2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

Uncovering the effect of different brain regions on behavioral classification using recurrent neural networks

Yongxu Zhang[1], Catalin Mitelut[2], Greg Silasi[3],
Federico Bolanos[4], Nicholas Swindale[5], Timothy Murphy[6], Shreya Saxena[1]

***Abstract*— As our ability to record neural activity from a larger number of brain areas increases, we need to develop tools to understand how this activity is related to ongoing behavior. Recurrent neural networks (RNNs) have been shown to perform successful classification for sequence data. However, they are black box models: once trained, it is difficult to uncover the mechanisms that they are using to classify. In this study, we analyze the effect of RNNs on classifying behavior using a simulated dataset and a widefield neural activity dataset as mice perform a self-initiated behavior. We show that RNNs are comparable to, or outperform, traditional classification methods such as Support Vector Machine (SVM), and can also lead to accurate prediction of behavior. Using dimensionality reduction, we visualize the activity of the RNNs to better understand the classification mechanisms of the RNNs. Finally, we are able to accurately pinpoint the effect of different regions on behavioral classification. This study highlights the utility and interpretability of RNNs while classifying behavior using neural activity from different regions.**

I. INTRODUCTION

Different regions in the cortex have been anatomically defined as having distinct relationships with behavioral activity, which consists of both receiving signals from and contributing towards specific behavior. For example, the somatosensory cortex receives somatic signals from touch and proprioceptive sensors across the body, the olfactory bulb encodes for smell-related signals from the nose, and the motor cortex can directly control muscles [1]. As our ability to record from larger swathes of the brain with unprecedented spatial and temporal resolution increases, our understanding of how neural activity is related to ongoing behavior is changing; it has recently been shown that movement signals are encoded in population recordings and single neurons across the mouse cortex, including in many sensory regions [2], [3]. However, how the neural signals from different regions contribute to behavioral decoding across time is less well understood. One important consideration, especially in brain machine interface applications, is the ability to decode different classes of behavior. For example, Soon et al. used fMRI data to decode human behavior, and showed that the human brain begins to prepare an upcoming decision before awareness [4], and López-Larraz et al. decoded upper limb self-initiated movements via EEG data [5]. However, pinpointing the contribution of each region towards classification of different behaviors is key towards illuminating the role of each region. Thus, methodological development towards revealing the importance of different brain regions for decoding is crucial.

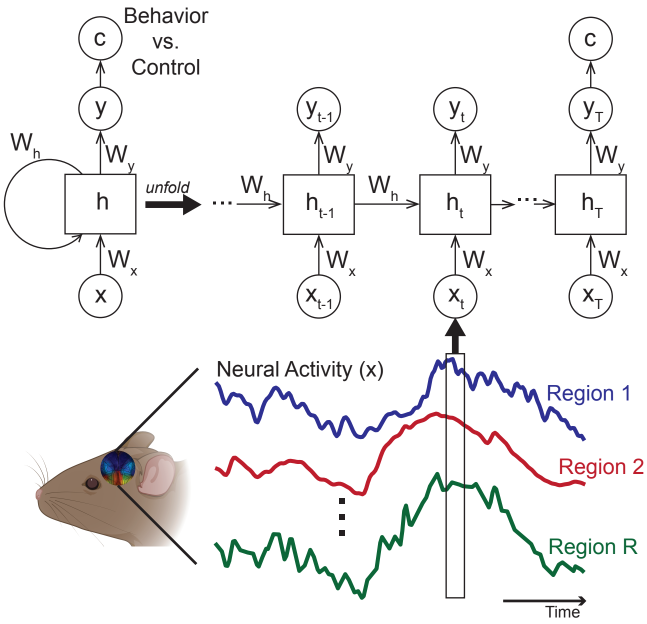

Fig. 1. A schematic showing the input data and architecture of the recurrent neural network (RNN) used for behavioral classification of neural activity from different brain regions

When classifying using temporal activity, it is important to work with sequence models. The Recurrent Neural Network (RNN) is a sequence model derived from feed-forward neural networks, which explicitly models the passage of time in the internal states. Using a memory component, the output of an RNN is influenced by the previous state, and thus also by past inputs. As a consequence of using weighted memory and feedback loops, RNNs are efficient in classification, and have been successfully used towards classification of sequence data. For instance, in [6], Yogatama et al. used RNNs to classify text, in [7], the authors performed image

[1] Yongxu Zhang and Shreya Saxena are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville 32603 FL, USA zhangyongxu@ufl.edu, shreya.saxena@ufl.edu

[2] Catalin Mitelut is with Department of Biology, New York University, New York, NY, USA.

[3] Greg Silasi is with Department of Cellular and Molecular Medicine, University of Ottawa, Ottawa, Ontario, Canada.

[4] Federico Bolanos is with RIKEN Center for Brain Science, Japan.

[5] Nicholas Swindale is with Department of Ophthalmology and Visual Sciences Research group University of British Columbia, Vancouver, British Columbia, Canada.

[6] Timothy Murphy is with Department of Psychiatry, Kinsmen Laboratory of Neurological Research, University of British Columbia, Vancouver, British Columbia, Canada.

978-1-7281-1178-0/21/$31.00 ©2021 IEEE

classification via RNNs, and in [8], Cui et al. analyzed the diagnosis of Alzheimer's disease by using RNNs to perform classification. However, training RNNs can be difficult: due to the complexity of their network structure, it usually takes the adjustment of many hyper-parameters to achieve the best capabilities and generalize on unseen data. Moreover, RNNs are black boxes: although they are a universal function approximator, they do not provide insights into the structure of these functions [9]. In this study, we analyze the effect of RNNs on classifying behavior using a simulated dataset and a widefield neural activity dataset as mice perform a self-initiated behavior. Moreover, we develop methods to visualize the effect of different regions on classification of brain-wide data. We show that (a) RNNs perform comparably or outperform traditional classification methods such as Support Vector Machines (SVM), (b) we are able to understand the classification mechanisms of the RNNs, and (c) we are able to accurately pinpoint the effect of different regions on behavioral classification. Lastly, we end with showcasing the ability of RNNs to perform predictive classification of the behavior.

II. METHODS

A. Recurrent Neural Networks (RNNs)

1) Architecture: Here, we describe our approach to build a classification model with temporal neural data $x \in \mathbb{R}^{R\times T}$ from R different brain regions and T time points as the input, with the outputs as the different classes of behavior. We implement a hidden recurrent layer with the *tanh* activation function, and a dense layer at the output with the sigmoid activation function σ to predict the class, here binary. Following are the equations of the RNN network.

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b_h) \quad \forall t \in [1,T] \tag{1}$$

$$y_t = \sigma(W_y h_t + b_y) \quad \forall t \in [1,T] \tag{2}$$

$$c = \begin{cases} 0, & \text{if } y_T < 0.5 \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

where x_t is the neural data from all R regions at time point t, $h_t \in \mathbb{R}^{N\times 1}$ is the value for the N hidden units at time point t, $W_x \in \mathbb{R}^{N\times R}$ is the input weight matrix, $W_h \in \mathbb{R}^{N\times N}$ contains the recurrent weights for the hidden layer, and $W_y \in \mathbb{R}^{1\times N}$ represents the output weight matrix. y_t is the output of dense layer. The specific structure is shown in Figure 1. For most of the analyses, we use the output at the last time point, i.e., y_T, to predict the class c (Equation 3). In Section III-D, to analyze predictive classification, we output a class c_t at every time step, based on the output y_t at every time step.

We chose the number of hidden units as 64, and we trained the network for 200 epochs using Adam at a learning rate of 0.0001. These hyper-parameters were determined using cross-validation on a sample session of the dataset. We used Keras with the Tensorflow backend to train our models [10], [11]. We performed all tasks on HiPerGator Computational Supercomputer at the University of Florida, with NVIDIA GeForce RTX 2080TI GPUs.

2) Accuracy quantification: We applied 10-fold cross-validation to all of our experiments which output the accuracy. Accuracy here is defined as $\frac{1}{K}\sum_{k=1}^{K}\frac{TP(k)+TN(k)}{TP(k)+TN(k)+FP(k)+FN(k)}$, where K is the number of folds, here 10, $TP(k)$ is true positives in the k^{th} fold, TN is true negatives, FP is false positives, and FN is false negatives. In addition to accuracy to quantify decoding performance, we used the following metrics for the widefield activity dataset.

- Area under curve (AUC): we calculated the area under the accuracy curve in different time windows, above chance level. This quantifies the decoding ability of the classifier.
- Earliest decoding time: this is the earliest time point after which we obtain consistent and significant decoding till behavior onset. Significance was determined using a one-tailed t-test at a significance level of $p < 0.05$ (after multiple hypothesis correction using the Benjamini-Hochberg procedure [12]). This represents the earliest time that the behavior can be reliably decoded.

As a comparison, we also applied SVM to classify the same data. The input of SVM consists of the flattened trials, i.e., the input dimension is RT.

3) Visualization: For deeper exploration of the mechanism of RNNs for decoding behavior, we visualized the RNN activity and the decoding process after the RNNs were trained. Principal component analysis (PCA) can be used to reduce the dimensionality of the RNN nodes' activity (h) from N to d, and these top PCs can then be visualized. We also use linear discriminant analysis (LDA) to perform supervised dimensionality reduction, which can reduce the dimensionality from N to $C-1$ dimensions, where C is the number of classes (here $C = 2$). We perform LDA using data from each time point, to characterize the maximum difference between the two classes in that time point.

For exploring the predicting ability of the RNNs over the course of the trial, we visualized the 'temporal output score' to understand how RNN predicts at each time step. Temporal output score S_t represents the classification accuracy at each time step of the input sequence t, using c_t as calculated using y_t [13], [14].

4) Quantifying the contribution of different regions: The important features for decoding are stored in both the time domain and brain region domain of the data. If the important features are occluded, the decoding accuracy may decrease [13], [15]. We first occlude time sub-sequence $x_{t:t+w_T}$ of the entire sequence $x_{1:T}$, where w_T refers to the width of the occlusion window in the time domain. Then, we calculate the sum score in time domain by adding the accuracy after occlusion to all the other time without occlusion. Intuitively, when the most important time periods are occluded, the decoding accuracy decreases the most, thus decreasing the accuracy added to other sub-sequences without occlusion. The occlusion in time domain is first applied to signals from all brain regions, and we use this to determine the reasonable occlusion window w_T. The equation for score sum $\bar{S}_t$ is

$\bar{S}_t = \sum_{t'=1}^{T_{sliding}} S_{t,t'}$. Here, $S_{t,t'}$ is defined as the following.

$$S_{t,t'} = \begin{cases} 0, & if \quad t \in [t' - \frac{w_T}{2}, t' + \frac{w_T}{2}], \\ S_{(x_{t'-\frac{w_T}{2}:t'+\frac{w_T}{2}}=0)}, & otherwise. \end{cases} \tag{4}$$

where $T_{sliding}$ is the number of sliding windows.

Finally, we explore the importance in both time and region domains with fixed window size as determined by time occlusion above. We call the accuracy of the base classifier (without occlusion) S. We then quantify the importance of a region as the decrease in accuracy after occlusion of each region in each time window (Equation 5). Thus, the regions and time periods with higher values for importance are considered to be the most important for decoding. The equation for importance $V_{r,t}$ for each region r and time point t is as following.

$$V_{r,t} = \frac{S - S_{(x_{r',t'}=0,\ \forall r' \in [r-\frac{w_R}{2}, r+\frac{w_R}{2}],\ \forall t' \in [t-\frac{w_T}{2}, t+\frac{w_T}{2}])}}{S} \tag{5}$$

Here, w_T was determined as above. For w_R, we used the number of sub-regions that fell within the same region as determined by their function and location.

B. Simulated Dataset

In order to uncover the effect of different features in both region and time domain on behavioral classification and characterize the rationality of the method, we generated a simulated dataset with very clear features that we wish to recover using our methods. In this data, each 'behavior' trial contains 200 time points, with 10 dimensions simulating the different brain region. For each brain region dimension, a 20-time point long peak with maximum amplitude 1 was included at sequential time points, simulating a traveling wave across brain regions, as shown in Figure 6A. Gaussian noise $\mathcal{N}(0,\sigma)$ was added to the trajectory from each region, with $\sigma = 0.05$. As a control, we simulated trials that have a peak in the same position as in the 'behavior' trials, with probability 0.5. We generated 4000 trials with equal numbers of 'behavior' and 'control' trials in the simulated dataset.

C. Widefield Neural Activity Dataset

Widefield experiments records large-scale neural activity from the mouse dorsal cortex through widefield calcium imaging. We analyze widefield neural activity while mice engage in a task. In the experiment, head-fixed water-deprived mice were trained to pull a lever and hold it at an angle (for $>$ 100ms) in order to receive a water supplement. A three-second lockout refractory period was implemented (i.e., only pulls $>$ 3 sec after a previous pull were rewarded). Rewarded lever pulls were identified online (using a lever analog signal). Widefield calcium imaging was recorded from the mouse dorsal cortex as previously described [16]. We identify the 'behavior' trials as trials that were tracked in real time to provide water reward, with the trial centered around the initiation of the lever pull behavior. As control trials, we take a random sequence from the task with the same number of time points as 'behavior'. Thus, the 'behavior' trials have a clear behavior initiated at the middle of the trial, unlike the 'control' trials. In order to further eliminate the influence of multiple instances of lever pulls occurring during a 'behavior' trial, we manually selected trials such that only one instance of lever pull is located at the middle in each 'behavior' trial. The neural activity is sampled at 30 time points per second, and each trial in this dataset contains 600 time points (20 seconds). We spatially align the imaged neural activity with the Allen mouse brain coordinate framework [17] using affine transformations, as previously performed in [2], [18]. We then take pixel-wise averages of the activity in each of 32 regions as identified by the Allen atlas, which form our input signals. We use a subset of the entire dataset to show the validity of our methods; here, we mainly show results from one mouse with multiple recorded sessions as it performs the trial. This mouse (M1) has 10 recorded sessions (76.2 ± 12.8 trials per session). We also use one session with 108 trials from another mouse (M2) performing the same task, to show the variability between how different mice perform the task, in Section III-C.

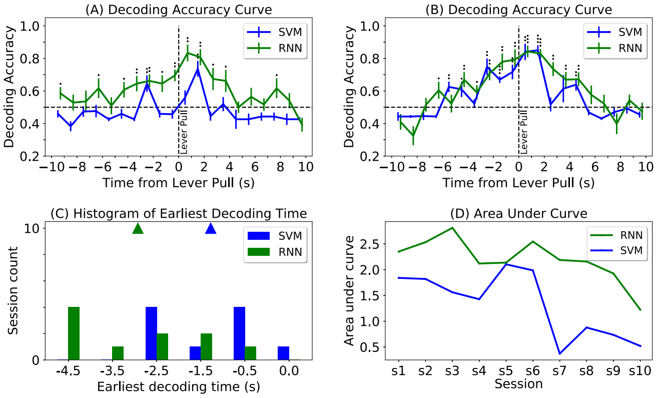

Fig. 2. (A) Decoding accuracy curve of SVM and RNN in a single session of M1. Blue (SVM) and green (RNN) curve are showing the decoding accuracy of 'lever pull' vs 'random' at each second (30 time points). We use the middle time point in each time window to represent the accuracy in this second. The horizontal dash line refers to the chance level (0.5), and the vertical dash line represents when the 'lever pull' happened. The stars represent the outcome of a one tailed t-test, with one star as $p < 0.05$, two stars as $p < 0.01$, and three stars as $p < 0.001$. (B) Decoding accuracy of SVM and RNN for another session of M1. (C) A histogram of the earliest decoding time using SVMs and RNNs for 10 sessions of M1. The earliest decoding time is the first time point after which we have consistent decoding with $p < 0.05$ (after multiple hypothesis correction). (D) Area under curve (AUC) for 10 sessions of M1 for SVMs and RNNs. The AUC quantifies the area under the accuracy curve above chance level.

III. Results

We first show that RNNs are either comparable in accuracy or significantly outperform SVMs for the datasets described above. We then examine the mechanisms of action of the RNNs, the ability of the RNN to perform continuous classification, and finally the effect of different regions on the classification accuracy. In the following, all results are reported on held-out test data.

A. Classification Accuracy using RNNs

We consider the classification accuracy of the widefield activity using RNNs as compared with SVMs. We first

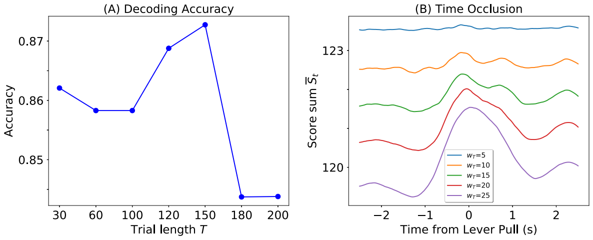

Fig. 3. (A) RNN decoding accuracy with different trial lengths T. The trial is always centered on ‘lever pull’. (B) Determining w_T: sum scores $\bar{S}_t$ from -3.3s to 3.3s with different time window sizes w_T, from 5 time points to 25 time points.

trained a series of models classifying 30 time points (around 1 second) of neural activity at a time, in order to quantify the earliest decoding time (see Methods). Figure 2A shows the classification accuracy in an example session, using the entire neural data. We see that the behavior classification accuracy is highest around lever pull. Using RNNs, the behavior can be classified significantly above chance up to several seconds prior to the lever pull. We also see that RNNs significantly outperform SVMs in several time windows. We also used Long Short-Term Memory networks (LSTMs) to perform the classification, with similar results as the RNNs (not shown). Figure 2B shows the accuracy curve in another example session for the RNNs as compared to the SVMs, and we see that the RNNs perform similar to SVMs for most time periods. Thus, RNNs either significantly outperform or are comparable to SVMs for classification. Next, we show the earliest decoding time in Figure 2C. We used multiple hypothesis correction to correct the p-value of a one-tailed t-test at each time window before ‘lever pull’, and we defined the earliest decoding time as the earliest time point that is significantly above chance level ($p < 0.05$) as this represents the first time point that can be reliably decoded above chance. For example, in Figure 2A, the earliest decoding time using RNNs is -4.5s since this is the first time window after which we can consistently decode significantly above chance. On the other hand, for this session, SVM has no earliest decoding time, thus 0s is considered as the earliest decoding time since the behavior happened in this time window. We see that, in these sessions, the RNN classifiers were able to accurately predict the behavior (above chance), earlier than the SVMs. Finally, we compare the accuracy using RNNs as compared to SVMs by showing the area under the accuracy curve (AUC) in Figure 2D for ten different sessions of M1 performing the same task. We consistently see that RNNs perform either comparably or with a higher accuracy than SVMs, presumably because RNNs explicitly take into account temporal structure in the model, and allow for a nonlinear representation while classifying. Note that SVMs have certain advantage in computational time: training an SVM classifier takes several seconds, as compared to training an RNN which can take several hours. As an example, it took 14 seconds to train an SVM classifier for the session shown in Figure 2A, as compared to 2 hours and 2 minutes to train the RNN.

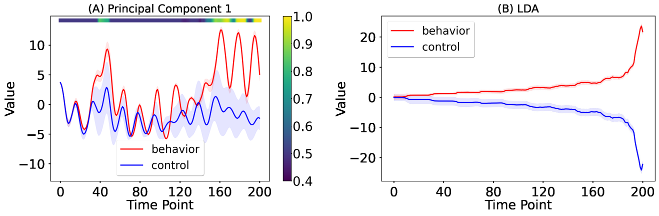

Fig. 4. Visualization of the trained RNN on the simulated dataset by using PCA, LDA, and temporal score. (A) 1^{st} PC of the RNN activity h, with inputs as the data from ‘behavior’ and ‘control’ test trials, the top color bar shows the temporal score: decoding accuracy at each time of classification; (B) LDA performed on the RNN activity at each time point, with the RNN activity h projected into the maximally differentiating LDA subspace.

We next considered the optimal length of temporal sequences for classification: taking too few time points may not adequately capture the relevant neural activity for classification purposes, and too many time points may encounter the limits of memory processing in the RNN. In Figure 3A, we show the relationship between the length of input sequence and the classification accuracy for the widefield activity dataset on validation data. Here, the models are trained on all trials from the 10 sessions from M1. We see that the accuracy is comparable across the different time window lengths, and a time window of 150 time points (around 5 seconds) is considered as the optimal time window length. Due to their higher validation accuracy, we focus on RNNs trained on input data of length 150 time points in Sections III-B, III-C and III-D.

Lastly, we considered the length of time window to occlude in the data in order to obtain a considerable decrease in classification accuracy. This inherently depends on the temporal correlations in the neural data itself. In Figure 3B, we plot the score sum $\bar{S}_t$ by occluding data from all regions in time windows (w_T) of length 5 to 25 time points, in a sliding window approach. Based on this analysis, we choose 15 time points to be the window size w_T that we can use to explore the importance of features in the time domain (Section III-C), since we see a clear increase in $\bar{S}_t$ for $w_T \geq 15$ time points.

B. Classification Mechanisms using RNNs

An RNN was trained to classify the ‘behavior’ vs. ’control’ trials in the simulated dataset (an example trial is shown in Figure 6A). We find that the classification accuracy using the RNN is 95%, comparable to that using SVM (99%). In order to visualize the trained RNN nodes’ activity ($h(t)$) succinctly, we apply principal component analysis (PCA) to $h(t)$ and show the evolution of the 1^{st} dimension in Figure 4A. The RNN trajectories starts to diverge between the two classes at an early time, and at around 140 time points in the first PC, the two trajectories from the two classes start to diverge quickly. In Figure 4B, we show the RNN activity after performing LDA at each time point; the RNN activity of two class are well separated from early in the trials. The temporal score reflects that the performance of RNN starts to improve only at the end of sequence, around 140 time points in Figure 4A. Thus, the divergence in activity between the

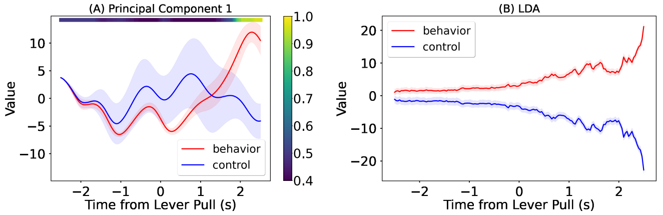

Fig. 5. Visualization of the trained RNN on the widefield dataset by using PCA, LDA, and temporal score. (A) 1^{st} PC of the RNN activity h, with inputs as the data from 'behavior' and 'control' test trials, the top color bar is the temporal score: decoding accuracy at each time of classification; (B) LDA performed on the RNN activity at each time point, with the RNN activity h projected into the maximally differentiating LDA subspace.

two classes (further accentuated using LDA in Figure 4B) does not exist in the output nodes until close to the final time step T, at which point the information moves from the memory to the output nodes and the classification is performed.

To analyze the classification mechanisms in the widefield neural data, an RNN was trained on the data from one session in M1 with 54 'behavior' trials and a matched number of control trials. Here, we consider a 150 length time window centered at lever pull, as suggested by Figure 3A. Again, we apply PCA to $h(t)$ and show the evolution of the 1^{st} dimension in Figure 5A (capturing 41% of the variance). While the PC does not start to diverge between the two classes until just before 2 seconds after the lever pull, we see that there are dimensions captured by LDA (Figure 5B) that show the difference in the RNN activity between the two classes from early on in the trial. We also visualize the temporal score of the RNN with the last timepoint accuracy of 0.94, and we see that the RNN has a high classification accuracy only near the end of the time sequence. Thus, the same principal applies as in the simulated dataset: the output nodes do not encode the information about the two classes until close to the final time step T, at which point the information moves from the memory nodes of the hidden layer to the output nodes.

C. Quantifying the contribution of different regions

In Figure 6, we show the importance matrix of simulated data by occluding different regions in time and region domain (see Methods for details). The matrix recovers the structure built into the trials, i.e., that the consistent presence of the peaks in different dimensions at sequential time windows determines which class is output. Next, we examine the results while applying this method to our widefield neural activity dataset.

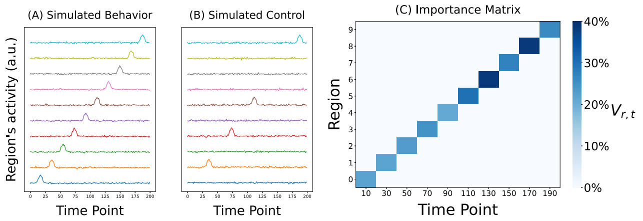

Fig. 6. (A) Example 'Behavior' trials for the simulated dataset. (B) Example 'Control' trials for the simulated dataset. (C) Importance matrix for the simulated dataset.

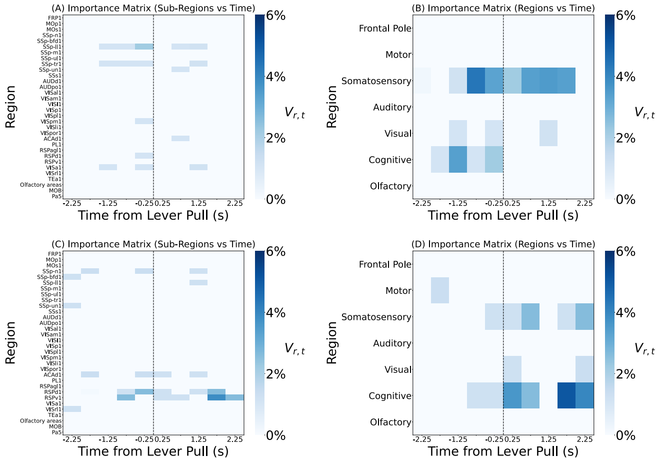

Fig. 7. Importance matrix of various regions vs time in two difference mice. M1: (A)(B); M2: (C)(D); (A)(C): Importance matrix of 32 sub-regions vs time; (B)(D): Importance matrix of 7 regions vs time. The dashed vertical line represents onset of the lever pull behavior. Here, Frontal Pole: FRP1; Motor: MOp1, MOs1; Somatosensory: SSp-n1 to SSs1 and Pa5; Auditory: AUDd1, AUDpo1; Visual: VISal1 to VISpor1, VISa1, VISrl1; Cognitive: ACAd1 to RSPv1 and TEa1; Olfactory: olfactory areas and MOB.

In Figure 7, we illustrate the temporal importance of different brain regions while classifying the behavior in single sessions of two different mice (M1 and M2) performing the same self-initiated behavior. Here, w_T is 15 time points (0.5s), and the darker color signifies that the relevant region is more important in that time window. The sessions we used for M1 and 2 both had 54 'behavior' trials and 54 'control' trials in the sessions. We trained separate RNNs on the data from M1 and 2; they had a non-occlusion decoding accuracy S of 0.87 and 0.73, respectively. Many sub-regions in the somatosensory and cognitive cortex show higher importance in decoding the behavior (Figure 7A,C). However, we see that the importance of any one region is quite small in the widefield dataset since movement signals may be encoded across the entire brain [2], [3]. The acronyms used in Figure 7A,C are defined in [19]. For M1, the most important region is primary somatosensory area, lower limb, layer 1 (SSp-ll1) at 0 to 0.5 second before the initiation of lever pull, perhaps showing expected somatic signals. For M2, the most important signal is in the retrosplenial area, ventral part, layer 1 (RSPv1) at 1.5 to 2 second after lever pull, perhaps due to reward consolidation after the behavior is carried out. Figure 7 shows considerable variability across mice in the regions that are important for classification.

Next, we combined the sub-regions together as regions with similar function in Figure 7B,D. Here, the regions have different numbers of sub-regions, i.e. w_R varies across the brain regions. Note that it may be the case that a region shows a high level of importance, but none of the sub-regions are important, implying that the signals in the sub-regions contain redundant information. In Figure 7B, the somatosensory region at 0.5 to 1 second before lever pull is most important, and in Figure 7D, the cognitive region

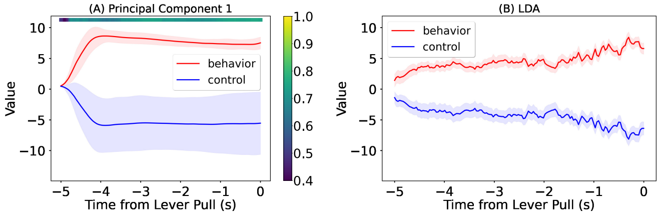

Fig. 8. Visualization of RNN activity by using PCA, LDA and temporal score with model of returning sequences in RNN layer. (A) 1^{st} PC of the RNN activity h, with inputs as the data from 'behavior' and 'control' test trials, the top color bar is the temporal score: decoding accuracy at each time of classification; (B) LDA performed on the RNN activity at each time point, with the RNN activity h projected into the maximally differentiating LDA subspace.

at 1.5 to 2 second after lever pull is most important. Thus, in M1, the time periods before behavior seem to be more important for classification, reflecting the importance of the planning phases of the behavior, whereas in M2, the regions that show activity in reaction to the movement seem to be more important towards classification. We also computed the importance matrix by using SVMs, but the importance of different regions is not as obvious as when using RNNs. This is mainly due to the high correlations in the input data, and SVM applies very similar weights to high correlation features [20].

D. Predictive Classification

In order to highlight the predictive capabilities of the RNN, we output a class at every time step c_t, and the total loss is amended to be the sum of the loss at each time point. We show the ability of the RNN to continuously classify the behavior in Figure 8. Note that this falls outside the scope of an SVM classifier. Here we consider a 150 length time window from 5s before lever pull to the onset of lever pull, in order to quantify the ability of the RNN to predict before the movement occurs. Here, the 1^{st} PC captures 79% of the variance in h and the RNN activity starts to consistently diverge between the two classes as early as 5s before lever pull. The temporal classification accuracy increases to a high level immediately, unlike in Figure 5A, and plateaus until the end of the sequence. However, the accuracy at the last time-point is 0.75. Therefore, the output nodes are able to encode the information about the two classes from an early stage when they are forced to predict early, but the final accuracy is seen to be lower.

IV. CONCLUSIONS

In this study, we used RNNs to explore behavioral classification using brain activity, and the effect of different brain regions at different times on the classification results. We showed that RNNs are comparable to or outperform SVMs at classification. Using dimensionality reduction, we visualize the RNNs mechanism of classification. Using a novel widefield neural activity dataset, we concluded that the self-initiated behavior can be predicted up to several seconds prior to action in mice, and that the somatosensory cognitive regions of the mouse cortex are important in decoding this behavior. In the future, we aim to develop methods to combine the activity of different sessions to develop across-session classifiers, and further analyze subject-to-subject variability. Additionally, we aim to develop methods to improve the classification accuracy while the network performs continuous classification, i.e. to devise networks that predict as accurately and as early as possible.

REFERENCES

[1] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, A. J. Hudspeth, and S. Mack, *Principles of neural science*. McGraw-hill New York, 2000, vol. 4.

[2] S. Musall, M. T. Kaufman, A. L. Juavinett, S. Gluf, and A. K. Churchland, "Single-trial neural dynamics are dominated by richly varied movements," *Nature neuroscience*, vol. 22, no. 10, pp. 1677–1686, 2019.

[3] C. Stringer, M. Pachitariu, N. Steinmetz, C. B. Reddy, M. Carandini, and K. D. Harris, "Spontaneous behaviors drive multidimensional, brainwide activity," *Science*, vol. 364, no. 6437, 2019.

[4] C. S. Soon, M. Brass, H.-J. Heinze, and J.-D. Haynes, "Unconscious determinants of free decisions in the human brain," *Nature neuroscience*, vol. 11, no. 5, pp. 543–545, 2008.

[5] E. López-Larraz, L. Montesano, Á. Gil-Agudo, and J. Minguez, "Continuous decoding of movement intention of upper limb self-initiated analytic movements from pre-movement eeg correlates," *Journal of neuroengineering and rehabilitation*, vol. 11, no. 1, pp. 1–15, 2014.

[6] D. Yogatama, C. Dyer, W. Ling, and P. Blunsom, "Generative and discriminative text classification with recurrent neural networks," *arXiv preprint arXiv:1703.01898*, 2017.

[7] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, 2017.

[8] R. Cui, M. Liu, A. D. N. Initiative *et al.*, "Rnn-based longitudinal analysis for diagnosis of alzheimer's disease," *Computerized Medical Imaging and Graphics*, vol. 73, pp. 1–10, 2019.

[9] A. M. Schäfer and H. G. Zimmermann, "Recurrent neural networks are universal approximators," in *International Conference on Artificial Neural Networks*. Springer, 2006, pp. 632–640.

[10] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[11] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[12] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal statistical society: series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995.

[13] J. Van Der Westhuizen and J. Lasenby, "Techniques for visualizing lstms applied to electrocardiograms," *arXiv preprint arXiv:1705.08153*, 2017.

[14] J. Lanchantin, R. Singh, B. Wang, and Y. Qi, "Deep motif dashboard: Visualizing and understanding genomic sequences using deep neural networks," in *Pacific Symposium on Biocomputing 2017*. World Scientific, 2017, pp. 254–265.

[15] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3429–3437.

[16] D. Xiao, M. P. Vanni, C. C. Mitelut, A. W. Chan, J. M. LeDue, Y. Xie, A. C. Chen, N. V. Swindale, and T. H. Murphy, "Mapping cortical mesoscopic networks of single spiking cortical or sub-cortical neurons," *Elife*, vol. 6, p. e19976, 2017.

[17] Q. Wang, S.-L. Ding, Y. Li, J. Royall, D. Feng, P. Lesnar, N. Graddis, M. Naeemi, B. Facer, A. Ho *et al.*, "The allen mouse brain common coordinate framework: a 3d reference atlas," *Cell*, vol. 181, no. 4, pp. 936–953, 2020.

[18] S. Saxena, I. Kinsella, S. Musall, S. H. Kim, J. Meszaros, D. N. Thibodeaux, C. Kim, J. Cunningham, E. M. Hillman, A. Churchland *et al.*, "Localized semi-nonnegative matrix factorization (locanmf) of widefield calcium imaging data," *PLoS computational biology*, vol. 16, no. 4, p. e1007791, 2020.

[19] A. I. for Brain Science, "Technical white paper: Allen mouse common coordinate framework and reference atlas," 2017.

[20] L. Toloşi and T. Lengauer, "Classification with correlated features: unreliability of feature ranking and solutions," *Bioinformatics*, vol. 27, no. 14, pp. 1986–1994, 2011.