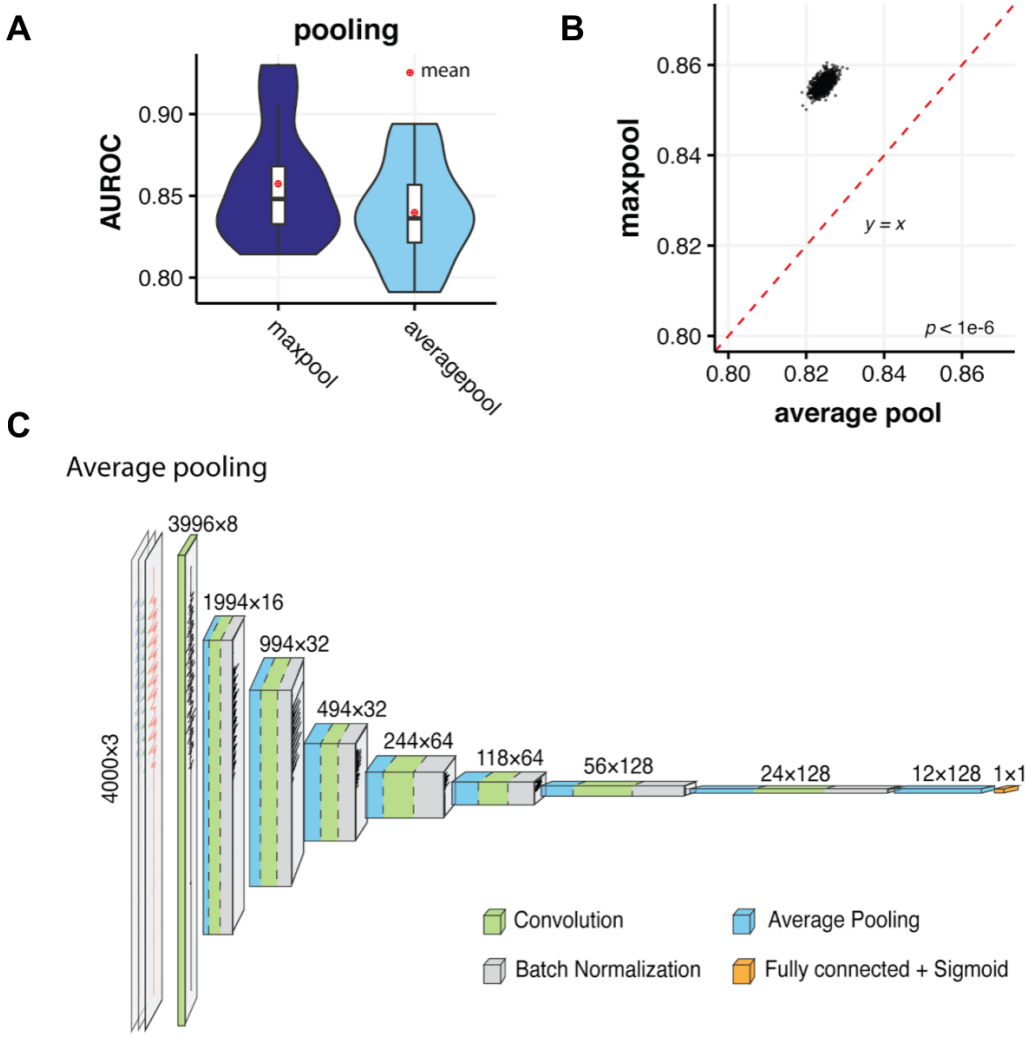
(A). Comparisons of AUROCs using both gyroscope and accelerometer signals as input (6-channel) and using either signal alone. **(B).** Paired AUROC value comparison between using 6-channel and gyroscope signal as input (mean[SD], 0.8499[0.0017] vs. 0.8558[0.0015]). **(C).** Paired AUROC value comparison between using 6-channel and accelerometer signals as input(mean[SD], 0.8499[0.0017] vs. 0.8552[0.0015]). No significant improvement was observed when using 6-channel input. **(D).** Demonstration of model with 6 input channels of accelerometer and gyroscope signals. **(E).** Demonstration of model with both accelerometer and gyroscope as input and concatenate at the last later. This model performs equally to using 6-channel input.

Figure S3. Comparison of performance of different vgg-like models.



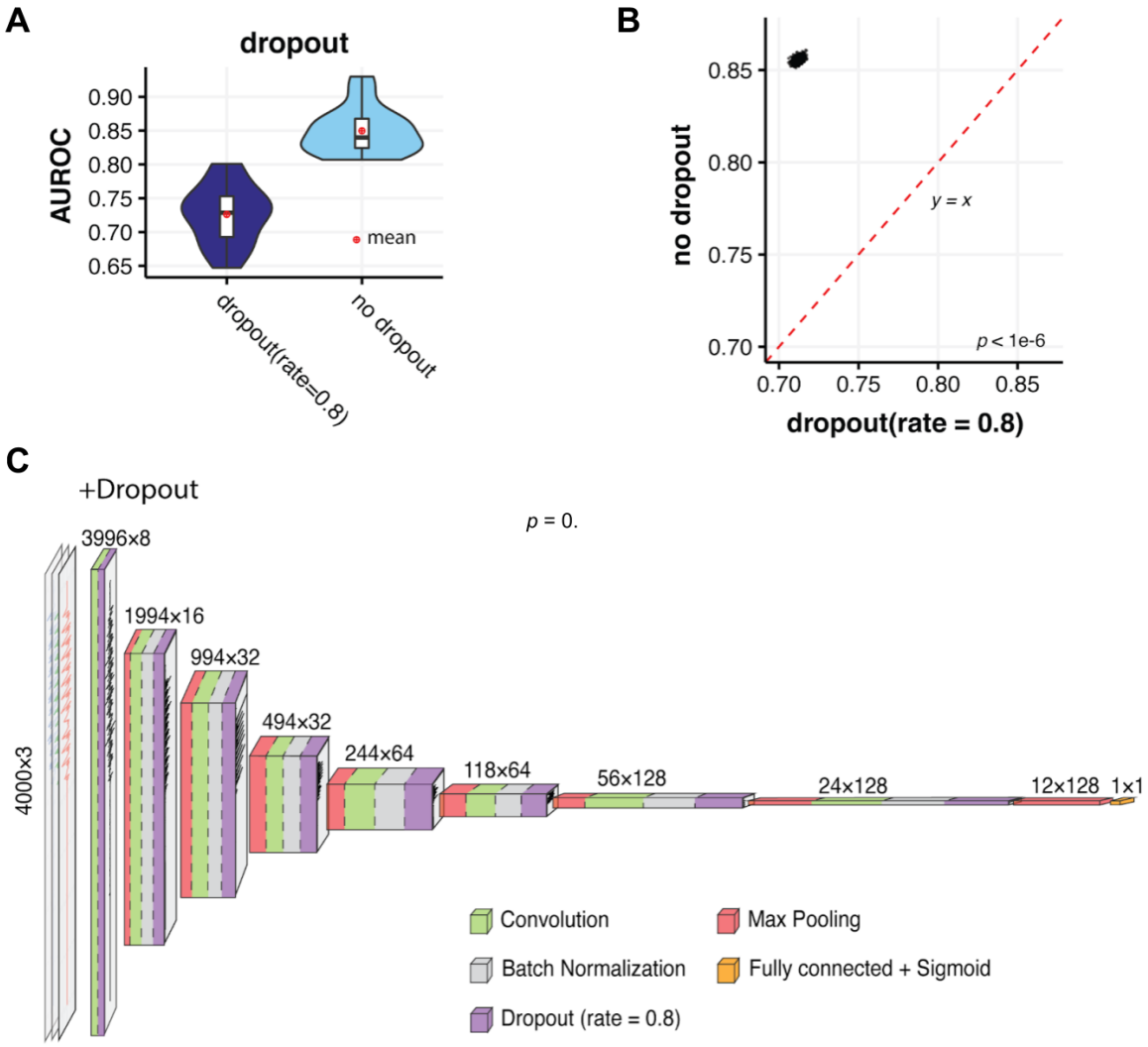
(A). Comparison of AUROCs of our final model and two vgg models we tested in this study. **(B).** Paired AUROC value comparison between our final model and vgg 16 model. No substantial difference was observed between two models (mean[SD], 0.8567[0.0016] vs. 0.8558[0.0015], p-value =0.81001), while our final model requires less training time as it contains fewer layers. **(C).** Paired AUROCs value comparison between our final model and vgg-like model. Our final model consistently performed better than the vgg-like model (mean[SD], 0.8558[0.0015] vs. 0.8268[0.0017], p-value <1e-6). **(D).** Demonstration of VGG model (with three dense layers). **(E).** Demonstration of VGG-like model (with two dense layers)

Figure S4. Comparison of performance of maximum and average pooling.



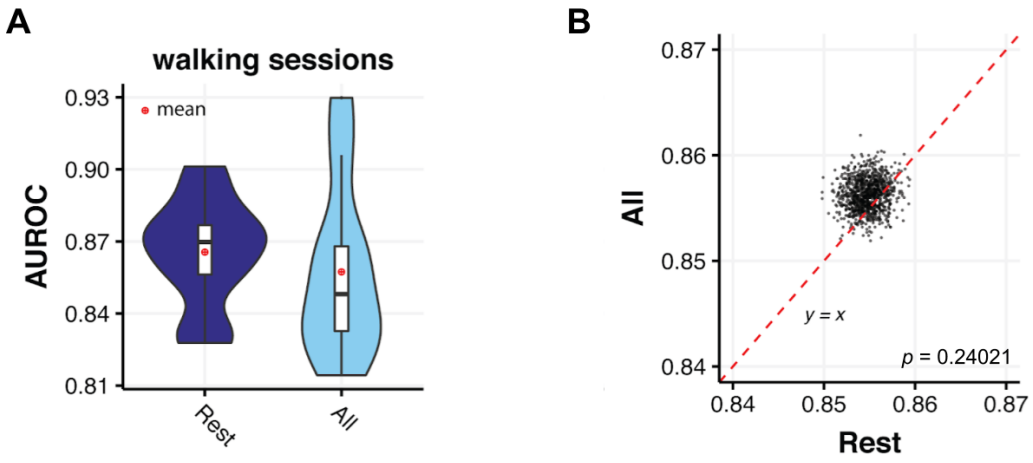
(A). Comparisons of AUROCs using max pooling layers (our final model) and average pooling layers in CNN model. **(B).** Paired AUROC value comparison between using max pooling and average pooling layers. Max pooling consistently performed better than average pooling (mean[SD], 0.8558[0.0015] vs. 0.8244[0.0016], p-value <1e-6). **(C).** Demonstration of model that replaces max pooling with mean pooling layers.

Figure S5. Comparison of performance of adding/no dropout.



(A). Comparisons of AUROCs adding dropout layers and no dropout layers. **(B).** Paired AUROC value comparison between using no dropout and after adding dropout layers. Adding dropout doesn't show significant improvement in model performance (mean[SD], 0.8558[0.0015] vs. 0.7120[0.0019], p-value <1e-6). **(C).** Demonstration of model that adds dropout layers (rate = 0.8).

Figure S6. Comparison of performance of models using quiet standing records alone and all records.



(A). Comparisons of AUROCs of models using only quiet standing (Rest) records and using all records (outbound walking, quiet standing and return walking) in 5-fold cross validation. **(B).** Paired AUROC value comparison between using quiet standing (Rest) records and using all records (outbound walking, quiet standing and return walking). No substantial difference was between using only quiet standing records and all records (mean[SD], 0.8558[0.0015] vs. 0.8548[0.0016], p-value = 0.24021).

Figure S7. Ten examples of original records and saliency maps of PD patients during the Outbound session.

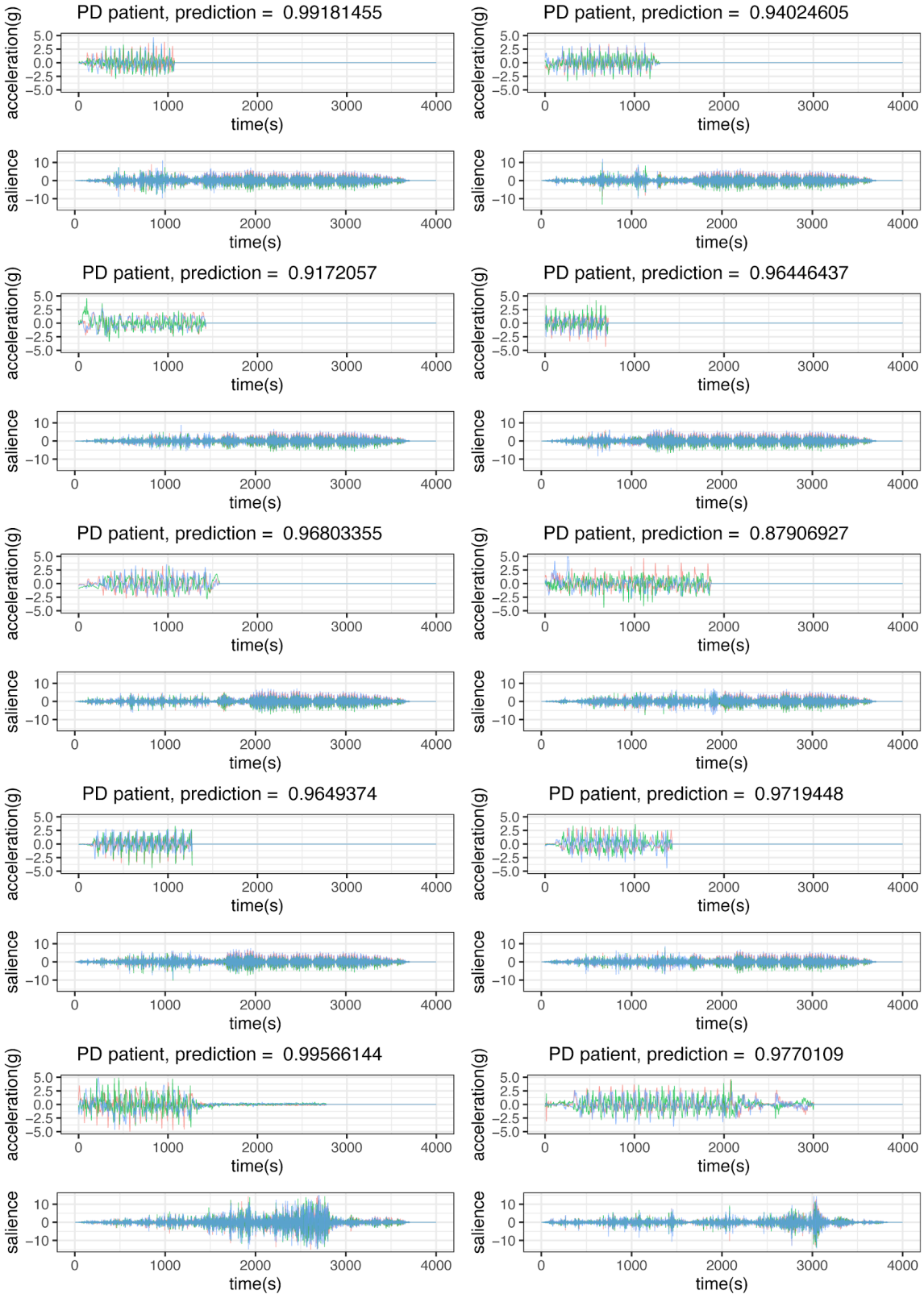


Figure S8. Ten examples of original records and saliency maps of healthy individuals during the Outbound session.

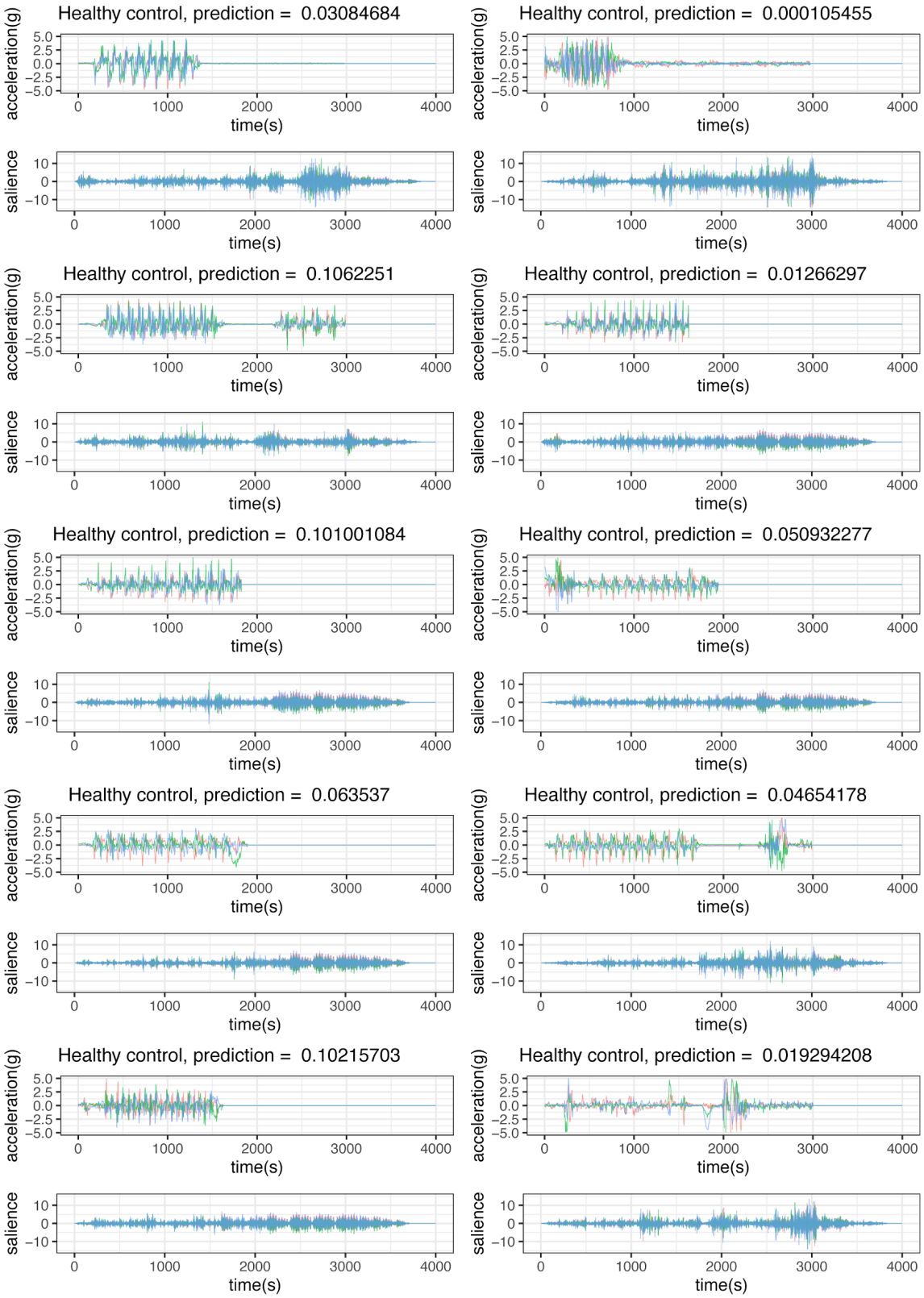


Figure S9. Ten examples of original records and saliency maps of PD patients during the Rest session.

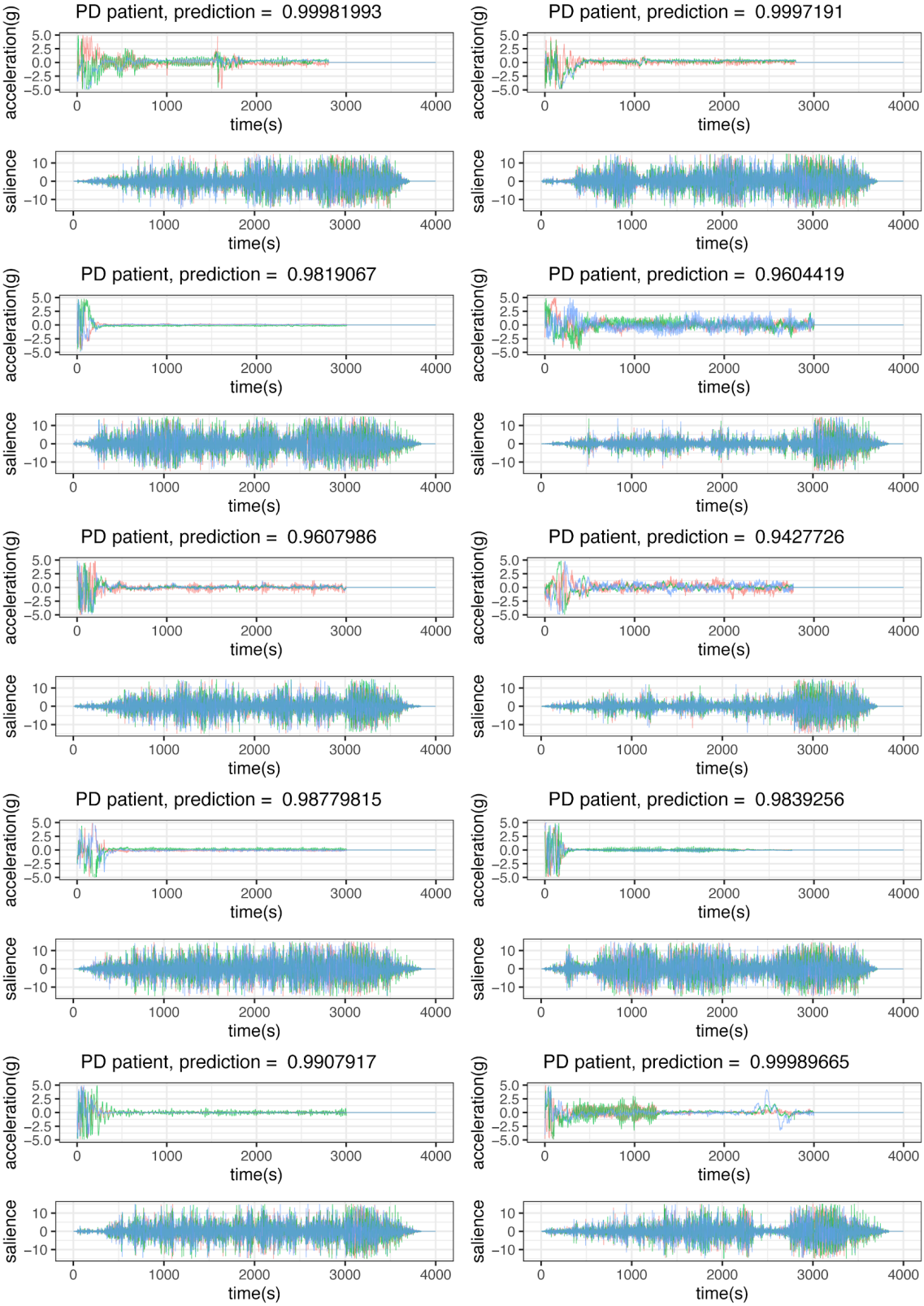


Figure S10. Ten examples of original records and saliency maps of healthy individuals during the Rest session.

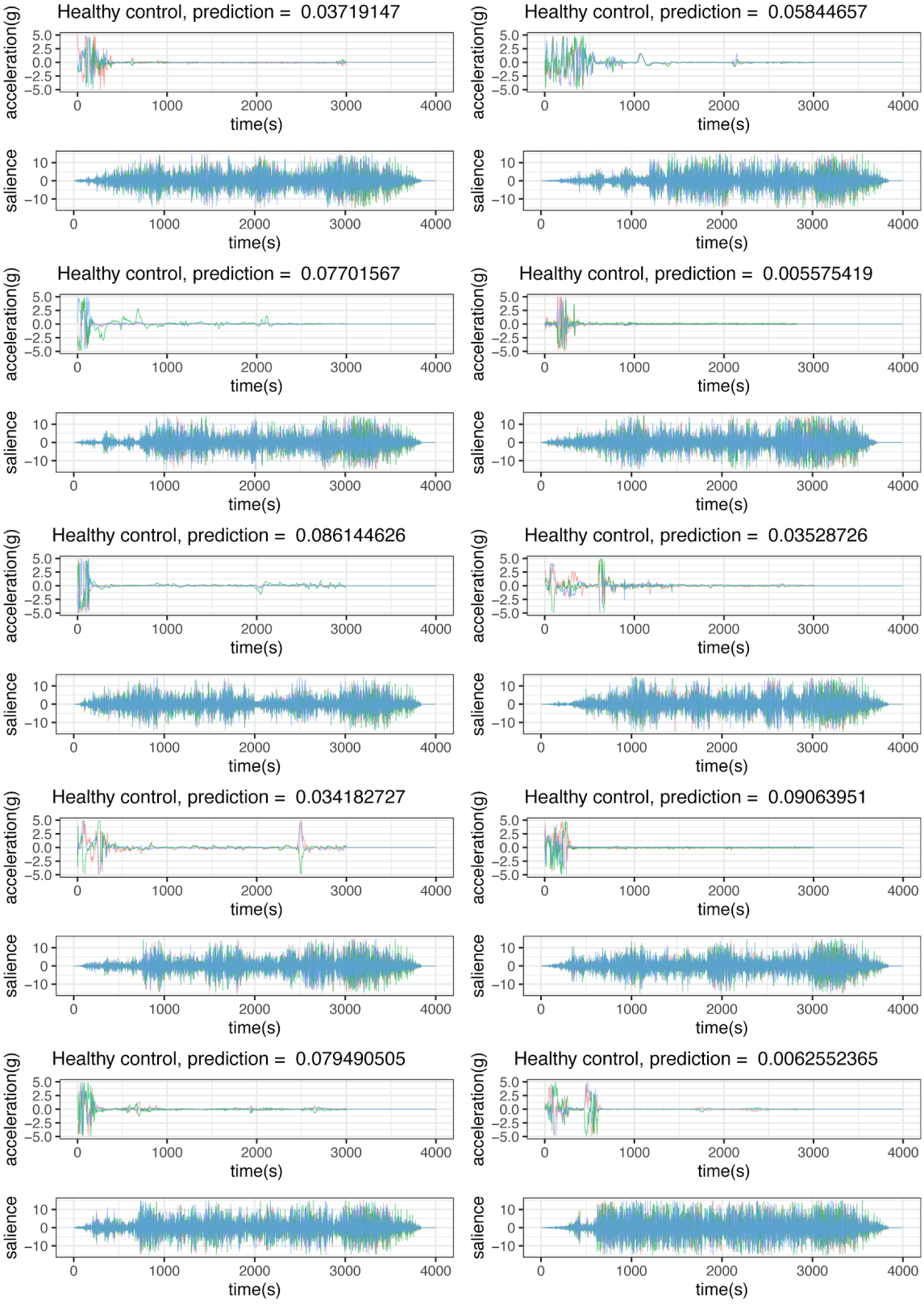


Figure S11. Ten examples of original records and saliency maps of PD patients during the Return session.

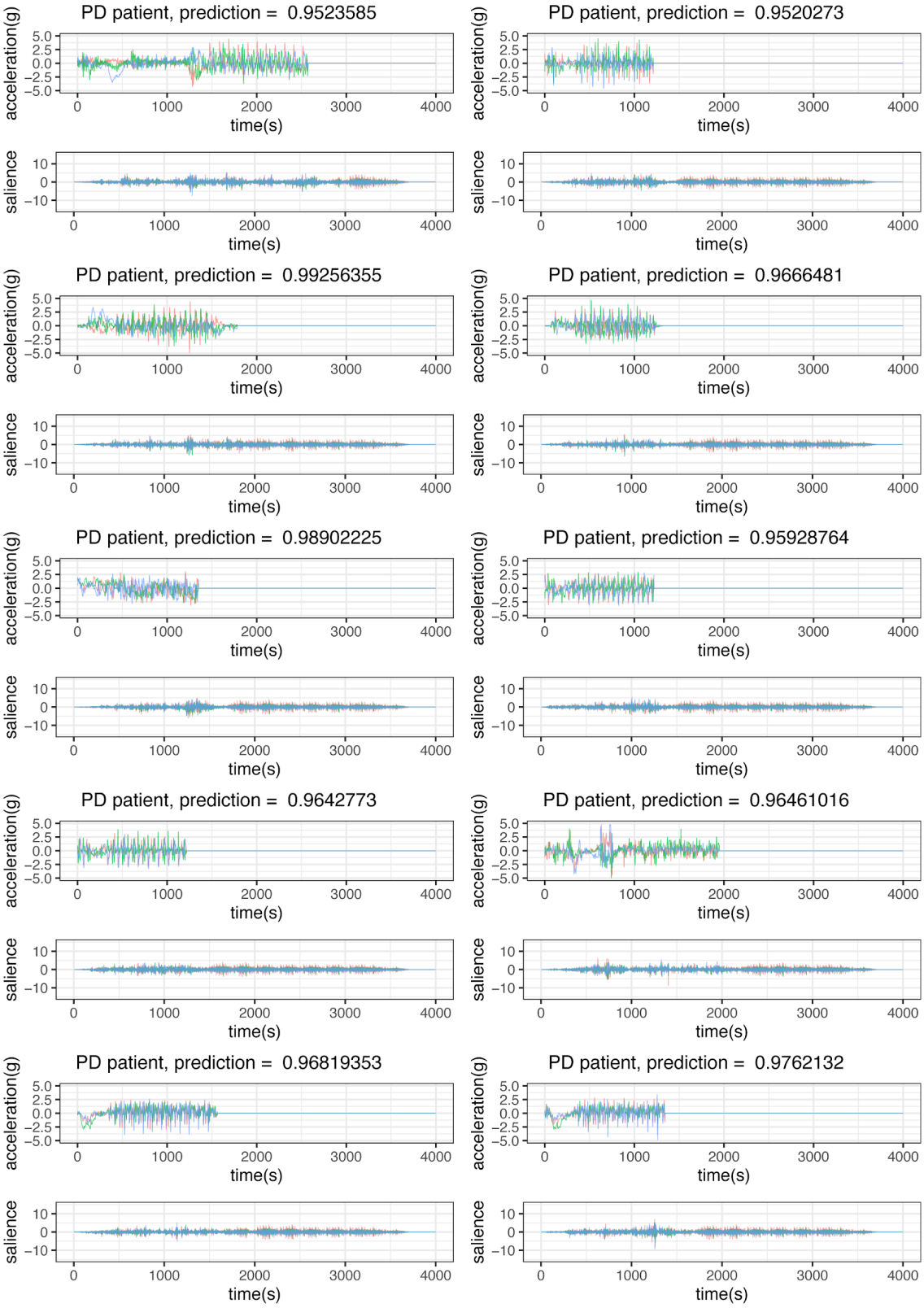
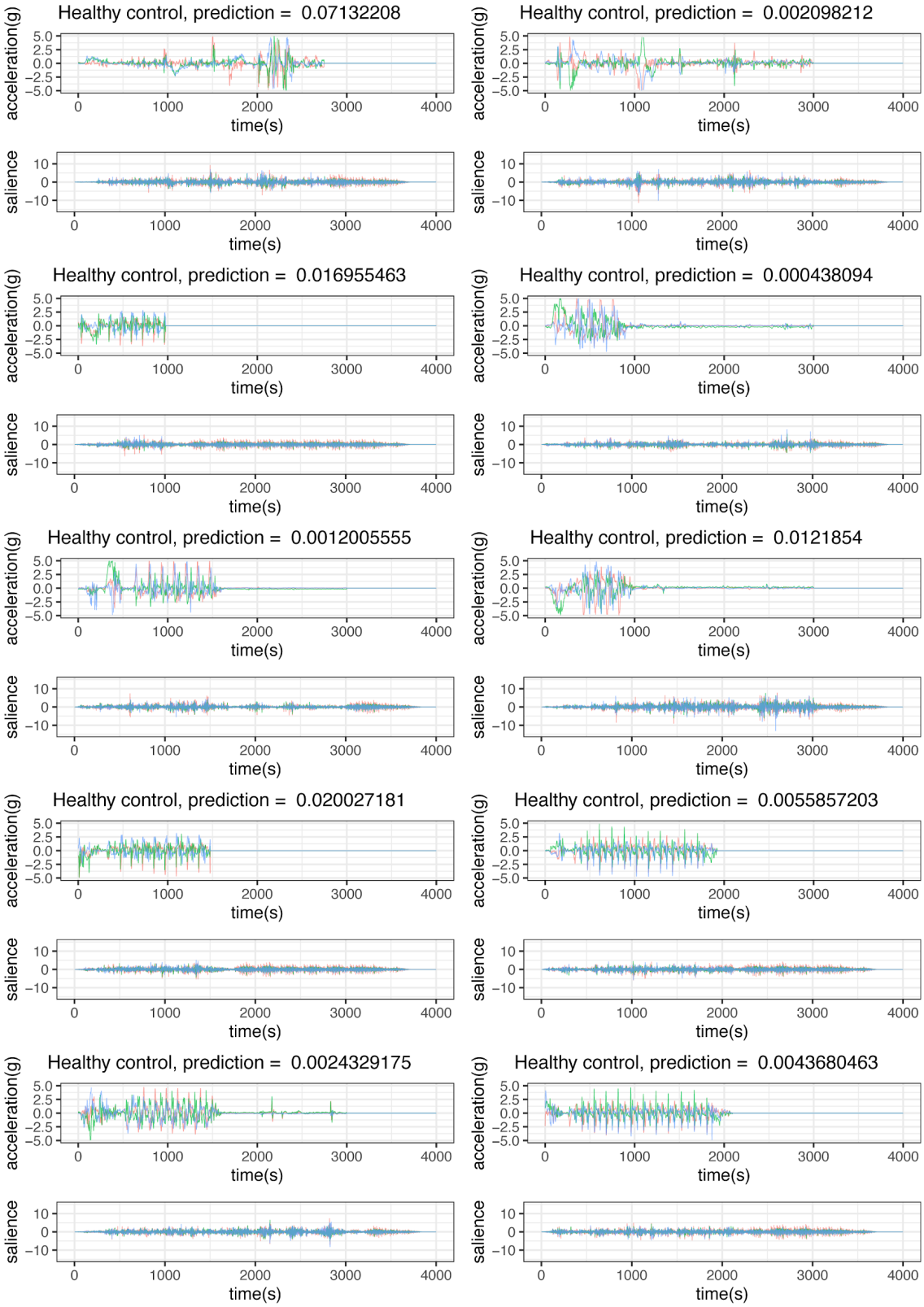


Figure S12. Ten examples of original records and saliency maps of healthy individuals during the Return session.



2. Supplemental Experimental Procedures

2.1 Cross-validation by Separating Individuals

Since walking and quiet standing records of the same person often show similar patterns, randomly dividing the data at the record level into training and testing sets may lead to overfitting and over-estimation of model performance. Thus, in the 5-fold cross-validation, we divided the training and testing set by individuals. Because the training and testing are done at the record level, we further mapped each record to the individual. The evaluation of the performance was the Area Under the Receiver Operating curve (AUROC) for classifying PD patients.

2.2 Nested Training for Calling Back Optimal Parameters

To simplify the training process, we zero-padded all matrices to 3×4000. For each cross-validation, the training samples from walking and quiet standing were randomly divided into the training set (50%) and validation set (50%), respectively. The best model generated through the epochs was called back by the validation set. This process was repeated by reseeding the training and validation set for five times separately for walking and quiet standing, generating five models for each. The training records were then resampled to balance the positives (with PD) and negatives (without PD) by bootstrap resampling. We trained the models for 50 epochs, equivalent to reading through approximately 750,000 samples during training, using Adam optimization and an initial learning rate of 0.0005. Relu activation was used in all intermediate convolution layers, and sigmoid is used for the last layer.

2.3 Quantile normalization of walking records

Before being fed into the feedforward neural network, the walking records were normalized by axis-wise quantile. For the original padded record with axis x, y and z, the original record *R* is:

$$R=\begin{bmatrix} X \\ Y \\ Z \end{bmatrix},\ where \begin{cases} X=\begin{bmatrix} x_1 & \cdots & x_{4000} \end{bmatrix} \\ Y=\begin{bmatrix} y_1 & \cdots & y_{4000} \end{bmatrix} \\ Z=\begin{bmatrix} z_1 & \cdots & z_{4000} \end{bmatrix} \end{cases}$$

The normalized record *R'* will be generated by quantile, which is adjusted to the average and then divided by the standard deviation:

$$R'=\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix},\ where \begin{cases} X' = \dfrac{X-\overline{X}}{sd(X)} \\ Y' = \dfrac{Y-\overline{Y}}{sd(Y)} \\ Z' = \dfrac{Z-\overline{Z}}{sd(Z)} \end{cases}$$

2.4 Loss-included Data Augmentation by time-series and magnitude rescaling

To simulate the perturbation on speed or range of movement by different individuals in a real-world situation, we randomly rescaled the original record by 0.8-1.2 by time series fold using Python *OpenCV* [1], and then padded/cropped to the original size. The time series rescaling might lose part of the information due to cropping.

2.5 Loss-free Data Augmentation by Random Rotation

To simulate the records in different reference frames, we rotated the original signal reference frames by random angles based on Euler's theorem. Each time, we seeded three random numbers *i, j,* and *k* between 0 to 1, and then defining a normalized *axis* = *(i',j',k')* by:

$$i' = \frac{i}{\sqrt{i^2 + j^2 + k^2}}$$

$$j' = \frac{j}{\sqrt{i^2 + j^2 + k^2}}$$

$$k' = \frac{k}{\sqrt{i^2 + j^2 + k^2}}$$

Next, we seeded a randomized angle θ between 0 to 2π:

$$a = \cos(\theta/2),$$
$$b,\ c,\ d = (i', j', k')\sin(\theta/2)$$

Then, we generated the rotation matrix:

$$\begin{bmatrix} aa+bb-cc-dd & 2(bc+ad) & 2(bd-ac) \\ 2(bc-ad) & aa+cc-bb-dd & 2(cd+ab) \\ 2(bd+ac) & 2(cd-ab) & aa+dd-bb-cc \end{bmatrix}$$

The above rotation matrix represented the difference between a new reference frame and the reference frame of the phone, which allowed us to sample the reference frames at all possible orientations. By multiplying the rotation matrix to the original record *R* of 3×4000, we could produce a new record of the same size but under a different reference frame:

$$R_{new} = \begin{bmatrix} aa+bb-cc-dd & 2(bc+ad) & 2(bd-ac) \\ 2(bc-ad) & aa+cc-bb-dd & 2(cd+ab) \\ 2(bd+ac) & 2(cd-ab) & aa+dd-bb-cc \end{bmatrix} \times R$$

2.6 Calculation of AUROC and Significance Tests for Comparing Models

The Area Under the Receiver Operating Characteristic curve (AUROC) is a measurement of the accuracy of binary classifiers [2]. It is calculated by plotting the true positive rate (TPR) against the false positive rate (FPR) at various thresholds and calculating the accumulated area under the curve. We used *sklearn.metrics* module in Python to calculate the AUROCs of the five-fold cross validation and the bootstrapping significance tests.

The five-fold cross-validation also allowed us to carry out bootstrapping to estimate the *p-value*s of differences between models. The bootstrapping was carried out by resampling the predictions on the subjects from the summation of the five test sets in the five-fold validation process, which was also the complete dataset used in this study. We carried out 1,000,000 bootstrapping for pairwise significance tests in this study to choose the optimal models. The *p*-value and 95% confidence interval were calculated based on the empirical probability during the 1,000,000 bootstrapping operations.

2.7 Visualization of Saliency Maps

To better interpret the deep learning neural network's understanding of PD movement pathology, we pulled out the saliency maps to show the attention of the neural network. The saliency was computed from the gradient of the sum of the outermost layer corresponding to the input. The gradients were computed by

the *threano.grad* function [3]. Then we visualized the saliency map as well as the original input using *ggplot2* in R (**Figure 4**). **More examples** of the saliency maps we extracted from PD patients and healthy controls were shown in **Figure S7-12**.

Supplementary References:

1. Bradski, G., and Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library ("O'Reilly Media, Inc.").
2. Bradley, A.P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit. 30, 1145–1159.
3. The Theano Development Team, Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., et al. (2016). Theano: A Python framework for fast computation of mathematical expressions.