Decoding task states by spotting salient patterns at time points and brain regions

Yi Hao Chan[0000−0002−2393−1110], Sukrit Gupta[0000−0002−8974−8482], L. L. Chamara Kasun, and Jagath C. Rajapakse[0000−0001−7944−1658]

School of Computer Science and Engineering,
Nanyang Technological University, Singapore.
{yihao001, sukrit001, chamarakasun, asjagath}@ntu.edu.sg

Abstract. During task performance, brain states change dynamically and can appear recurrently. Recently, recurrent neural networks (RNN) have been used for identifying functional signatures underlying such brain states from task functional Magnetic Resonance Imaging (fMRI) data. While RNNs only model temporal dependence between time points, brain task decoding needs to model temporal dependencies of the underlying *brain states*. Furthermore, as only a subset of brain regions are involved in task performance, it is important to consider subsets of brain regions for brain decoding. To address these issues, we present a customised neural network architecture, Salient Patterns Over Time and Space (SPOTS), which not only captures dependencies of brain states at different time points but also pays attention to key brain regions associated with the task. On language and motor task data gathered in the Human Connectome Project, SPOTS improves brain state prediction by 17% to 40% as compared to the baseline RNN model. By spotting salient spatio-temporal patterns, SPOTS is able to infer brain states even on small time windows of fMRI data, which the present state-of-the-art methods struggle with. This allows for quick identification of abnormal task-fMRI scans, leading to possible future applications in task-fMRI data quality assurance and disease detection. Code is available at `https://github.com/SCSE-Biomedical-Computing-Group/SPOTS`.

Keywords: Attention · Decoding brain activations · Embeddings · Recurrent neural networks · Task functional magnetic resonance imaging

1 Introduction

Deep neural network models have been recently used to decode brain states in functional Magnetic Resonance Imaging (fMRI) scans. For example, recurrent neural networks (RNN) were used to infer underlying task states from task-fMRI data [12,13] and feedforward neural networks were used to identify brain regions associated with disease states from resting-state fMRI data [9,10]. However, spatio-temporal variations that define dynamic brain states during task performance [15] have not been considered. This requires a decoding model that is able to pick out both spatial and temporal patterns associated with the task.

In this paper, we improve the state-of-the-art for brain state classification of task-fMRI data by ameliorating issues that are present in previous approaches.

The first issue is the effect of task state repetition due to sub-tasks being performed repeatedly. During task-fMRI experiments, participants are often asked to perform different sub-tasks while their responses are recorded along with corresponding response times. In order to decode brain activations, dynamic brain state labels are assigned to time points for each sub-task. Although RNNs have been used to predict brain states dynamically from task-fMRI data [12,13], they do not deal with brain states (denoted by the sub-task label) appearing within the time window of the stimuli as they simply model the current state based on previous time points. This is insufficient as the present state may not just depend on previous time points, but on states occurring within the time window too. For example, when listening to a question based on a context described beforehand, the subject thinks about the answer that has to be selected after the question. Therefore, to determine the present state, we may have to give attention to time points (representing brain states that are important for the sub-task) within the entire stimulation window and not just to the previous time points.

The second issue is that they did not consider functional specialisation during task performance. In the human brain, specific regions perform specialized functions [11]. While there is a set of task-general regions that participate across all tasks, there is another set of task-specific regions that differentiates one task from another [17]. Thus, brain decoding needs to focus on specific brain regions associated with the task, instead of learning from all activations to the network.

We consider both issues and proposed a customised architecture, Salient Patterns Over Time and Space (SPOTS), that learns dependencies between brain states and handles spatial interactions, for the purpose of brain decoding. SPOTS uses spatial embedding to learn a subset of regions that can differentiate between the brain states. It also uses an RNN encoder-decoder to learn the dependencies between the states within the whole window of time points [21], so as to better model brain states. An attention mechanism [1] is implemented to focus on specific brain regions and time points that are associated with the task state.

The potential of the proposed SPOTS model is demonstrated on language and motor task-fMRI data obtained from the Human Connectome Project [6]. We showed that SPOTS performs significantly better than the existing state-of-the-art methods using RNN models for decoding brain states, and that such results are possible only with the combination of both spatial embedding and attention mechanism. Furthermore, we showed how SPOTS provides greater interpretability than baseline RNN models, in terms of studying spatio-temporal relationships. In sum, we have made the following three key contributions:

- Proposed a customised architecture, SPOTS, that considers both spatial and temporal relationships in task-fMRI data for brain decoding;
- Performance of the proposed architecture is significantly better than the state-of-the art RNN models [13], especially for smaller window sizes; and
- Provided an interpretable method to study salient spatio-temporal patterns that are important considerations when predicting a brain state

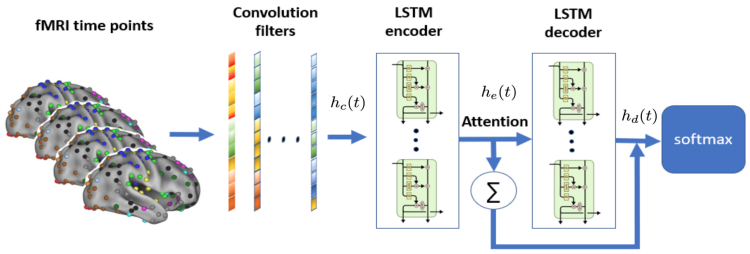


Fig. 1: The proposed architecture for SPOTS. Convolution filters are used to learn an embedding, reducing N brain regions into K dimensions. The spatially-compressed representation of the fMRI time series is then passed into an encoder-decoder network with an attention layer to identify salient time points.

2 Salient Patterns Over Time and Space (SPOTS)

The SPOTS architecture (figure 1) is made up of 4 sub-modules: (1) an embedding layer consisting of 1-dimensional convolution filters; (2) an encoder consisting of one or more long short term memory (LSTM) layers; (3) a decoder consisting of one or more LSTM layers; and (4) an attention module consisting of an attention layer. These sub-modules are then followed by a softmax classification layer that outputs the predicted task state.

Input SPOTS takes in fMRI time series in a time window of size T, from N functionally relevant brain regions. Let the input time series be $x = (x(t))_{t=1}^{T}$ where $x(t) \in \mathbb{R}^N$ denotes the input features at time point t.

Embedding layer The embedding layer aims to learn a compact representation of task-specific brain regions using K 1-dimensional spatial convolution operations. The use of such embeddings is motivated by the presence of functional specialisation in the brain whereby only a subset of regions are involved in task performance. Also, this has the benefit of shorter training time and a smaller model with fewer parameters (as compared to using time series from all N region of interests). However, this is different from directly learning a $N \times K$ embedding - we use K convolution filters that convolve across the N regions for each time point in T. This is done because we want to predict labels for each time point in the window and not a single label for the whole window. Additionally, fMRI datasets are small (< 1000) and directly learning a $N \times K$ embedding is unlikely to work well with limited data.

Let w_k be the filter weights for kth filter and $h_c(t) = (h_{c,k}(t))_{k=1}^{K}$ be the convolution layer output where $h_{c,k}(t)$ is given by:

$$h_{c,k}(t) = x(t) \circledast w_k \tag{1}$$

where $\circledast$ denotes the convolution operation in spatial domain. Note that $w_k \in \mathbb{R}^N$ and output $h_c(t) \in \mathbb{R}^{1 \times K}$ where K is the number of filters.

Encoder In order to capture long term dependencies, we use a RNN encoder-decoder architecture to learn temporal relationships in fMRI data. The encoder of SPOTS consists of one or more LSTM layers. Considering one LSTM layer, the encoder output $h_e(t) \in \mathbb{R}^{E \times K}$, where E is the number of hidden units of the encoder LSTM, is given by:

$$h_e(t) = \text{lstm}(h_c(t), h_e(t-1)) \tag{2}$$

Decoder The decoder consists of one or more LSTM layers. Instead of a traditional LSTM decoder, we use a random decoder that passes random values drawn from a normal distribution as inputs to the decoder [20]. For one LSTM layer, the decoder output $h_d(t) \in \mathbb{R}^{D \times K}$, where D is the number of hidden units of the encoder LSTM, is given by:

$$h_d(t) = \text{lstm}(h_e^*(t), h_d(t-1)) \tag{3}$$

$h_e^*(t)$ is obtained from drawing random samples Gaussianly from $h_e(t)$.

Attention The attention layer aims to find the relevant time points in the task fMRI data that are useful to determine the brain state. Attention score output $s(t)$ is given by:

$$\begin{aligned} \alpha(t) &= \text{softmax}\big(V \tanh(W h_d(t) + U h_e(t))\big) \\ s(t) &= \alpha(t) \cdot h_e(t) \end{aligned} \tag{4}$$

where $U \in \mathbb{R}^{D \times E}, V \in \mathbb{R}^{1 \times E}, W \in \mathbb{R}^{E \times D}$ are weight matrices forming the attention module and the dot product $(\cdot)$ is taken in the spatial domain.

Softmax layer The attention output $s(t)$ and the decoder output $h_d(t)$ are concatenated to obtain the input $h(t)$ to the output softmax layer:

$$h(t) = [h_d(t), s(t)] \tag{5}$$

where $h(t) \in \mathbb{R}^{D+E}$. The softmax layer output $\hat{y}(t)$ is given by:

$$\hat{y}(t) = \text{softmax}(Z h(t) + b) \tag{6}$$

where $Z \in \mathbb{R}^{C \times (D+E)}$ and $b \in \mathbb{R}^C$ denote the weight matrix and bias vector of the softmax layer and C denotes the number of brain states.

Learning of SPOTS Cross-entropy is used as the cost function J as:

$$J = -E_x[y(t) \log(p(\hat{y}(t)))] \tag{7}$$

where E_x denotes the expectation over inputs $x(t)$ and $y(t)$ denotes the brain state label. The cross-entropy cost function is minimized using Adam optimiser.

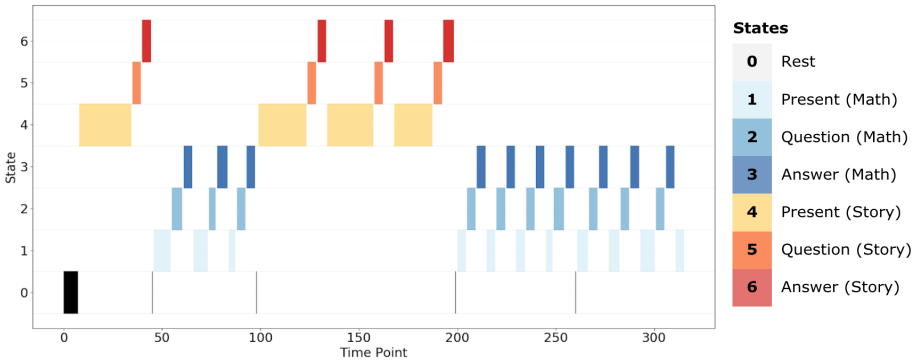


Fig. 2: Distribution of sub-tasks over time points (3 time points = 4s) for the language task. Each block lasted for around 30s. As math sub-tasks are shorter than story sub-tasks, more math sub-tasks were performed within a block.

3 Results

3.1 Dataset and pre-processing

Data used in this paper were obtained from the S900 release of the Human Connectome Project (HCP) [2]. In total, 848 task-fMRI scans (each for right to left and left to right phase encoding) were obtained using a Siemens Skyra 3T scanner at the Washington University. The scans were pre-processed using the HCP pre-processing pipeline [7], which performs correction for gradient distortion, motion and echo-planar imaging distortion, as well as registration to the T1-weighted image. From the time series produced by the pipeline, the average time series of voxels found within a 2.5mm radius for 264 regions of interests (ROI) derived from the Power atlas [17] were computed and used as the input.

Specifically, we used task-fMRI data from the language [3] and motor [4] tasks. The language task was chosen as it was made up of two sub-tasks of very different lengths: math sub-tasks and story sub-tasks. The math sub-tasks were much shorter than story sub-tasks. The motor task was made up of 5 sub-tasks involving feet (left and right), hand (left and right) and tongue movement. It was chosen as the length of its sub-tasks are relatively uniform (and are in between math and story sub-tasks), allowing us to study how the model performance changes depending on the mix of sub-tasks lengths present in the task. Furthermore, the two tasks engage different regions of the brain.

The language task is made up of multiple interleaved activity blocks with each block comprising math (presentation of math question, math question, and answer to question) and story (presentation of story, question based on story, and answer to question) sub-tasks as seen in figure 2. The initial state was a 'rest' interval between each block. A similar illustration of the motor task can be found in figure S1 of the supplementary materials. All states were brought forward by 6s to account for the delay between stimulus and hemodynamic response [14] and were used as brain state labels for training the models.

Table 1: Comparison of model accuracies from baseline LSTM (model adapted from Li et al.[13]), LSTM with embedding, LSTM with attention, and SPOTS across various window sizes and across language and motor tasks. Results for window sizes 10-50 are shown, Fig S4 shows the full results (window sizes 10-80).

Window Size	LSTM (Li et al.[13])	LSTM (with embedding)	LSTM (with attention)	SPOTS
		Language		
10	34.5% ± 0.8%	47.3% ± 0.5%	44.9% ± 0.6%	**68.0%** ± 1.2%
20	43.8% ± 0.9%	62.7% ± 1.0%	45.7% ± 3.9%	**81.4%** ± 0.4%
30	54.3% ± 0.7%	72.7% ± 0.6%	45.6% ± 5.9%	**84.8%** ± 0.6%
40	60.1% ± 1.0%	77.3% ± 0.5%	46.0% ± 3.2%	**85.5%** ± 2.4%
50	62.9% ± 0.9%	78.4% ± 0.4%	40.2% ± 0.6%	**85.6%** ± 0.6%
		Motor		
10	30.3% ± 0.6%	36.0% ± 0.5%	36.2% ± 0.8%	**58.7%** ± 1.0%
20	36.6% ± 1.2%	66.9% ± 0.7%	44.8% ± 1.0%	**88.3%** ± 0.9%
30	51.4% ± 0.7%	79.2% ± 0.5%	72.2% ± 2.1%	**93.8%** ± 2.0%
40	62.7% ± 0.8%	86.0% ± 0.6%	64.5% ± 2.8%	**94.7%** ± 0.5%
50	78.6% ± 1.1%	90.3% ± 0.6%	72.2% ± 4.3%	**95.9%** ± 1.2%
Parameters	1.6 million	1.1 million	3.2 million	2.7 million

3.2 Classifier performance

SPOTS was implemented in Keras [5] with TensorFlow backend. Experiments were done on a server with 4 Nvidia P100 GPUs. As LSTMs are expensive to train, extensive hyperparameter tuning was infeasible. Thus, we determined the hyperparameters by referring to previous work done by [13]. They experimented with LSTMs ranging from 1 to 3 layers and number of hidden units ranging from 32 to 1024. We chose the model with 1 LSTM layer and 512 hidden units as it had the highest accuracy and was the least complex model with. We varied window size T from 10 to 80. We used 85% of the samples [1] for training and the rest for testing. For each window size, we performed the experiment for 5 different seeds. The following hyperparameters were used: batch size = 32, early stopping with patience = 10, Adam optimizer with learning rate = 0.05, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ with gradients clipped to a maximum norm of 1. Gaussian noise with $\mu = 0.4$ and $\sigma = 0.2$ was used as inputs to the decoder. The same model architecture and hyperparameters were used for the two tasks.

Performance of SPOTS are shown in Table 1 in comparison with various RNN models that were implemented with LSTM. We experimented with only the LSTM layer (configuration is similar to [13]), LSTM with embedding, LSTM encoder-decoder with attention mechanism, and then SPOTS. We found that

[1] A sample refers to input data of length T. However, the model makes a prediction at every time point, thus accuracy is computed based on single time points.

SPOTS gave the best classification performance across window sizes and tasks. These results were produced on the fMRI scans using the right to left phase encoding. We further validated them with the left to right phase encoding scans (for window size 10 to 50) and got similar performances, with SPOTS outperforming baseline RNNs (figure S3, supplementary materials). From these results, it can be inferred that the identification of both spatial and temporal patterns is needed to obtain the best model performance. Three key aspects are:

1. The advantage gained from finding spatio-temporal patterns increases with decreasing window sizes. With smaller window sizes, the amount of variation present in a time series window increases, which makes it harder for the LSTM to learn due to catastrophic forgetting [8]. For example, when the model is trained on windows containing math sub-tasks, the memory stored in LSTM units loses information about the story sub-tasks. Attention can overcome this by learning direct mappings between the time points [18].
2. However, when performed without spatial embedding, the LSTM encoder seems to struggle with the large dimensionality of the input and the attention mechanism isn't able to learn useful relationships. Interestingly, LSTM + Attention showed greater improvement (as time window increases) for the motor task than the language task. Thus, it can be inferred that attention (without embeddings) thrives in datasets where short sub-tasks are present.
3. For both tasks, using a more compressed representation obtained via the learnt embedding leads to an increase in model performance. While the use of embeddings results in huge performance gains by themselves, adding the attention mechanism further leads to improved performance and this increase is especially significant for small window sizes. One can see from Table 1 that LSTM + Attention has more parameters than LSTM but did not always perform better. On the contrary, SPOTS outperformed LSTM with attention despite being smaller. This shows the value of using spatial embeddings, and how the reduced dimensionality from the learnt embeddings helped the attention module to learn useful relationships.

3.3 Interpretation of results

Spatial analysis Algorithms such as DeepLIFT [19] can be used to compute salience scores for each feature. However, they are difficult to be implemented on a customised model like SPOTS. As a proxy to evaluating salient regions, we looked into the filter weights of the convolution filters in the embedding layer. Using ROIs obtained from [11,17], we observed that rows which map to the ROIs in the auditory module have consistently greater weights despite having inputs that are well spread across the distribution of inputs (figure S3). Similarly, the learnt embeddings placed greater weight on motor-related ROIs for motor tasks (figure S3). These findings correspond to the expected task-based activations but they are not a guarantee of the saliency of these regions - they should be further reaffirmed with model interpretability approaches such as DeepSHAP [16].

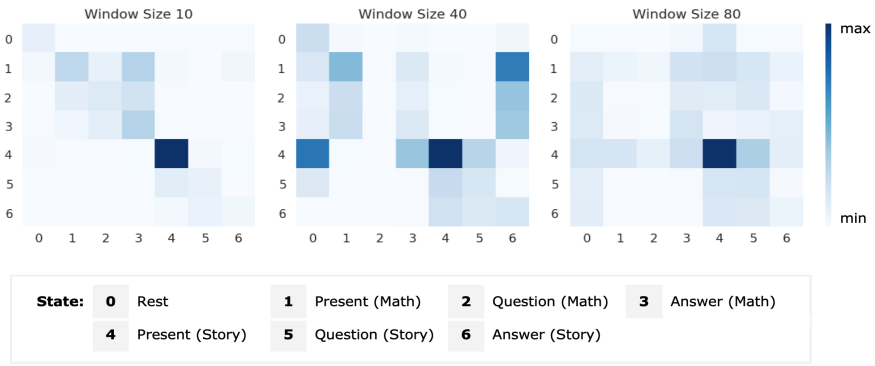


Fig. 3: Color-coded attention maps at various window sizes. Reading the attention map row by row shows which states (on the x-axis) are being focused on when predicting that the present state belongs to the current row.

Temporal analysis To study what states are important to discern between different states, we define the attention matrix $A = (a_{pq}) \in \mathbb{R}^{C \times C}$ where a_{pq} denotes the sum of attention scores that task state p (occurring at any time point) pays to task state q for the sample. We computed the attention map for all time windows by adding attention matrices derived from all the test samples where at least half of the predictions for that window are correct. We produced a representative attention map for the language task (figure 3) and motor task (figure S2 in the supplementary materials). Although attention maps show mappings between time points only, studying the patterns produced on the maps allow us to make inferences about relationships between states.

For window size 10, not only was self-attention prominent but the story-based tasks and math-based tasks paid greater attention to their respective sub-tasks too. On the other hand, the motor task only had self-attention. This shows how the attention mechanism is able to pick up temporal dependencies of underlying brain states within the time window - not just previous time points.

For window size 40, maps from both tasks showed that the model started to pay attention on the other sub-tasks. Interestingly, self-attention is reduced more greatly in the motor task.

For window size 80, the attention map became more diffused for both tasks and the focus of the map is dominated by self-attention of the most prevalent class (state 4 for language and state 0 for motor). This could explain the reduced advantage of SPOTS as window size gets larger - larger windows means fewer data samples to train the attention map on, leading to less accurate attention weighting on the states. We expect this to improve with more data.

4 Conclusion

In this paper, we showed that SPOTS was able to decode brain states in task-fMRI better than existing methods such as those based on RNN. By spotting time points and salient regions, SPOTS helps to boost the performance of baseline RNNs by a large margin, especially for small window sizes. This is especially useful when considering how SPOTS could be used as part of the fMRI data collection process for large-scale studies. With a model trained on other subjects performing the same task, the model will be able to detect subtle differences in task performance and check if the subject is performing the task well. Being able to predict well with smaller window sizes, SPOTS makes early intervention possible, thus improving the quality of acquired data. Another avenue to explore is the use of SPOTS for disease diagnosis. After being trained on a dataset of healthy subjects, SPOTS could be used to identify patterns of activity that deviate from the norm. Requiring only a small window size will allow greater amount of training data to be generated, helping the model to learn better.

5 Acknowledgement

This work was partially supported by AcRF Tier 1 grant RG 116/19 of Ministry of Education, Singapore. Data were provided [in part] by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. ICLR (2015)
2. Barch, D.M., Burgess, G.C., Harms, M.P., Petersen, S.E., Schlaggar, B.L., Corbetta, M., Glasser, M.F., Curtiss, S., Dixit, S., Feldt, C., et al.: Function in the human connectome: task-fMRI and individual differences in behavior. Neuroimage **80**, 169–189 (2013)
3. Binder, J.R., Gross, W.L., Allendorfer, J.B., Bonilha, L., Chapin, J., Edwards, J.C., Grabowski, T.J., Langfitt, J.T., Loring, D.W., Lowe, M.J., et al.: Mapping anterior temporal lobe language areas with fMRI: A multi-center normative study. NeuroImage **54**(2), 1465 (2011)
4. Buckner, R.L., Krienen, F.M., Castellanos, A., Diaz, J.C., Yeo, B.T.: The organization of the human cerebellum estimated by intrinsic functional connectivity. Journal of neurophysiology (2011)
5. Chollet, F.: Deep Learning with Python and Keras: The practical guide from the developer of the Keras library. MITP-Verlags GmbH & Co. KG (2018)
6. Glasser, M.F., Smith, S.M., Marcus, D.S., Andersson, J.L., Auerbach, E.J., Behrens, T.E., Coalson, T.S., Harms, M.P., Jenkinson, M., Moeller, S., et al.: The human connectome project's neuroimaging approach. Nature neuroscience **19**(9), 1175 (2016)

7. Glasser, M.F., Sotiropoulos, S.N., Wilson, J.A., Coalson, T.S., Fischl, B., Andersson, J.L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J.R., et al.: The minimal preprocessing pipelines for the human connectome project. Neuroimage **80**, 105–124 (2013)
8. Goodfellow, I.J., Mirza, M., Xiao, D., Courville, A., Bengio, Y.: An empirical investigation of catastrophic forgetting in gradient-based neural networks. arXiv preprint arXiv:1312.6211 (2013)
9. Gupta, S., Chan, Y.H., Rajapakse, J.C., Initiative, A.D.N., et al.: Decoding brain functional connectivity implicated in AD and MCI. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 781–789. Springer (2019)
10. Gupta, S., Chan, Y.H., Rajapakse, J.C.: Obtaining leaner deep neural networks for decoding brain functional connectome in a single shot. Neurocomputing (Accepted, In Press) (2020)
11. Gupta, S., Rajapakse, J.C.: Iterative consensus spectral clustering improves detection of subject and group level brain functional modules. Scientific reports **10**(1), 1–15 (2020)
12. Li, H., Fan, Y.: Brain decoding from functional MRI using long short-term memory recurrent neural networks. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 320–328. Springer (2018)
13. Li, H., Fan, Y.: Interpretable, highly accurate brain decoding of subtly distinct brain states from functional MRI using intrinsic functional networks and long short-term memory recurrent neural networks. NeuroImage **202**, 116059 (2019)
14. Liao, C.H., Worsley, K.J., Poline, J.B., Aston, J.A., Duncan, G.H., Evans, A.C.: Estimating the delay of the fMRI response. NeuroImage **16**(3), 593–606 (2002)
15. Loula, J., Varoquaux, G., Thirion, B.: Decoding fMRI activity in the time domain improves classification performance. NeuroImage pp. 1–8 (Apr 2018)
16. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in neural information processing systems. pp. 4765–4774 (2017)
17. Power, J.D., Cohen, A.L., Nelson, S.M., Wig, G.S., Barnes, K.A., Church, J.A., Vogel, A.C., Laumann, T.O., Miezin, F.M., Schlaggar, B.L., et al.: Functional network organization of the human brain. Neuron **72**(4), 665–678 (2011)
18. Serra, J., Suris, D., Miron, M., Karatzoglou, A.: Overcoming catastrophic forgetting with hard attention to the task. arXiv preprint arXiv:1801.01423 (2018)
19. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. arXiv preprint arXiv:1704.02685 (2017)
20. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised Learning of Video Representations using LSTMs. In: Proceedings of the 32nd International Conference on International Conference on Machine Learnings. vol. 37, pp. 843–852 (2015)
21. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to Sequence Learning with Neural Networks. NIPS (2014)

Supplementary Materials: Decoding task states by spotting salient patterns at time points and brain regions

Yi Hao Chan[0000−0002−2393−1110], Sukrit Gupta[0000−0002−8974−8482], L. L. Chamara Kasun, and Jagath C. Rajapakse[0000−0001−7944−1658]

School of Computer Science and Engineering,
Nanyang Technological University, Singapore.
{yihao001, sukrit001, chamarakasun, asjagath}@ntu.edu.sg

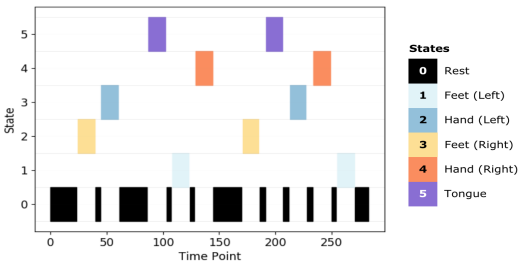


Fig. S1: Distribution of sub-tasks over time points for motor task stimuli. Each sub-tasks are of similar duration. The Rest state is inclusive of 3 fixation blocks (the thicker black boxes) lasting around 15s each.

Fig. S2: Color-coded attention maps at various window sizes, for motor task. Reading the map row by row shows which states (on the x-axis) are being focused on when predicting that the present state belongs to the current row.

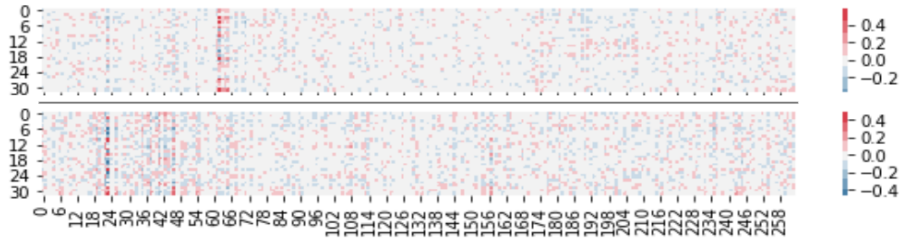
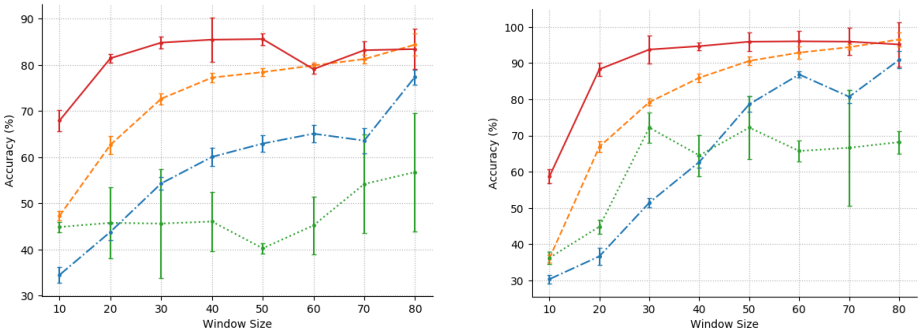
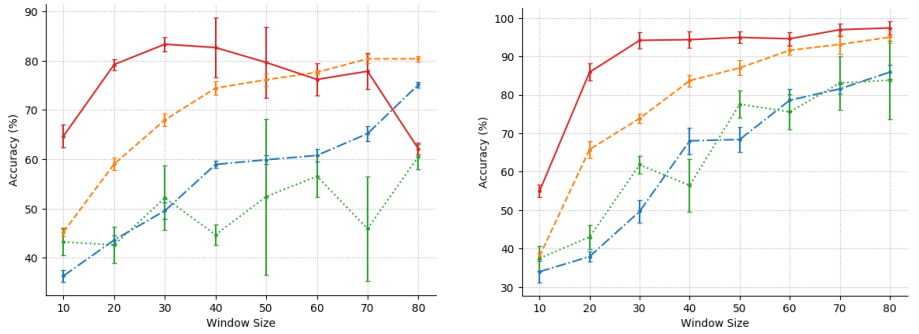


Fig. S3: Learnt embedding for the Language (top) and Motor (bottom) task. Column 14 to Column 45 corresponds to the Motor module, including Hand, Feet and Mouth. Column 60 to Column 72 corresponds to the Auditory module



(a) Language (RL)

(b) Motor (RL)



(c) Language (LR)

(d) Motor (LR)

Fig. S4: Comparison of model accuracies for language and motor task. Red solid line = SPOTS, Orange dashed line = LSTM + Embedding, Green dotted line = LSTM + Attention, Blue dash-dotted line = LSTM. Error bars = 95% confidence intervals.