Using Support Vector Machine and 3D Convolutional Neural Networks to Predict Brain States in Task-Based Functional Magnetic Resonance Imaging (fMRI)

Beck S. , Merrill B. , Soules M.
University of Michigan Master of Applied Data Science
April 24th, 2022

This is where we can explain how to access our work
You can find our work in our Repository on Github at : Team Brainiacs
Our landing page can be accessed at : Team Brainiac Landing Page

INTRODUCTION

Real-time functional magnetic resonance imaging neurofeedback (rtfMRI-NFB) is a technique used in functional magnetic resonance imaging (fMRI) to measure an individual's control over target brain activity. Brain computer interfaces are used to provide subjects with neurofeedback during a scan in the magnetic resonance imaging (MRI) machine in order to help subjects learn to modulate their neural activity during a given task[1]. Researchers use these tools to help study brain states and regions of interest (ROI) in the brain that may be involved in the given task. Traditionally, research using these tools has been analyzed using a univariate approach. Researchers have been studying the application of multivariate analyses[2] on rtfMRI-NFB using machine learning models, such as Support Vector Machines (SVM), and deep learning models, such as convolutional neural networks (CNN). These machine learning and deep learning methods being studied hold promise in clinical and exploratory research.

Real-time fMRI has potential to aid mental health experts in helping people who have substance use disorders learn to better regulate their reward system leading to rehabilitation and recovery. Research in rtfMRI for substance use disorders typically uses an a priori knowledge of predefined ROI centered around the left and right Nucleus Accumbens (NAcc) to analyze the output from images. Subjects studied are provided feedback using a brain computer interface based on percent signal change in voxel activity from the previous timepoints in these regions. However, little is known whether more regions exist that could be involved in reward circuitry. It may also be the case that there are individual differences in the mental strategies applied to the increase or decrease of neural activation.

We are interested in studying reward circuitry and regions of interest in task-based rtfMRI using these brain images to conduct a data-driven analysis. Through the use of SVM, we hope to discover other brain regions that may be active during the up and down regulation during rtfMRI. We are also interested in how a deep learning approach would compare against our predictions with SVM models.

MOTIVATION

We are collaborating with the University of Michigan Medicine Psychology Research Department under the research of Principal Investigator Dr. Meghan Martz, PhD. This opportunity was presented to us by one of our authors, Mary Soules, who works as an Application Programming Analyst Senior in the UM research lab processing and analyzing brain images captured through Dr. Martz study of real-time fMRI. We have three approaches to looking at the fMRI data captured through this study: First, single subject SVM to look at individual differences in brain activation and metrics associated within a single subject. This approach would allow us to use trained models at the individual level to help in personalizing treatment for addiction to help individuals up or down-regulate areas in the brain that could be playing a role in their addiction. Second, a group-level SVM approach to study whether we can predict brain states on a group level and apply it across subjects. And, finally, a deep-learning approach to group level analysis to see if we can provide better predictions to brain-state data. The brain data collected so far has been collected on a task that captures percent signal change in the Nucleus Accumbens (NAcc) since it has been shown that the NAcc is a key area of interest in the brain associated with the reward circuitry and may play a vital role in drug and alcohol addiction[3]. This approach takes an apriori approach to an area we think may play an active role in addiction. However, we are also interested in whether other regions are involved in addiction, and whether individual differences exist at the group and individual level. Additionally, we want to investigate whether there are key differences in adolescents versus young adults. Because young adults versus adolescents are better able to control inhibition and impulse through their reward circuitry, they are less vulnerable to risk-taking behaviors and are less likely to show substance use

---

[1] Sato, João R., et al. “Real-Time Fmri Pattern Decoding and Neurofeedback Using Friend: An FSL-Integrated BCI Toolbox.” *PLoS ONE*, vol. 8, no. 12, 2013, https://doi.org/10.1371/journal.pone.0081658.

[2] Sitaram, Ranganatha, et al. “Real-Time Support Vector Classification and Feedback of Multiple Emotional Brain States.” *NeuroImage*, vol. 56, no. 2, 2011, pp. 753–765., https://doi.org/10.1016/j.neuroimage.2010.08.007.

[3] Xu, Le, et al. “The Nucleus Accumbens: A Common Target in the Comorbidity of Depression and Addiction.” *Frontiers in Neural Circuits*, vol. 14, 30 June 2020, 10.3389/fncir.2020.00037. Accessed 6 Sept. 2020.

behavior. As adolescents grow more mature, these differences could be heightened or diminished. Meaning, those that show potential for substance abuse, could through maturity, learn to control it, while those that don't, could go on to develop substance abuse disorders. Therefore, we are interested in comparing adolescent brain states with young adults. The product from these analyses would be to develop classifiers able to predict brain states from the images between groups of adolescents and young adults, and within individuals and deploy these models in production to be used in real-time. This would improve the feedback for training to participants in the scanner and help researchers uncover potential ways to use this as therapy for addiction.

TASK DESIGN

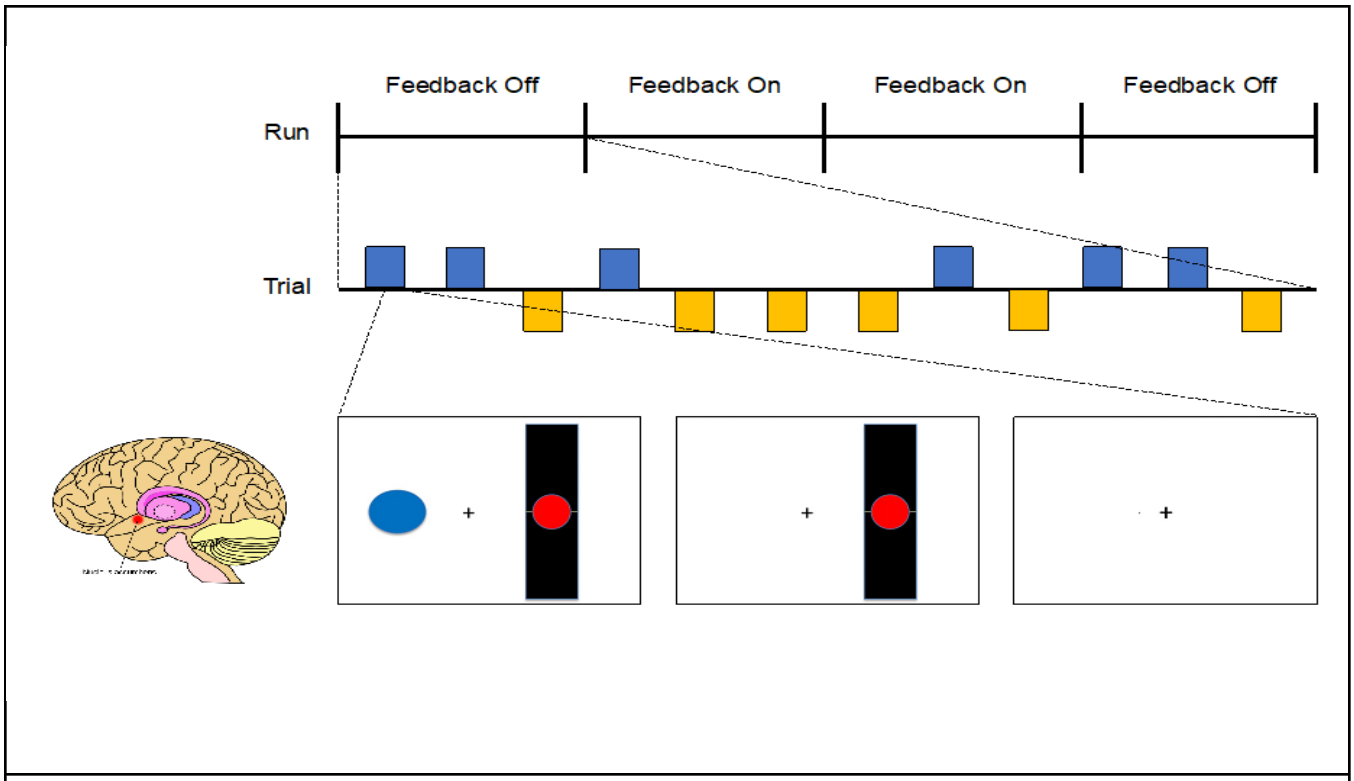


Figure 1: NeuroMod Study real-time fMRI neurofeedback paradigm sequence. The order of run sessions was feedback off, feedback on, feedback on, and feedback off. Within each run, there were 12 pseudorandomly ordered increase and decrease trials. For each run, participants were instructed to try to increase (blue ball cue) or decrease (yellow ball cue) nucleus accumbens activity for each trial, which was linked to the red ball in the thermometer image. During increased trials, participants were instructed to increase nucleus accumbens activity by using high reward-related cognitions (e.g., thinking of enjoyable activities). During decrease trials, participants were instructed to decrease nucleus accumbens activity by using low reward-related cognitions (e.g., thinking of boring or dull activities). Figure and paradigm adapted from Greer et al. (2014).

The data we will be using are rtfMRI images of temporal brain states that are captured during tasks that specifically target the NAcc (Figure 1). Subjects are taught to increase or decrease their brain activity in the reward system through a real-time NAcc task that calculates the percent signal change of the NAcc during the task. The task consists of four runs. Run 1, no feedback, the participant only sees a cue to up-regulate or down-regulate the NAcc. Run 2, feedback, the participant, is given feedback based on the percent signal change of the NAcc, Run 3, is another feedback run and the participant is given feedback on whether they are up regulating or down regulating, Run 4, is a no feedback run, the participant will regulate on their own from prior training whether to up- or down-regulate the NAcc.

DATA

The fMRI brain data we used for these analyses was obtained from 52 voluntary, healthy subjects of adolescents ranging from 14 -16 years old (50% female) and young adults from the ages of 25 - 27 years old (50% female) from the University of Michigan Medicine Research lab. The data is proprietary and every measure has been taken to ensure privacy of all subjects. There are 33 adolescent and 19 young adult subjects. The data structure is temporal, such that a single subject MR run contains 144 volumes or time points of full brain image scans. Each scan represents a 3D brain image, where the pixels are 3D voxels. These pixels represent the feature space for our machine and deep learning models. There are 79 voxels in the x direction, 95 voxels in the y direction and 79 voxels in the z direction. The data is gathered and preprocessed in SPM 12 then flattened so that each brain volume of a given run is flattened to 592,895 voxels per time point. Each subject participates in four runs of equal duration, and 84 time points in a given run are dedicated to the subject attempting to down-regulate or up-regulate based on the task given in the MR machine. From the 84 time points, there are 42 labeled time points for down-regulating and 42 labeled time points for up-regulating. The remaining 60 time points are rest periods in the scanner. Therefore, for a given subject there are 576 total brain volumes, 336 of which are used in our analyses and are labeled for regulation. Our total dataset includes 17,472 brain volumes of 592,895 voxels each.

QUALITY CONTROL AND PRE-PROCESSING

All data captured for this analysis went through a stringent quality control and pre-processing pipeline before being uploaded to AWS for further analysis. All pre-processing steps were employed using SPM12 which was first released 1st October 2014 and last updated 13th January 2020, which is a major update to the SPM software, containing substantial theoretical, algorithmic, structural and interface enhancements over previous versions. First, images were manually inspected in their raw form to look for any deformities or anomalies that would make the images unusable for analysis. If a run did not pass the quality control criteria, it was excluded from analysis and the subject was excluded from our dataset. Second, subjects and runs that passed our criteria were put through a pre-processing pipeline using SPM12 (Table 1). Third, individual runs were inspected to make sure co-registration and warping were consistent across structural and functional images. Fourth, since individuals in the scanner may move and introduce noise into the images, we want to exclude runs that have a lot of motion in them. This is done by creating a threshold and looking at motion from one brain image to the next, if a run exceeds the threshold, we exclude it from analysis. Because we only wanted individuals included that had four quality runs, only individuals that passed all of our quality measures were included in the dataset. Finally, runs were flattened to a 2-D array and stored in a mat file to be uploaded to AWS for analysis.

Pre-Processing Steps	
Slice Timing Correction	Unlike a photograph that is captured in a single instance, a full brain image is captured over time. In order to analyze the brain, we need to apply slice-timing correction to be able to analyze the voxels of the brain as though they were captured simultaneously.
Realignment	If a person moves a lot during their scan, they could introduce noise in the data and one image to the next could be shifted. In order to correct this, we need to shift everyone's individual images in each run and over runs to the same space.
Co-registration	In order to put the fMRI images into a common space, we need to have their

	high-resolution structural images aligned with their functional images and this is achieved in this step.
Warping	In order to do between subject analysis, the data need to be in a common space. In our lab, we use the MNI152 brain to change the subject's functional data to a common space.

Table 1. Pre-processing steps. It should be noted SPM runs under MATLAB which requires a license. All of the pre-processing was done outside of this analysis. For more information about SPM and pre-processing: https://www.fil.ion.ucl.ac.uk/spm/doc/

MASKING

Masking fMRI data is an important preprocessing step where regions of the brain can be removed to allow you to focus on regions of interest. The mask data we used exists in 3D with the shape of (79,95,79) and needs to be flattened before application to our brain data. We applied various masks to the original data and utilized a fourier transformation to ensure proper masking coordinates in Python's numpy reshape package matched MATLAB's masking application technique. The first application we used for model training was a whole brain mask to remove areas of the image that were not considered brain. We also studied other masking strategies which remove a certain area of the brain. Such areas that we studied and removed for separate analyses include a submask of the Anterior Cingulate Cortex, Anterior Insula (Right-side), Nucleus Accumbens, and the Medial Prefrontal Cortex. Other masks that we studied included removing all brain data except the area of interest (ROI). The same areas we negated out in the submasks were the regions we kept in the ROI masks (Table 1).

Regions of Interest	**Function**
Anterior Cingulate Cortex	The ACC is known for error detection in tasks and monitoring conflict, alerting other areas of the brain of this conflict to activate a resolution.[4] It is also an area of the brain involved in memory and motivation, decision-making, and cognitive inhibition. These functions are important in understanding substance use disorders.[5]
Anterior Insula (Right Hemisphere)	This is an area of the brain thought to be involved in various brain disorders and serves as an area of the brain to create subjective feelings or the ability to sense oneself. This is important for understanding how one prioritizes stimuli and salient information, and thus impacting motivation. With substance abuse, users seek hedonic experiences, influencing this motivational salience. It is also an area of the brain linked to the ACC. [6]
Medial Prefrontal Cortex	Research has shown the Medial Prefrontal Cortex to be inhibited in subjects with substance use disorders.[7] It is an important part of self-regulation, inhibition, decision-making and conflict resolution/monitoring. [8]
Nucleus Accumbens	The Nucleus Accumbens regulates depression and addiction and is the reward

[4] "Anterior Cingulate Cortex." *Wikipedia*, 16 Apr. 2020, en.wikipedia.org/wiki/Anterior_cingulate_cortex.

[5] Zhao, Yijie, et al. "Anterior Cingulate Cortex in Addiction: New Insights for Neuromodulation." *Neuromodulation: Technology at the Neural Interface*, vol. 24, no. 2, Feb. 2021, pp. 187–196, 10.1111/ner.13291. Accessed 24 Apr. 2022.

[6] Namkung, Ho, et al. "The Insula: An Underestimated Brain Area in Clinical Neuroscience, Psychiatry, and Neurology." *Trends in Neurosciences*, vol. 41, no. 8, Aug. 2018, pp. 551–554, 10.1016/j.tins.2018.05.004. Accessed 28 Mar. 2021.

[7] Volkow, N. D. "Activation of Orbital and Medial Prefrontal Cortex by Methylphenidate in Cocaine-Addicted Subjects but Not in Controls: Relevance to Addiction." *Journal of Neuroscience*, vol. 25, no. 15, 13 Apr. 2005, pp. 3932–3939, 10.1523/jneurosci.0433-05.2005. Accessed 22 Oct. 2019.

[8] Euston, David R., et al. "The Role of Medial Prefrontal Cortex in Memory and Decision Making." *Neuron*, vol. 76, no. 6, Dec. 2012, pp. 1057–1070, www.ncbi.nlm.nih.gov/pmc/articles/PMC3562704/, 10.1016/j.neuron.2012.12.002.

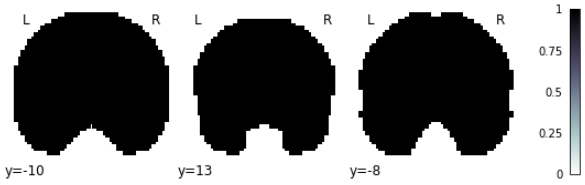
	center of the brain that activates dopamine and pleasure. This area of the brain has been studied as one activated in people with substance use disorders.

MASKING

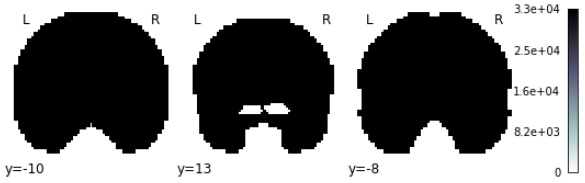
Masking fMRI data is an important preprocessing step where regions of the brain can be removed to allow you to focus on regions of interest. The mask data we used exists in 3D with the shape of (79,95,79) and needs to be flattened before application to our brain data. We applied various masks to the original data and utilized a fourier transformation to ensure proper masking coordinates in Python's numpy reshape package matched MATLAB's masking application technique. The first application we used for model training was a whole brain mask to remove areas of the image that were not considered brain. We also studied other masking strategies which remove a certain area of the brain. Such areas that we studied and removed for separate analyses include a submask of the Anterior Cingulate Cortex, Anterior Insula (Right-side), Nucleus Accumbens, and the Medial Prefrontal Cortex. Other masks that we studied included removing all brain data except the area of interest (ROI). The same areas we negated out in the submasks were the regions we kept in the ROI masks. See Figure 2, Appendix for ROI details.

Figure 2.

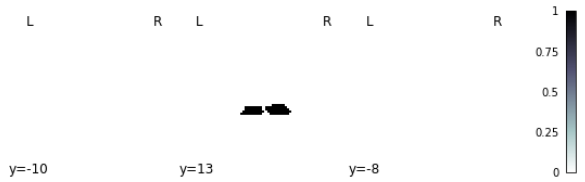
Whole Brain Mask



Sub Mask of the Nucleus Accumbens



ROI Mask of the Nucleus Accumbens



All brain masks were created outside of the repository using SPM 12. Whole brain mask was created using the single subject T1 supplied by SPM. We used IMCALC to set dimensions outside of the brain to values of 0, and

masks within the brain to a value of 1. Individual ROI masks were created using aal toolbox for SPM12. Once masks were created, we had to run the spm reslice tool to get masks into our correct dimensions (79,95,79) which are different from the default (91,109,91). Once all the masks were in the correct dimension, I used IMCALC, a function in SPM12, to subtract the individual ROIs to give us our masks that had the voxels containing the ROIs masked out. Once we had all the masks, we created a mat file that could be read from python. We also created the labels to be used in our analysis. These labels were created by prior knowledge of the timing of the task and the task set-up.

DATA STORAGE AND ENVIRONMENT

We used various types of data storage in this project. Once the lab preprocessing was completed, we uploaded the data to an AWS S3 bucket where each subject's data, label data and mask data are stored in their own folders. We would access the data from AWS to further process the data to apply masks to each image as well as filter the time points of the image by labels. Once these processes were completed we uploaded these files back to AWS. Eventually, it would make more sense to run the further processing steps in real-time since we had to make various changes to the data in the processing steps and it was not feasible to upload back to AWS after each process iteration. In some cases, we were able to upload model and metric data to AWS. Due to pickle file formatting issues, we opted to save models locally, and some metric data back to the cloud.
We set up our work environment initially by connecting to a Docker container in Pycharm Student Edition, working locally and within Jupyter Notebooks, pushing code to Github. Once we realized the size of our data, we started to build using a virtual environment provided to us with a subscription to Google Colab Pro+. All data training and analyses require enabling 'High-RAM' in the runtime settings, which provides 51GB of memory. Each step of the single subject, group-level and deep learning data pipelines is organized in python files and imported into Google Colab for execution and demonstration. The python files are organized by project analysis type and task as are the notebooks.

NORMALIZATION

Before we began training with machine learning models, we researched the most commonly used strategies for fMRI data normalization and consulted with domain experts. We decided to perform an exploratory analysis on the different normalization strategies applied to fMRI data (Figure 2).

Within these analyses, we plotted voxel distributions on subjects split by age groups and looked at distributions across MR scan runs. We also performed cross validation on each different normalization strategy and compared the accuracy results. These processes were repeated twice as we later realized from our research on fMRI analyzes that the temporal aspect of fMRI data requires detrending as the data drifts over time points. We looked at the voxel distributions on non-detrended unnormalized data, detrended unnormalized data, detrended percent signal change normalized and detrended Z-score normalized data. The non-detrended, unnormalized data was not normally distributed, and that adding a normalization technique to the data would improve the distribution. Detrending unnormalized data surprisingly improved the variance of the data, as did applying percent signal change to detrended data. However, the data in these graphs are still noisy, so we lastly applied Z-score normalization to the detrended data and found that this distribution was the least noisy with lower variance (Figure 3, Appendix I).

Accuracy Scores of Normalization Strategies

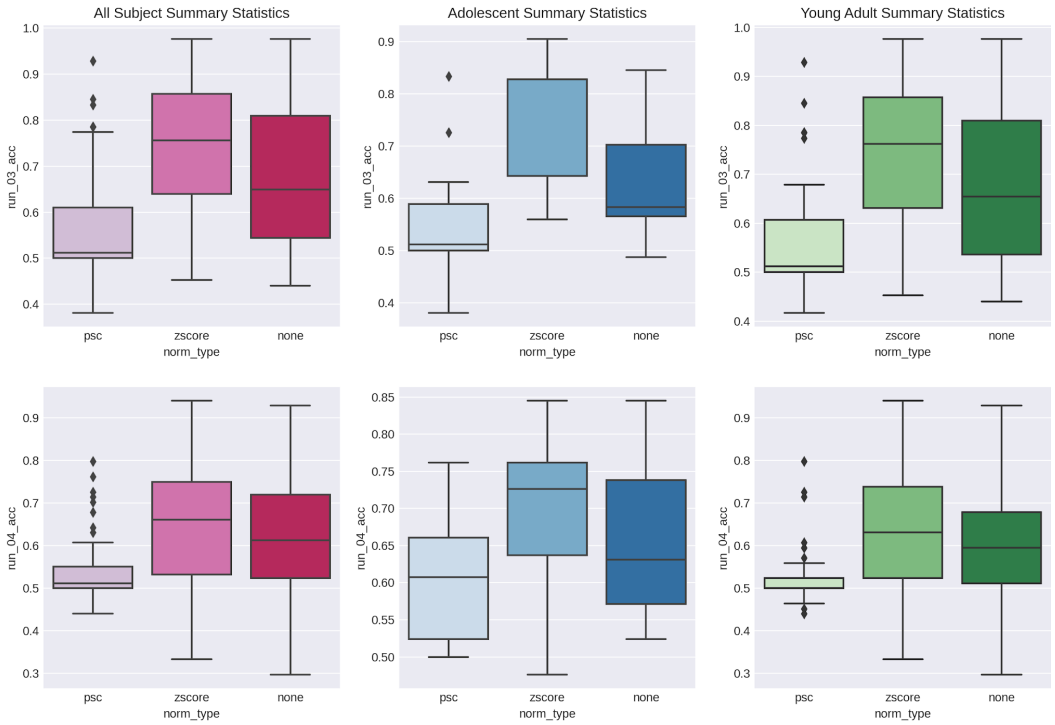
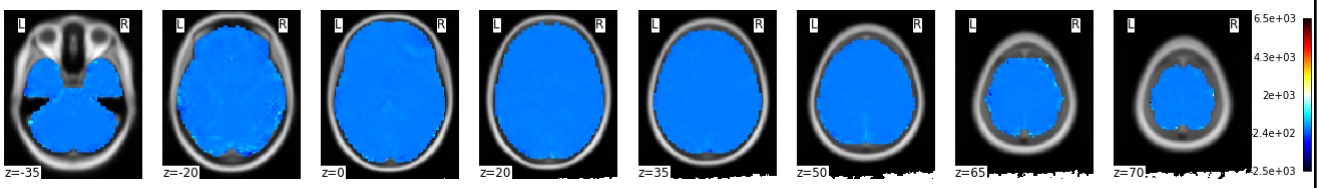


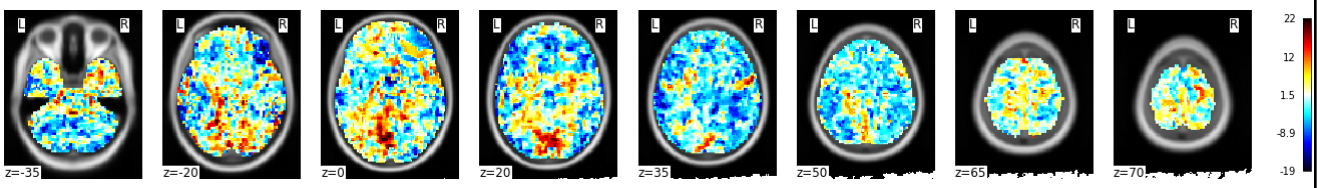
Figure 2. As seen above, we can see that z-normalization on detrended data out performs percent signal change and no normalization.

Brain Images of Normalization Strategies

Percent Signal Change



Z-Normalization



No Normalization

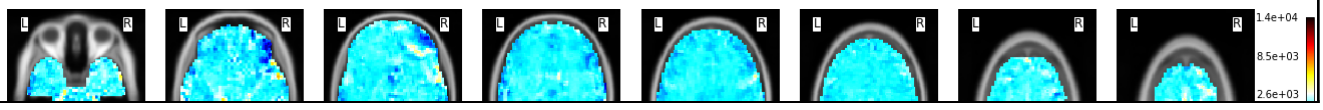


Figure 3. As can seen from the three panels above, z-normalization out performs both no normalization and percent signal change. The voxels of the brain are more distinguishable in the z-normalization plus detrending.

SINGLE SUBJECT SVM

Methods

Data Processing and Model Building

The single-subject level pipeline reads in each subject's data, the masking data and the label data from AWS using a dictionary of key paths. We then applied the mask and label data to each subject's run data separately which resulted in each run containing 84 timepoints and 237,979 features. Once we got our final masked data per run, we detrended and z-score normalized the time series voxels using Nilearn's signal clean package. The detrending and normalization was done per subject per run.

Once the data was normalized, we built our models training on run 2 using Scikit-Learn's svm.SVC classifier. The parameters we chose were found during the cross-validation phase of the single-subject data exploration. The final customized model parameters we chose were C=5, gamma='auto', and kernel='rbf.

Cross Validation

To ensure we were getting the best parameters for our models, we chose to use Scikit-Learn's GridSearchCV. Data for the cross validation grid search was processed the same way as for model building. We chose to only look at whole brain masking in the cross-validation and apply to all masked models. Each individual's run 2 went through a 5-fold cross validation and the data was summarized across all individuals to find the best parameters for use across all subjects (Figure 4 and Figure 5).
We also chose to summarize the data and not customize the parameters per subject which could be an area for further research.

Cross-Validation Exhaustive

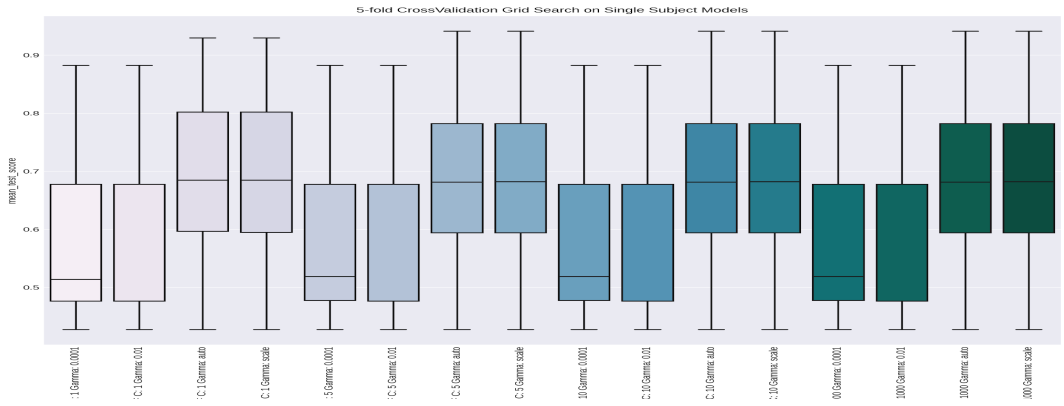


Figure 4. Cross-Validation across varying kernel types, C values, and Gamma values. From this you can see that RBF performs the best for single subject SVM with a C=5 and gamma could be either auto or scale.

Cross-Validation Condensed

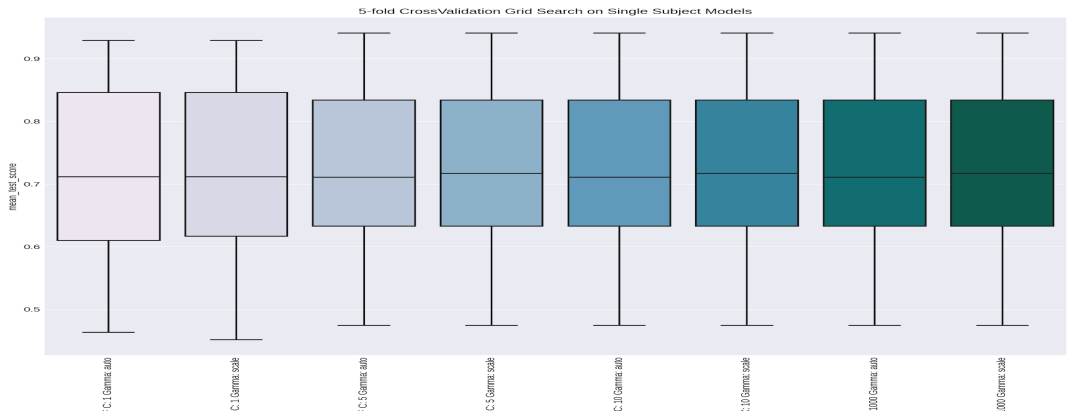


Figure 5. Cross-Validation on just RBF, with varying C values, and Gamma values. From this, we chose C = 5 and gamma = auto as they outperformed all other models.

Single Subject SVM Results

Being able to individualize SVM for use on individuals that may utilize different brain regions for use in up and down-regulation is an important data-driven approach to studying addiction. Through this we may be able to customize therapies for individuals to help aid in recovery from substance abuse disorders. Through this analysis, we were able to explore whether we could predict brain states on the individual level. As seen in figure 1, there are four distinct runs in the task design. For this analysis we chose to use run 2 as our training run as this was the run that provided the first feedback as to whether the individuals were up or down-regulating their NAcc. We also chose to test separately on run 3 (another feed back run) and run 4 (a no feedback run). This would give us insight into whether there were differences in individuals ability to maintain what they had learned in the feedback runs. Further analysis was done looking at the different masks which either removed ROIs from brain data or only trained on ROIs to see if areas that have been studied and thought to be important in substance abuse disorders had an effect on the model's ability to predict brain states in individual subjects.

As can be seen in figure 6, we see that there is a lot of variation in model performance across individuals. Some of the individual model's performed very well at predicting brain states while other's individual models performed were at or below chance. This could be due to the fact that there are individual differences in how well an individual is at regulating their brain states and need further training on how to regulate. We investigated the models performance looking at sensitivity and specificity which looks at how well the model predicts the positive and negative class respectively. In the single subject models, the mean sensitivity and specificity were very similar which suggests the model was pretty good at distinguishing the two classes when it was able to predict well.

Mean Metrics for Single Subject SVM for Run 3 Feedback Run

Mask Type	Run	Accuracy	Sensitivity	Specificity	Precision	AUC
mask	run_03	0.74	0.741	0.739	0.746	0.809
masksubACC	run_03	0.741	0.74	0.742	0.748	0.808
masksubAI	run_03	0.741	0.742	0.739	0.746	0.808
masksubNAcc	run_03	0.74	0.74	0.739	0.746	0.809
masksubmPFC	run_03	0.74	0.741	0.739	0.746	0.808
acc_aal	run_03	0.68	0.695	0.664	0.678	0.737
anterior_insula_aal	run_03	0.613	0.632	0.595	0.611	0.644
mPFC	run_03	0.612	0.634	0.588	0.61	0.655
nacc_aal	run_03	0.636	0.641	0.631	0.636	0.681

Mean Metrics for Single Subject SVM for Run 4 No Feedback Run

Mask Type	Run	Accuracy	Sensitivity	Specificity	Precision	AUC
mask	run_04	0.663	0.665	0.662	0.665	0.712
masksubACC	run_04	0.665	0.667	0.664	0.667	0.712
masksubAI	run_04	0.664	0.665	0.663	0.666	0.712
masksubNAcc	run_04	0.663	0.665	0.663	0.666	0.712
masksubmPFC	run_04	0.664	0.665	0.663	0.666	0.712
acc_aal	run_04	0.602	0.623	0.58	0.599	0.647
anterior_insula_aal	run_04	0.557	0.563	0.554	0.557	0.578
mPFC	run_04	0.57	0.588	0.552	0.567	0.592
nacc_aal	run_04	0.569	0.577	0.561	0.572	0.592

Table 2: Shows mean metrics measures for feedback vs. no feedback runs used to test the model. Across all metrics and different masking strategies, we can see that the model was better able to predict on the feedback run.

Single Subject Model SVM Boxplots Showing Variation of Metrics of Individuals

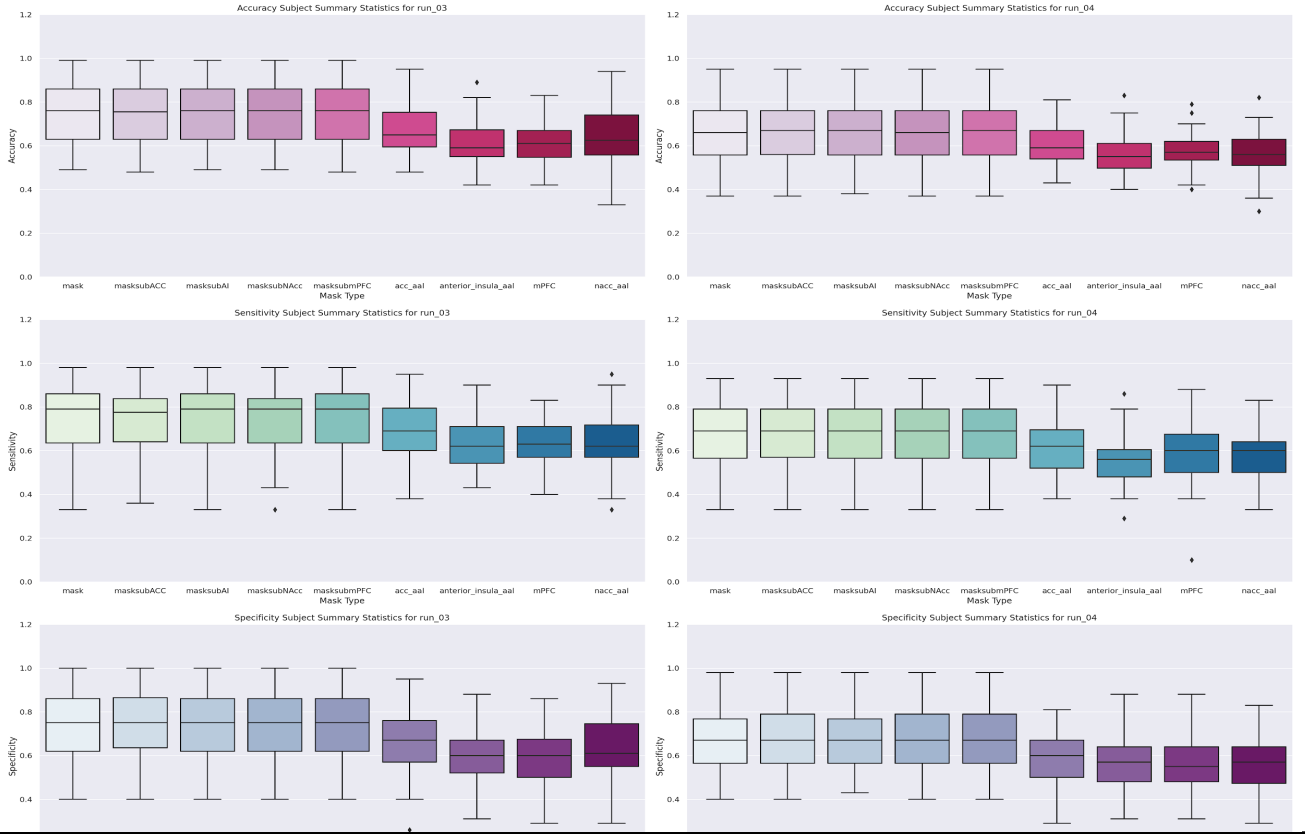


Figure 6. Boxplots showing variation in the predictions for single subject models. Top Panel is showing accuracy scores, middle panel is showing sensitivity scores, and the bottom panel is showing specificity scores. The left panel shows the metrics on feedback run and the right panel shows metrics on the no feedback run.

We wanted to also look at how removing different ROIs from the brain affected prediction. The subtraction of ROIs didn't affect model performance suggesting that there may be individual differences in areas of the brain a person uses to control their brain states (Table 2, Figure 6). Furthermore, individual ROIs showed a decrease in all metrics which may further suggest that there may be more regions than just that interconnect when a person tries to regulate their brain state.

Investigating the ability to maintain up or down-regulation of brain states, we analyzed performance of the predictions over the feedback vs no feedback runs. Individual models were better able to predict on the feedback run versus the no feedback run in all our metrics (Table 2). This could suggest that the individuals may need more training to maintain reward regulation when not being given feedback. The ROC curves and AUC scores show that in the no feedback runs, the model performance did not evaluate as well as in the feedback run(Table 2 and Figure XX).

ROC Curves for Single Subject Models for High and Low Performing Models

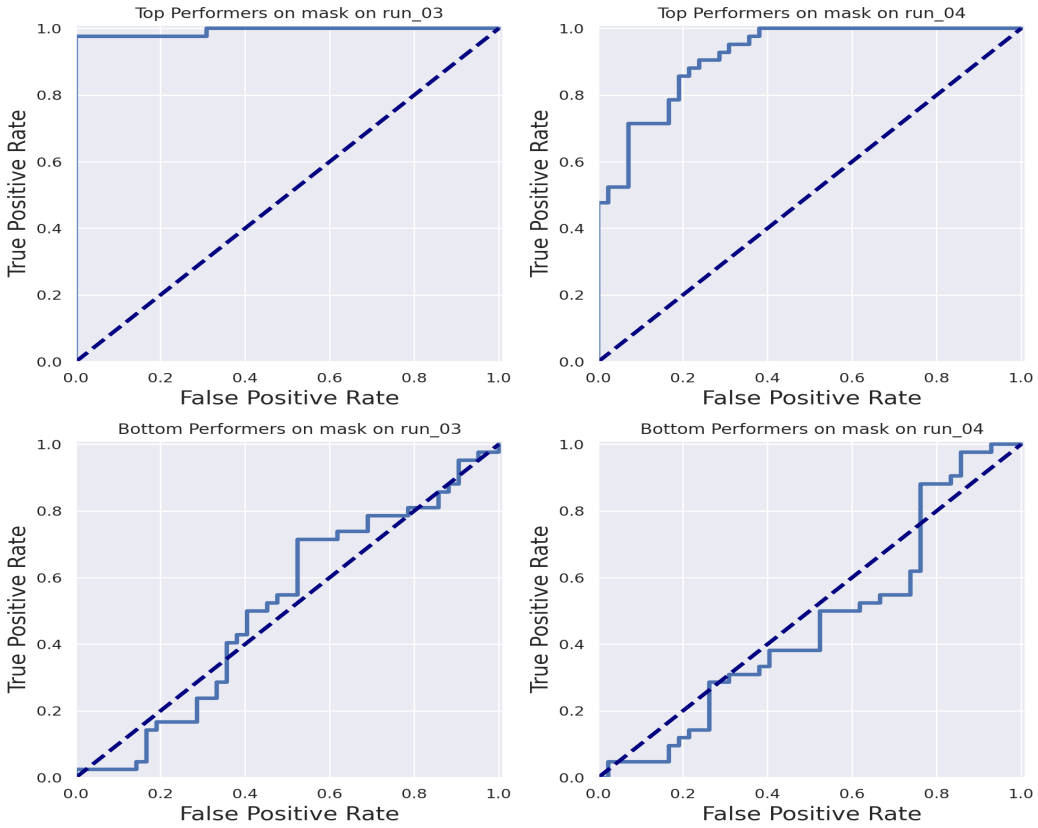


Figure 7. ROC Curves on for high and low performing models as well as looking at differences in the models ability to predict between feedback and no feedback runs.

In order to understand how individual models performed differently between individuals we explored data from an individual that had good model evaluation and an individual that had bad model evaluation. We used different visualization and metric techniques to explore these differences. We looked at model ROC curves and AUC scores to compare model performance between these individuals for the whole brain masked models. ROC curves showed that the model was unable to distinguish well between the two brain states for models that perform poorly.

To understand how the individual models were differing, we looked at how the model predicted labels for the up and down-regulation classes. In the models that were able to predict well, the model was able to evenly classify both up and down regulation brain states. The poor performance model had trouble classifying both up and down, but had a harder time distinguishing the positive class. This could suggest that this person was not able to properly regulate their brain state. We also show that the model was able to more accurately predict both the positive and negative class in the feedback run versus the no feedback run. Again this suggests individuals having a harder time carrying over from feedback runs to no feedback runs.

Confusion Matrix for Single Subject Models for High and Low Performing Models

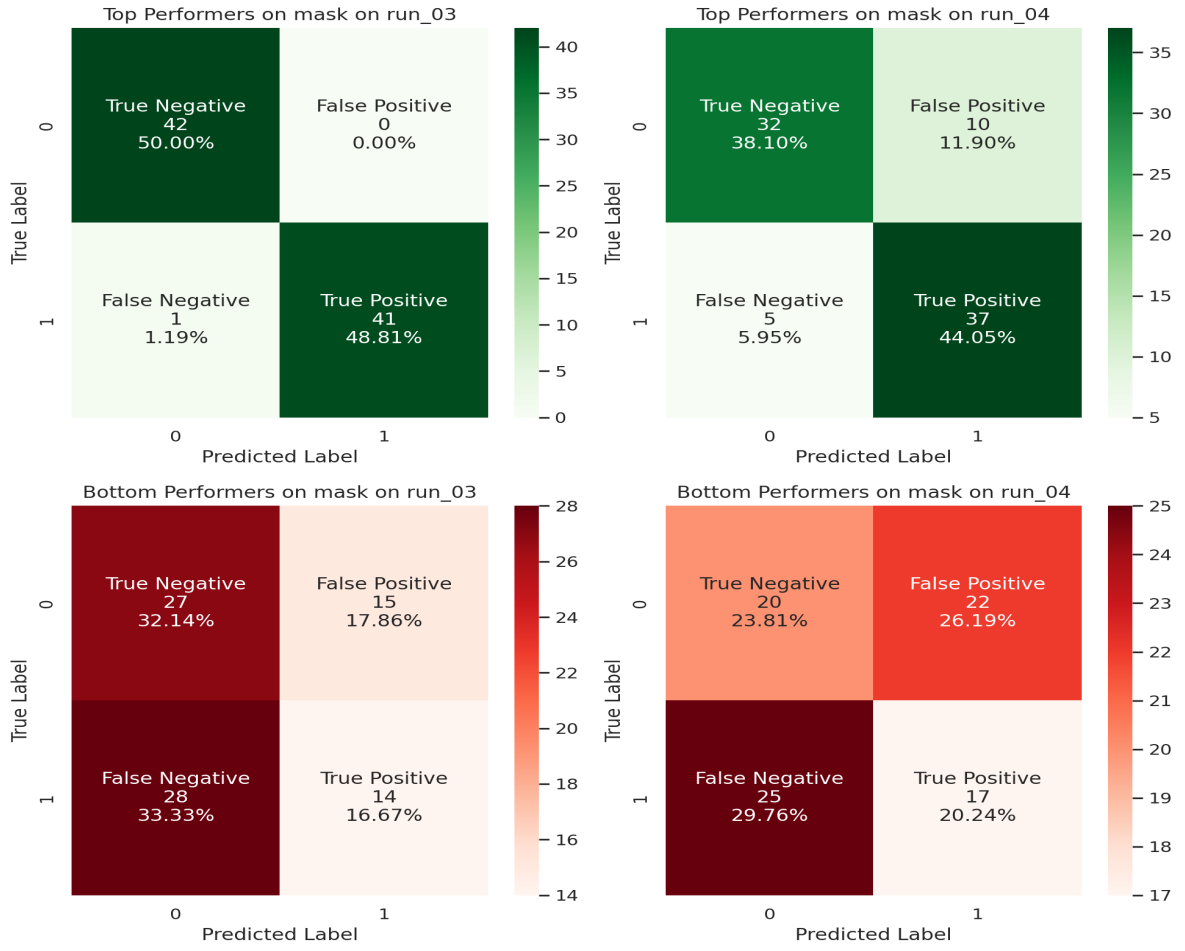


Figure 8. Confusion Matrix for high and low performing models showing how well the classifier predicts on the positive and negative classes.

In order to understand how the classifier was making its decisions on the two individual models, we explored the difference in decision function scores,which are measurements of the distances of the significant voxels from

separating hyperplanes the classifier was using to make its predictions. We plotted these against the true labels over the time course. You can see that the low performing models had a harder time distinguishing between the two classes (shown in the bottom panel of figure). We also can see that it had a harder time distinguishing the no feedback run (seen in the right side of figure). Which may suggest that individuals are not able to carry over their training from the feedback runs. You can also see how different the brain maps which are used to show how significantly different regions of the brain map on to up and down-regulation states.

Decision Function Scores for High and Low Performing Models

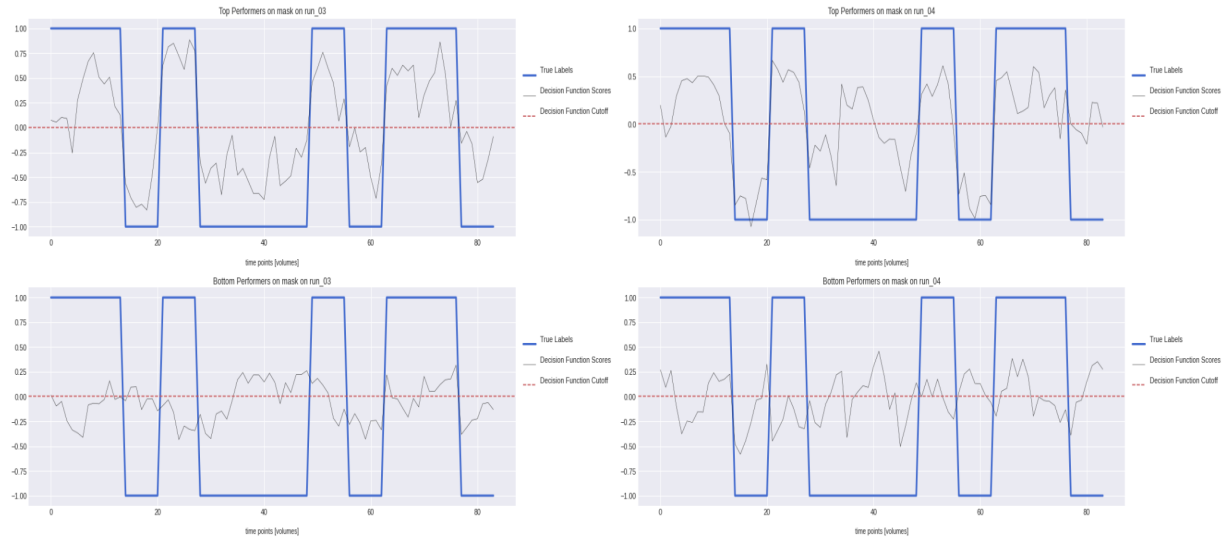


Figure 9. Decision Function Scores for top and low performing models. Top panel shows decision function on a top performer and the bottom panel shows decision function on low performers. The left panel shows feedback run and the right panel shows no feedback run.

Visualize Voxel Patterns for Whole Brain Masks for High and Low Performance Models

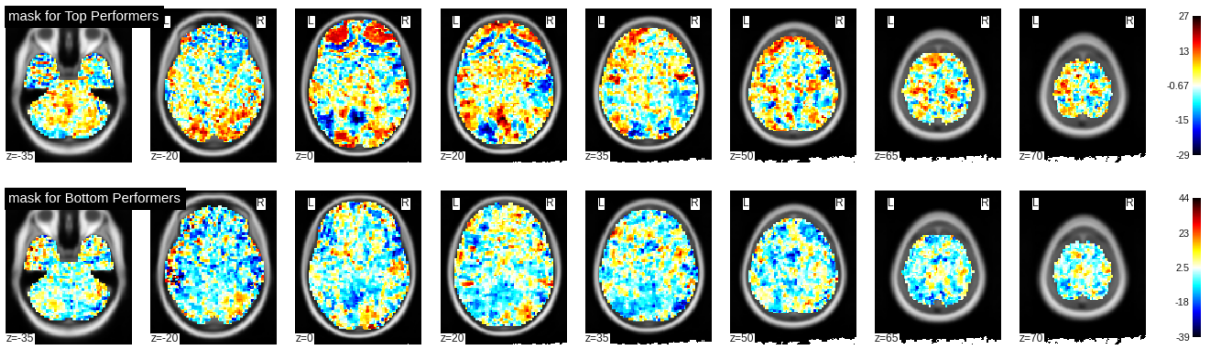


Figure 10. Showing br voxel distributions across the whole brain. The top panel shows distributions for the top performing model and the bottom panel shows voxel distributions for the low performing model. The high performer has more clustered voxels than the low performing model. This suggests that the individual had greater control in up and down regulating their brain states.

GROUP LEVEL METHODS AND RESULTS

Data Processing, Preparation and Model Building

The group level pipeline reads in subject, mask and label data from AWS using a dictionary of key paths and then applies the specified mask to the data and filters out the resting time points per subject per run. The data for group analyses is split between adolescent and young adult. Adolescent subject Ids begin with a '1' and young adult Ids begin with a '3'. Once groups are separated, we perform an 80-20 split for training and test data and do this using subject Ids to ensure that each dataset contains unique Ids to prevent data leakage. Adolescent training is performed on 26 subjects and tested on 7 subjects for runs 2 and 3 in the scanner. Young Adult training is performed on 15 subjects and tested on 4 subjects on runs 2 and 3.

Once the data is split, masked and labeled filtered, it is then detrended and scaled using z-score normalization using Nilearn's signal.clean package by subject by run over the time points. The training and test subject timepoints are concatenated separately. This provides a data shape of (4368, 237979) and (1176, 237979) for adolescent training and test subjects, respectively. Young adult data shapes after concatenation are (2520, 237979) and (672, 237979) for training and test sets, respectively. We chose to use Scikit-Learn's svm.SVC classifier for model training with parameters for class balance, max_iter of 1000, probabilities as True, and gamma and regularization parameters of 5, 'scale' and 10, 'auto' for adolescent and young adult, respectively. The data metrics are saved as a dictionary in a pickle file and uploaded to AWS. The models are saved locally to Google Drive.

Time Series Cross-Validation with Halving Gridsearch

Preparing the data for cross-validation was performed differently than for model training. We wanted to ensure that there would be a held out test set for model training that would not be included in the cross-validation study. We separated the data on subject Ids between adolescent and young adult subjects and split the data by subject Ids into 80% training for cross-validation and 20% for held-out test sets for each group. In the adolescent group, we further split the training set by 80% training for a total of 20 subject Ids and approximately 20% validation set of 6 subject Ids. For young adults, we split the original set of 19 subjects into 80% training and 20% held out test data. Of the 80% training, 12 subjects were assigned the training set, and 3 subjects assigned for the validation set. Each split throughout was on unique subject Ids to prevent data leakage. The data was then processed for detrending, z-score normalization, and all the subjects were concatenated for the training as well as for validation sets.

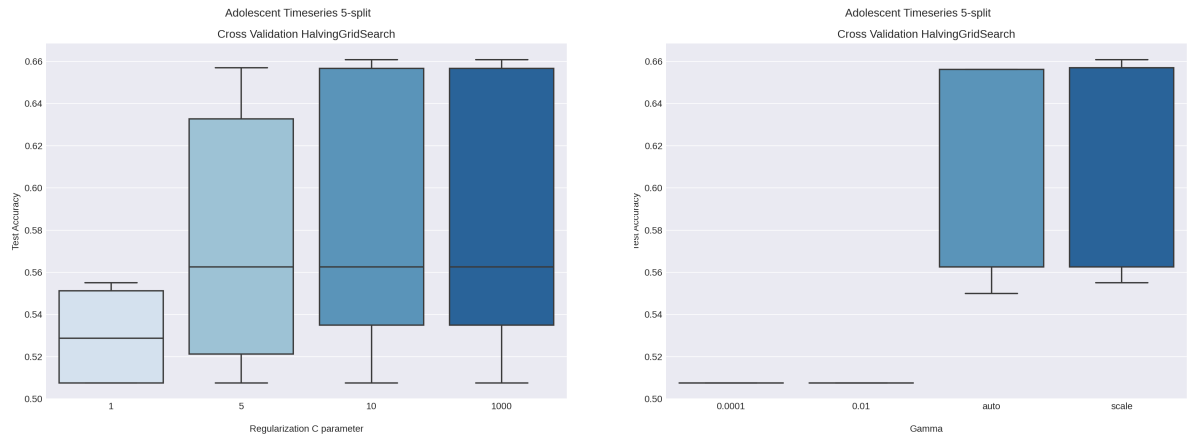
Time series cross-validation was the method used at the group level since the same subject data is not included in both the training and test sets. We used Scikit-Learn's TimeSeriesSplit package to perform the 5-fold split of the data. This method consists of splitting a data set into a training set of size n, with the max size defined prior to calling the function, for observations that occurred prior to the time points for the validation set of size n_samples /(n_splits +1).[9] This method preserves the time series aspect of the data and shuffling is not performed. This prevents the model from observing future time points during training, providing a more accurate prediction on the validation set. The data was then fed into Scikit-learn's HalvingGridSearch package for hyperparameter tuning.

[9] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html

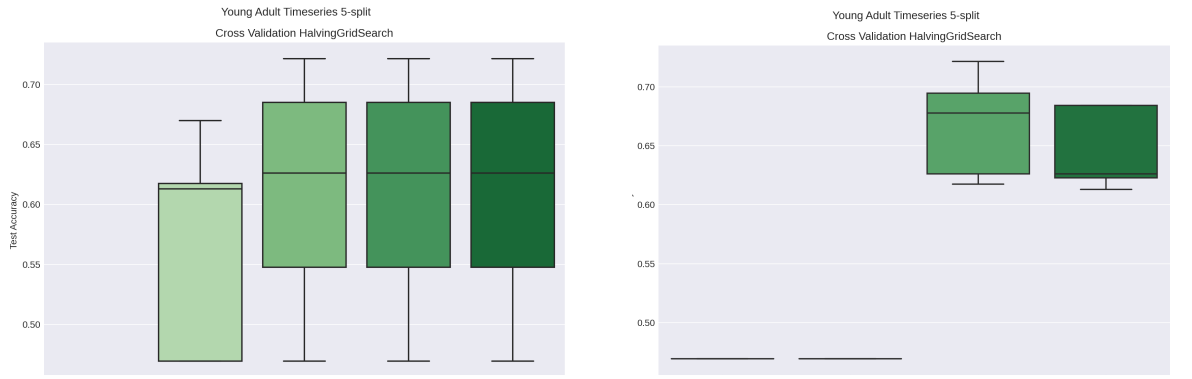
Cross-Validation Results

The parameters were chosen in the cross-validation grid search for both adolescent and young adults using Scikit-Learn's svm.SVC API. Since we iterated this process more than once due to fundamental changes in the data we needed to make in regards to detrending and normalization, and where the time to complete averaged 6 hours per group, we limited the number of parameters tested and did not fine-tune the search as much as we originally expected. We did observe through these iterations that a radial basis function kernel consistently outperformed the other optional kernels. The regularization C parameter 5.0 and 10.0 for the adolescent group had overall better accuracy scores, though the spread was larger, likely due differences between subjects. The medians of both scores are the same with less spread for C = 5, therefore, C parameter of 5 and the gamma parameter 'scale,' which uses a value of 1/(n_features * X) or 1/(237979 *(n_time_points, 237979).var()) are chosen for the final model. The young adult group C parameters after 5 statistically appeared the same and so choosing the regularization value was more decided on trial and error testing. We found 10.0 to work with a gamma parameter 'auto,' which uses 1/n_features or 1/237979.[10] See Figure 6.

Adolescent HalvingGridSearch C and Gamma Boxplots



Young Adult HalvingGridSearch C and Gamma Boxplots



[10] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

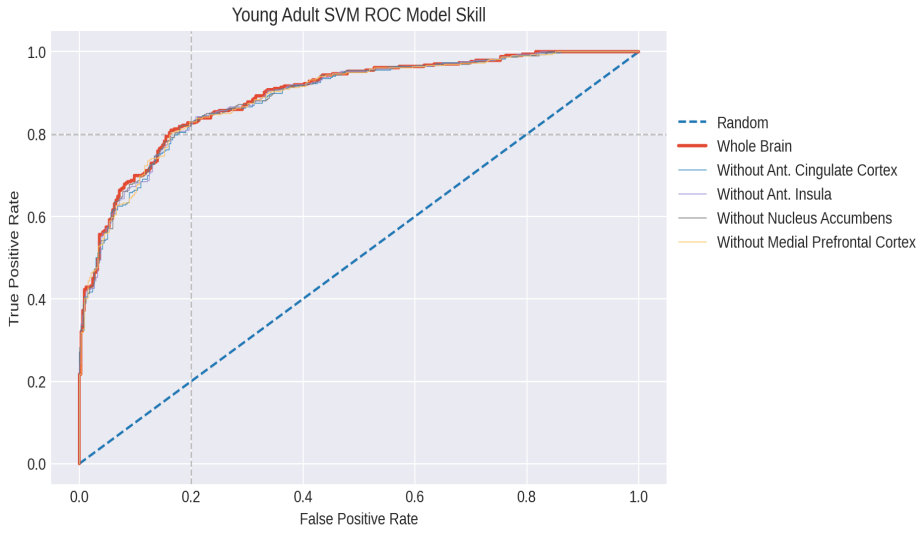
Figure 6. Adolescent C parameters have large spreads beyond 1, though the medians are higher. We use C = 10 for both Adolescent and Young Adults. Gamma ‘auto’ for Young Adult and ‘scale’ for Adolescent.

Adolescent and Young Adult Results

We chose to train our group level models on brain scan runs 2 and 3. These runs contain data where researchers provide subjects in the scanner with neurofeedback to practice up-regulation and down-regulation strategies while looking at a brain computer interface to stimulate the reward centers in the brain. Run 2 is the first time when subjects are given feedback. Run 3 is when subjects are provided with feedback again, though this time it is thought they have already been able to learn or practice strategies since this is their second exposure. Further analyses at the group level could look at training and testing on other runs to determine if subjects carried over their learning strategies using the neurofeedback from previous runs. We were able to achieve this first step in our analyses.

We looked at model ROC curves and AUC scores to compare model skill between the whole brain mask and all the submasks for young adults and adolescents as well as comparing whole brain mask model skill between the two groups. Young adults score better than adolescents with an average score of 89% and 82%, respectively. The whole brain model underperformed compared to the submasks in the adolescent group and all models performed similarly in young adults.. See Fig 7.

Receiving Operating Curve within Groups and Between Groups



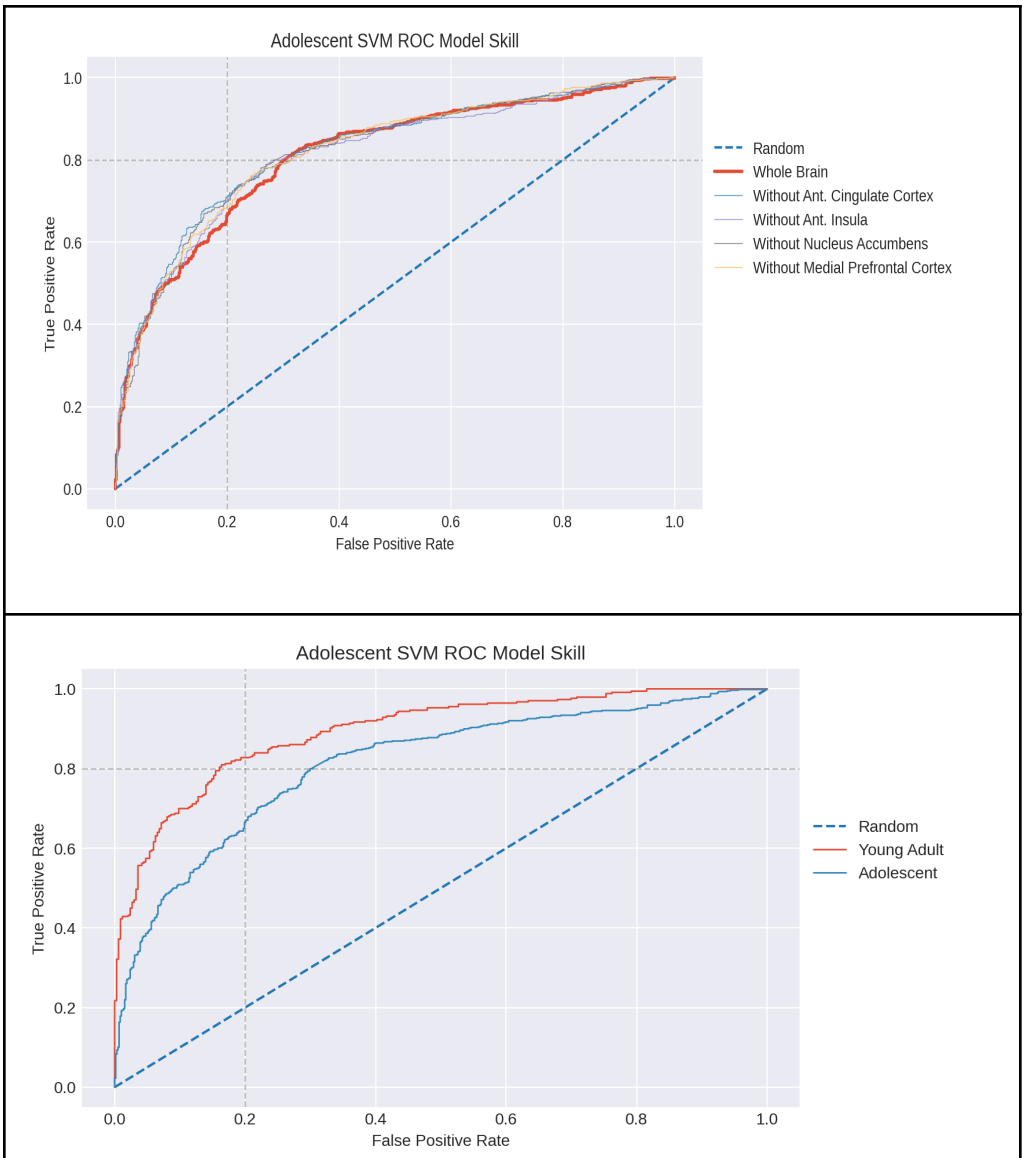


Figure 7. Young Adult models perform better than Adolescent models

We chose to look at the whole brain mask to compare label predictions between adolescent and young adults. Both group models predicted the negative class, down-regulation, more poorly than the positive class, up-regulation. In the adolescent group, there's an 82% chance that the subjects are up-regulating and a 69% chance that they are down-regulating when expected. The young adult group performed better in sensitivity and specificity, where 82% of subjects were up-regulating and 81% were down-regulating when expected. See Figure 8.

Confusion Matrix Comparison of Young Adult and Adolescent

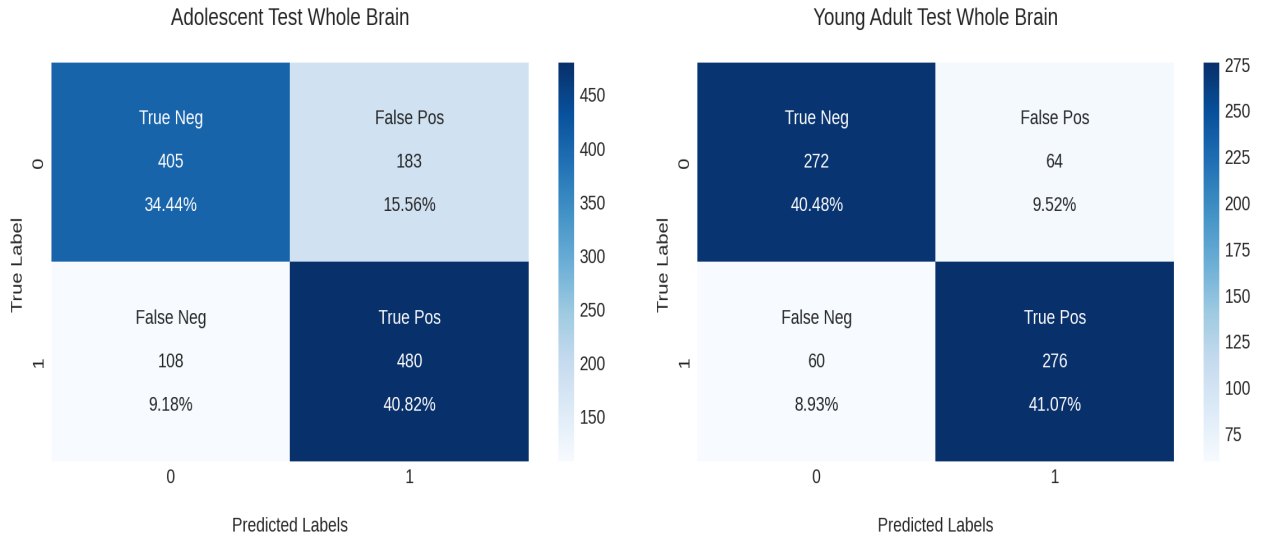
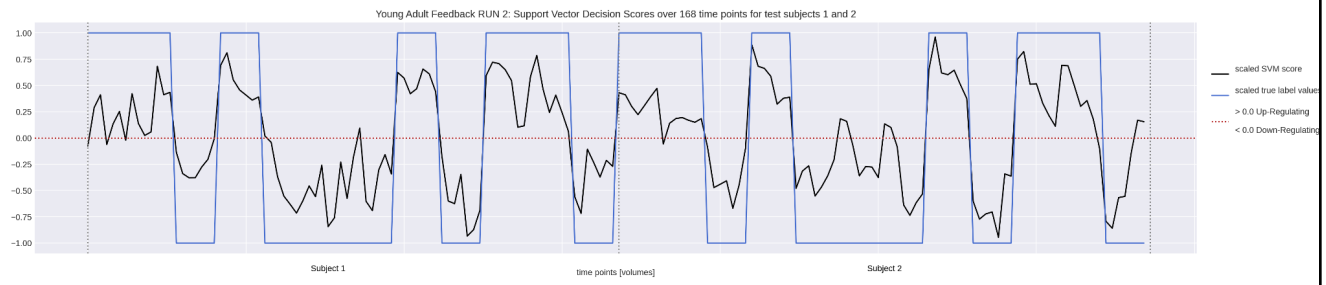


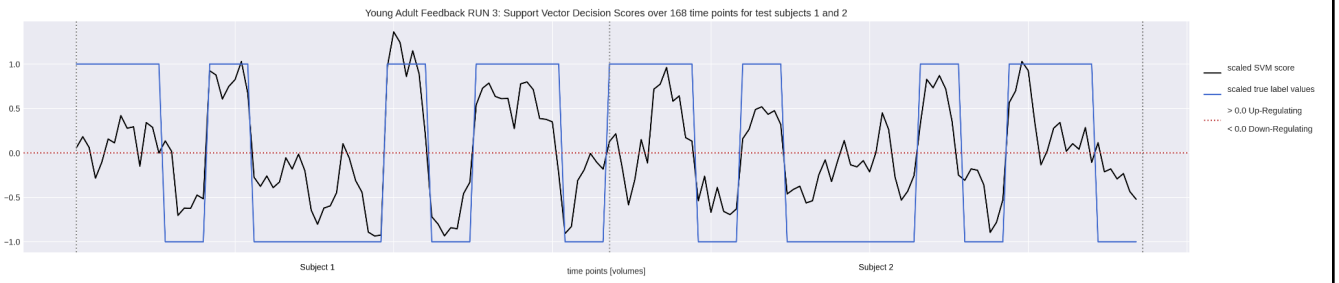
Figure 8. Young adults expect to up-regulate similarly to adolescents. The expected outcome with adolescent down-regulation is 17% less than young adults.

Since our data feature space is highly dimensional, we are not able to visualize the hyperplane and decision boundaries of the SVM classifier. In order to appreciate the decisions made by the classifier, we looked at decision function scores, which are measurements of the distances of the significant voxels from separating hyperplanes. Below are examples of young adult and adolescent decision scores plotted against time and true labels. We chose to show an example of how two subjects might differ from one another as well as differ between themselves from Run 2 to Run 3 where feedback from the task might be carried over. Some subjects are able to regulate better than others, such as young adult to adolescent subjects, as well as, adolescent subject 4 versus subject 5. Interestingly, we see an improvement in both of these subjects from Run 2 to Run 3, which may suggest some level of carryover from one feedback run to the next. See Figure 9.

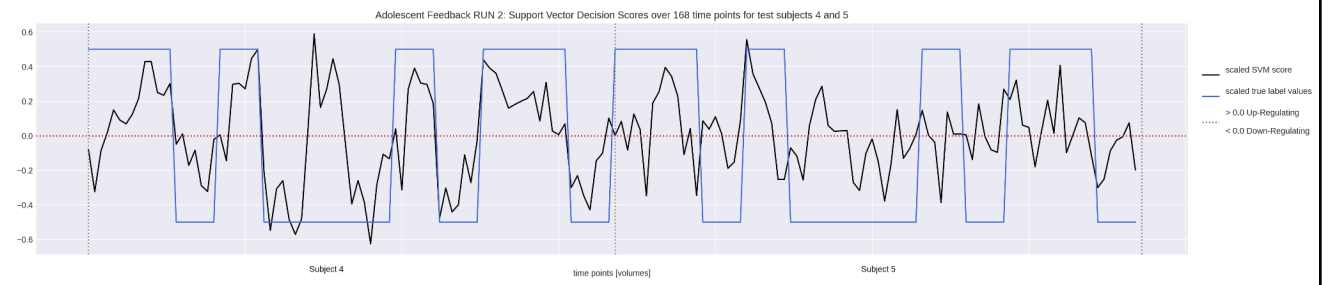
Run 2: Young Adult Support Vector Decision Scores for subjects 1 and 2



Run 3: Young Adult Support Vector Decision Scores for subjects 1 and 2



Run 2: Adolescent Support Vector Decision Scores for subjects 4 and 5



Run 3: Adolescent Support Vector Decision Scores for subjects 4 and 5

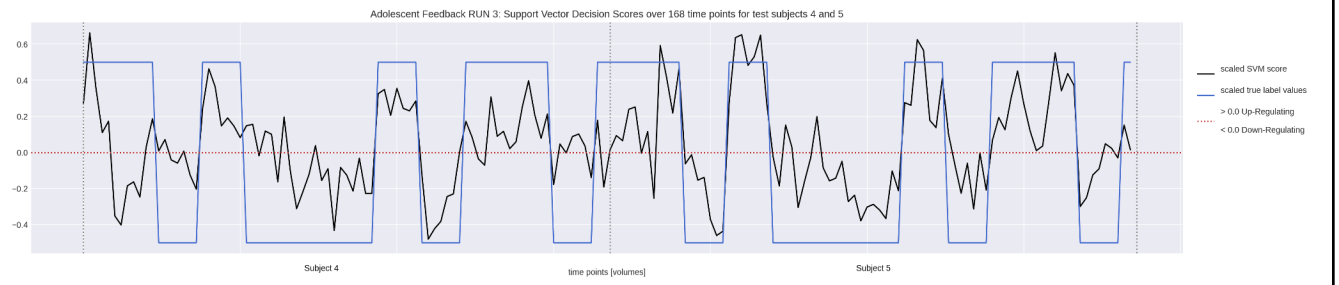


Figure 9. You can see how the adolescent subjects 4 and 5 appear to learn to regulate and carry over this knowledge from runs 2 and 3.

We can summarize all of our observations together in a histogram of the decision scores to represent the boundaries that the classifier created to make predictions. True-positive predictions sit to the right of the center of the x-axis, true-negative predictions sit to the left, while areas that overlap between -0.5 to 0.5 have been misclassified, (brown shaded region between down-regulating and up-regulating). Overall, you can see how the SVM is better at separating the decision boundaries for young adults versus adolescents. The histogram for young adults overlaps less frequently between classes, where the overlap's spread appears similar to adolescents. See Figure 10.

Histogram representing the Decision Boundary of the SVM using Decision Function Scores

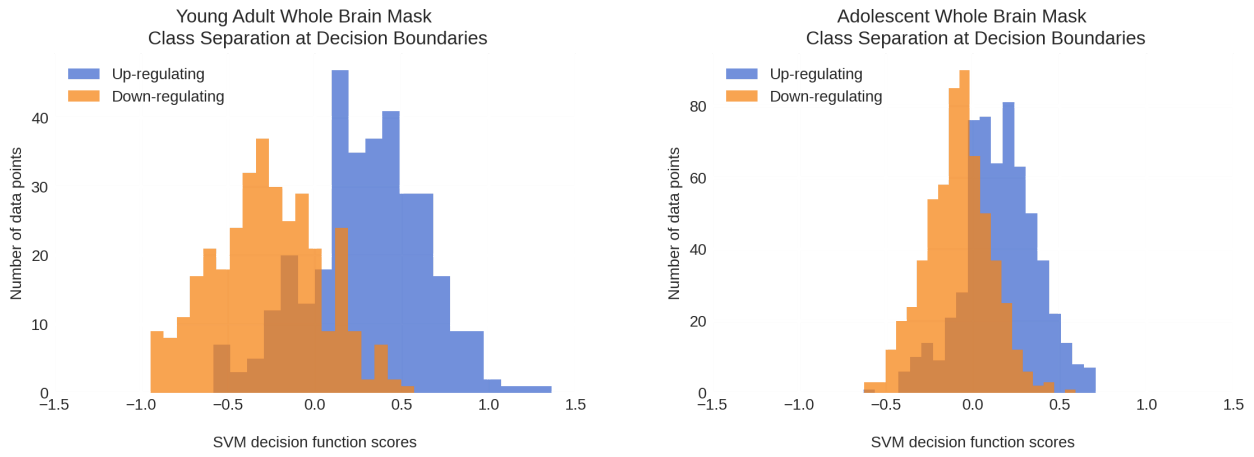


Figure 10. Better separation of decisions made by the SVM in young adults versus adolescents

Analysis beyond metric comparisons at the group level looked at statistical tests of the decision function scores and beta maps between adolescents and young adults as well as within groups on the whole brain mask, submasks and regions of interest. The beta maps represent the applied support vector voxel weights to the data at the voxel indices that correspond to the support vectors (voxels that are most significant and sit near the decision boundaries). Of the 27 tests on decision scores, we show the most notable significant differences, which were between groups in the Anterior Insula, Medial Prefrontal Cortex as well as the whole brain mask without the Anterior Cingulate Cortex. It does not seem clear to compare the significant scores we obtained from tests comparing the whole brain to the region of interest since we discovered model predictions on regions of interest to be 30% lower on average than prediction scores on the whole brain. What is most notable from these tests are the differences between young adult and adolescent decision scores and beta maps for the Medial Prefrontal Cortex and the Anterior Insula. Further analysis is needed to understand how these regions of interest are different from each group and within groups. See Figure 11.

Statistical tests between groups

Two Tailed T-Test Between Young Adult and Adolescent Decision Scores

Region	Test Statistic	p-value	alpha	Hypothesis
Medial Prefrontal Cortex decision score	3.32825	0.000891	0.05	Reject Null H0
Anterior Insula (Right side)	2.309826	0.021008	0.05	Reject Null H0
Whole Brain without the Anterior Cingulate Cortex	-1.997647	0.045901	0.05	Reject Null H0

Two Tailed T-Test Between Beta Maps for Young Adults and Adolescents

Region	Test Statistic	p-value	alpha	Hypothesis
Whole Brain Beta Maps	62.00712	0	0.05	Reject Null H0
Medial Prefrontal Cortex from Whole Brain Beta Map	-4.1317	3.6011e-5	0.05	Reject Null H0
Nucleus Accumbens from Whole Brain Beta Map	-1.44904	0.14733	0.05	Fail to Reject Null H0
Anterior Cingulate Cortex from Whole Brain Beta Map	16.0073	1.15223e-57	0.05	Reject Null H0
Anterior Insula (Right) from Whole Brain Beta Map	0.26111	0.79401	0.05	Fail to Reject Null H0

Figure 11. Further analyses is needed to investigate how the AI differs between groups

DEEP LEARNING METHODS AND RESULTS

Data and Preprocessing

Convolutional neural networks (CNNs) are the most widely used deep learning models in computer vision and imaging tasks. We used Pytorch, a useful library to build CNNs for 3D imaging tasks.

We built the model on runs 2 and 3 for all subjects. These runs were set aside because in these runs, patients received real-time feedback in the scanner of their ability to up and down regulate. This creates a more consistent model premise to train, validate, and test the CNN. CNNs also need lots of data to train and can pick up complex

features in images. To increase the number of images given to the CNN, we decided not to create separate models for young adults and adolescents and instead take runs 2 and 3 for all subjects, regardless of their age group. This was a reasonable choice because CNNs have the ability to pick up features in the data, like age, that are too complex for non-deep-learning models.

The preprocessing pipeline for deep learning was slightly different than single subject and group model preprocessing regarding masking and preparing images for modeling. Researchers have found success in previous deep learning real-time fMRI imaging projects (Wang 1509) using full brain masks, and we decided to format our images similarly. This involved the same normalization and detrending seen in single and group models with the addition of demasking and casting images back into their original 3D space.

We split 52 subjects into training, test, and validation sets. The training, validation, and test sets contained 36, 5, and 11 subjects respectively as a 70%, 10%, and 20% split. We split the data by subject to avoid data leakage. Only 36 subjects and 72 runs in the training set is not enough data to train by run. Instead, we decided to increase the training data size and train by individual image and label. This allowed us to have 6048 total labels and images in our final training dataset. With the images and labels separated, the data was model ready and saved in AWS S3.

CNN Training

The CNN parameters that we chose were based on another fMRI CNN research paper (Wang 1509). The structure of the CNN can be seen in the image below.

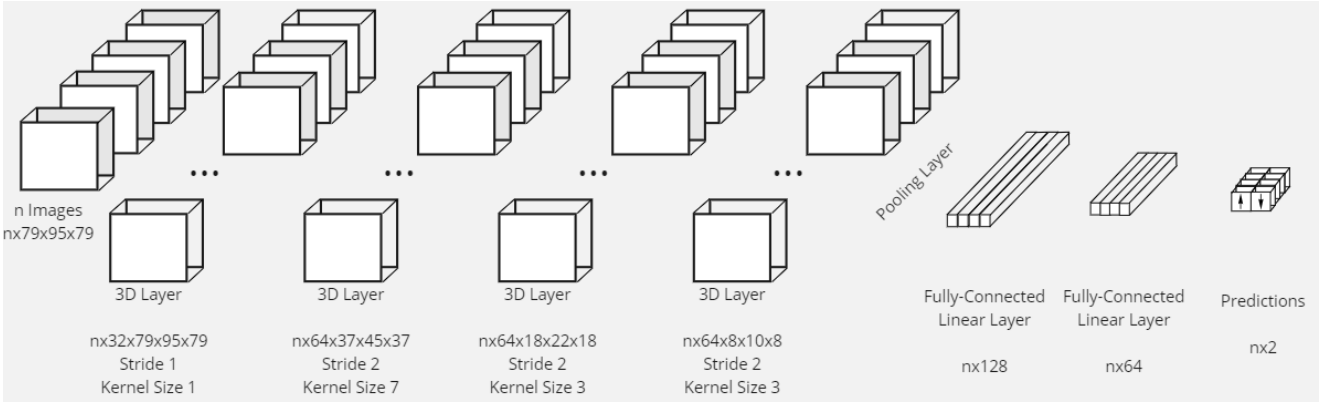


Figure 12: CNN Structure

To start, we trained the data in 8 partitions of 756 images with 10 epochs for each partition. Due to the large size of our dataset, the complexity of the CNN, and the limit of 50GB of RAM in Google Colab Pro+, we decided to train the data in 8 partitions. This involved random shuffling of individual images and labels and loading them into a Pytorch dataloader object. You can see the training results in Figure 13. As a note, some of the partition metrics were deprecated, returning only metrics on partitions 1, 3, 7, and 8. This was one of the many struggles that we had with technology. Although each partition's first epoch's accuracy was under 70%, the initial partitions of data trained much slower than the later partitions. We implemented early stopping when any batch of images returned perfect predictions in hopes to avoid model overfitting.

With the first round of 8 partitions and 10 epochs complete, it was clear that the model was overfit to the later partitions. To generalize the model, we trained the model a second time on each partition for a single epoch. The accuracy and spread of the predictions decrease from early to late partitions. This could either be because the model is able to generalize better on new data or the model remembered the later partitions better than the earlier partitions. When we ran the validation set through the model, we saw similar accuracies to partition 1. This confirmed that the CNN continued to overfit the later partitions of the data.

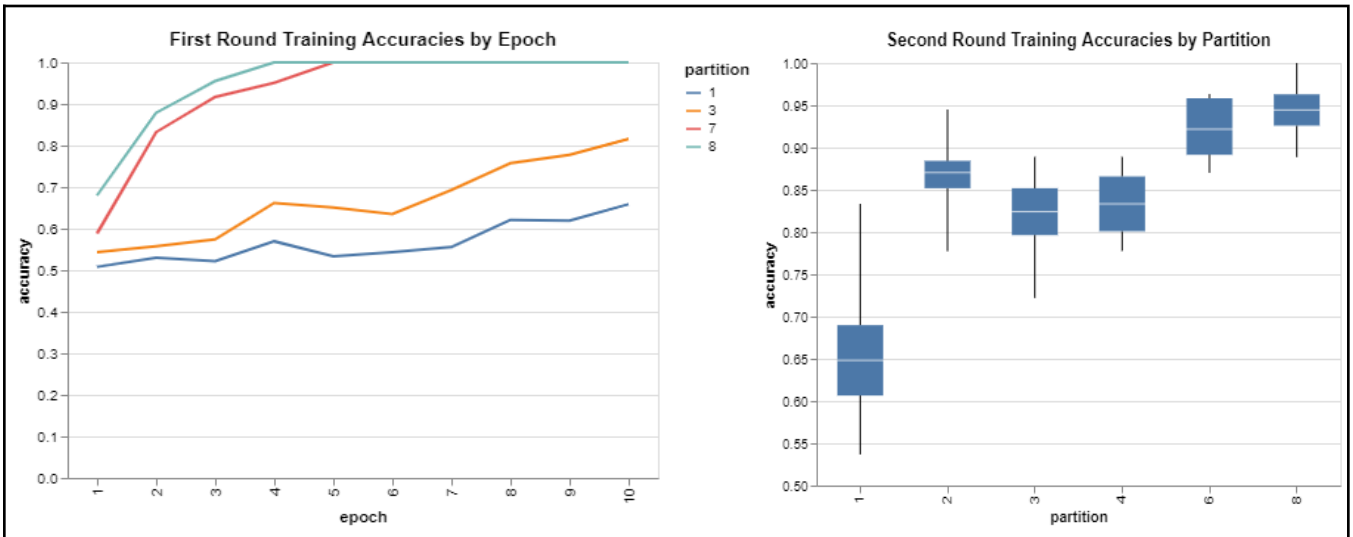


Figure 13: Training Process in First and Second Rounds

Test Results

We then took the 11 subjects in the test set and predicted each run individually through the trained CNN. 6 of the subject runs 2 and 3 are shown in the graph to the right, while the metrics from the other 5 were deprecated. We see here that the trained CNN had inconsistent accuracies among subjects, ranging from below random prediction accuracies to 85% accuracy. The losses were equally inconsistent. Despite these inconsistencies, each subject generally had comparable losses and accuracies between runs.

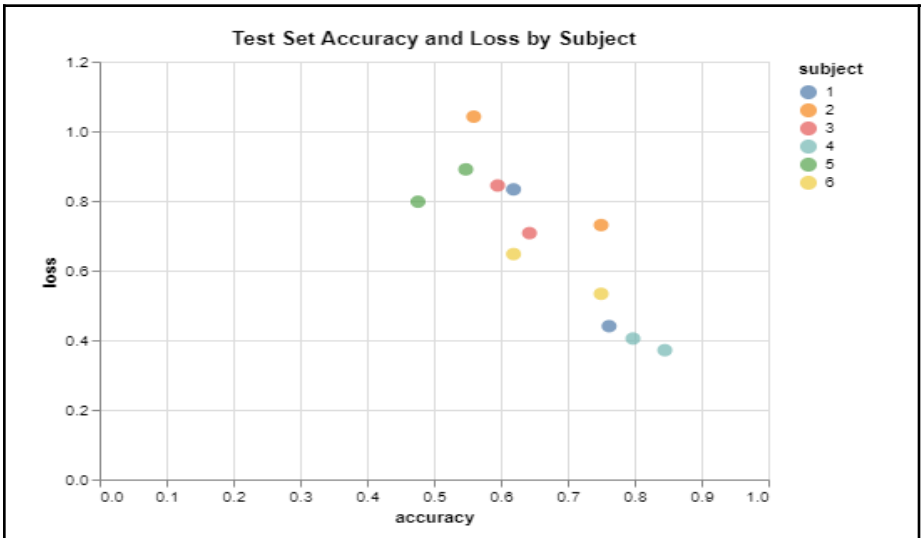


Figure 14:
Results of test subjects accuracies and loss by run.

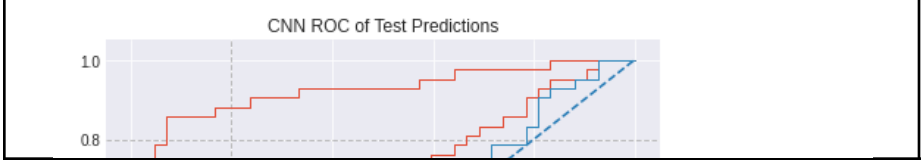
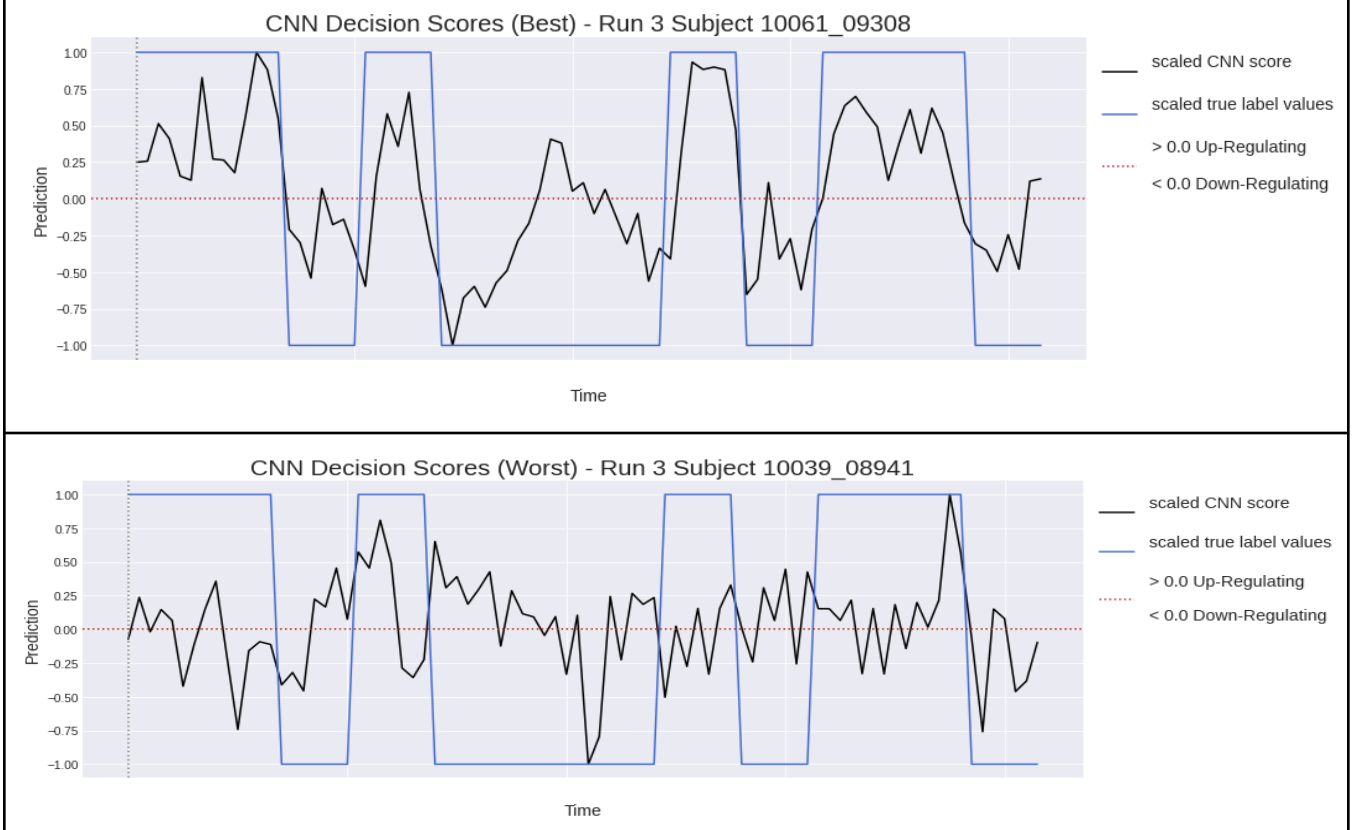


Figure 15:
ROC AUC Curves, representing best, average, and worst subject runs.

Finally, to better understand the results and the models ability to predict a subject's regulation in the scanner, we selected the best and worst subject runs by accuracy. In the ROC AUC curve to the right, we see that the best prediction has an even curve, indicating that true and false positives are best distributed with around an 85% accuracy, which is very promising. In the worst test prediction, we see that the outcome follows the random line, indicating that there is little to no model indication of this subject's ability to up and down regulate. Below, we see two decision function graphs that compare model prediction with a subject's ability to up and down regulate. In the best test prediction, the model clearly understands the subject's ability to up and down regulate except for a small down regulation in the middle of the run. The worst prediction does the opposite, and is predicting inconsistently with low prediction values. This is an indication that our CNN model makes volatile predictions on unseen data.

Figure 16: Best and worst decision functions



CONCLUSION

We have only begun to understand how single subject models can be trained using SVM and used in real-time to predict brain states for individuals trying to up and down-regulate the reward system. There are a lot of potential avenues to take when looking at these types of real-time classifiers. Our analyses failed to show ROIs and ROI sub-masks that may potentially play a role in the reward circuitry as being important predictors of up and down-regulation brain states. This is likely to do with the complexity of the brain and how different regions of the brain work together to control our emotional states and drive our reward system. Although we had true labels of up and down-regulation, we are not certain that all individuals will actually perform the task the way we would expect. Individuals are highly unique and the approaches for controlling the reward system could be different between individuals. We do know they are getting feedback based on percent signal change in the NAcc and future work could be to analyze subjects prediction scores against their percent signal change neurofeedback data.

Future work in single subject SVM model training would include using the program SearchLight supplied by nilearn to create F-statistics and probability maps based on the classifier built by SVM to see if individuals that are successful at up and down-regulation have specific regions they are employing that differ between individuals.

Like in the single subject analyses, we have determined that we are able to take a machine learning approach using a SVM classifier to study at the group level. We were able to observe differences in model training between young adults and adolescents as well as differences in the ability to up and down-regulate. We also noted significant differences in statistical scores between adolescents and young adults at the group level when comparing decision scores as well as beta maps for the Medial Prefrontal Cortex and Anterior Cingulate Cortex and the Anterior Insula. These findings should be further analyzed in SearchLight as previously mentioned. Although we focused mainly on whole brain comparisons of the two groups, we also looked at masking out regions of interest in the brain as well as regions of interest. Further model development and analyses for the other mask models is needed to form robust conclusions between young adults and adolescents in this study to better understand regions of interest and influence in substance use disorders.

The convolutional neural network we built in this situation underperformed when compared to the support vector machines models. It had lower prediction accuracies overall and greater volatility. This comes as no great surprise because CNNs need lots of training data. In many CNN projects, there are thousands or millions of images, and likely six thousand training images is not enough to build a consistent classifier. It is also difficult to avoid overfitting the model with such a small set of training images.

In conclusion, CNNs have lots of potential for better accuracy in the future. As the size of training data increases, better models will likely emerge. To increase the size of data, it may be possible to use data augmentation techniques. Similarly, using transfer learning techniques on previously trained models in fMRI and then parameter tuning with our data may also increase the accuracy of a CNN, although it would involve a much more complex data pipeline.

STATEMENT OF WORK

Stacey Rivet Beck set up the repository and Docker container, storage bucket and group access in AWS, wrote various functions for accessing/loading and uploading data to and from AWS, created the group-level analyses pipeline from accessing the data in AWS, cross-validation, normalization exploration, model training, brain visualization and metrics analyses. She also set-up and added to the landing page, attempted to rework a pre-trained deep learning model to run with the project data, researched and built out the deep learning model architecture, participated in each general aspect of the write-up as well as full group-level write up.

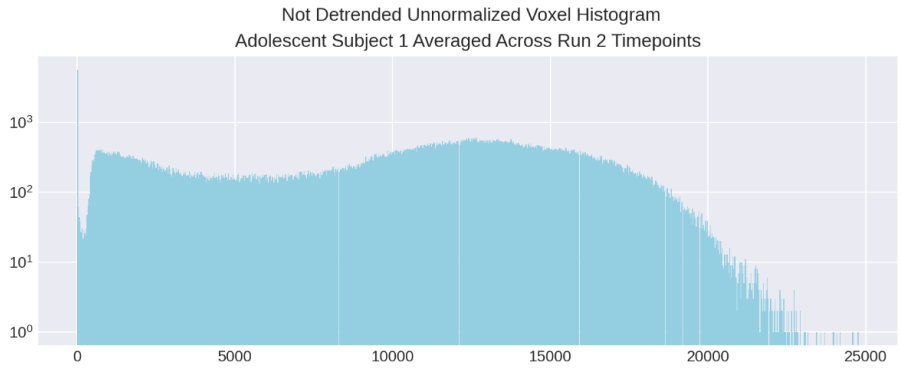
Ben Merrill created the deep learning portion of the study. He created a modified pipeline to turn data into a Pytorch compatible format in AWS, dealt with scalability issues related to training the CNN and computing hardware, and created a framework for deep learning metrics and interpretation. He participated in group aspects of the project including peer review, project process and workflow, and creation of many video presentation slides.

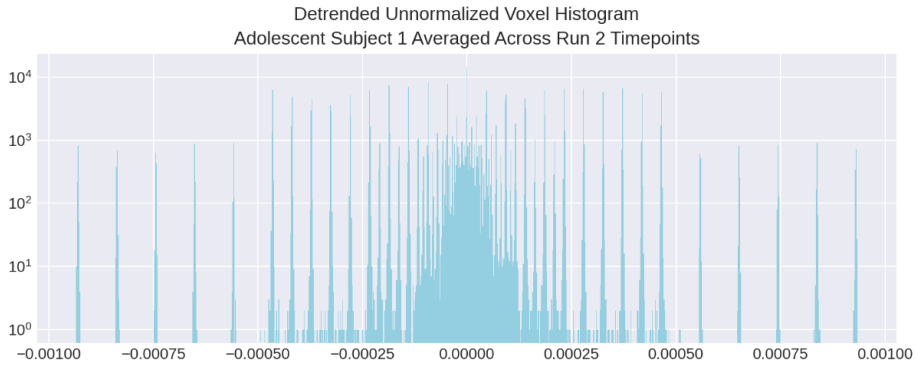
Mary Soules contributed domain knowledge of real-time NAcc task design and fMRI domain knowledge. She supplied knowledge on how to translate SVM outputs to brain visualizations and explored how to visualize these outputs in python. She did all the quality control and preprocessing of the data to get ready for analysis. Once she created the final dataset to deploy for the project, she wrote a script to flatten the 4-D images to a 2-D matrix for each subject run and saved these data in a file that could be read from python and uploaded all data to AWS. She created all masks, sub-masks, and ROIs and uploaded those to AWS. She created the single-subject pipelines for accessing data in AWS, cross-validation, normalization exploration, model training, brain visualizations, and metric analysis. She also participated in the general aspects of the write-up as well as the full single subject SVM write up.

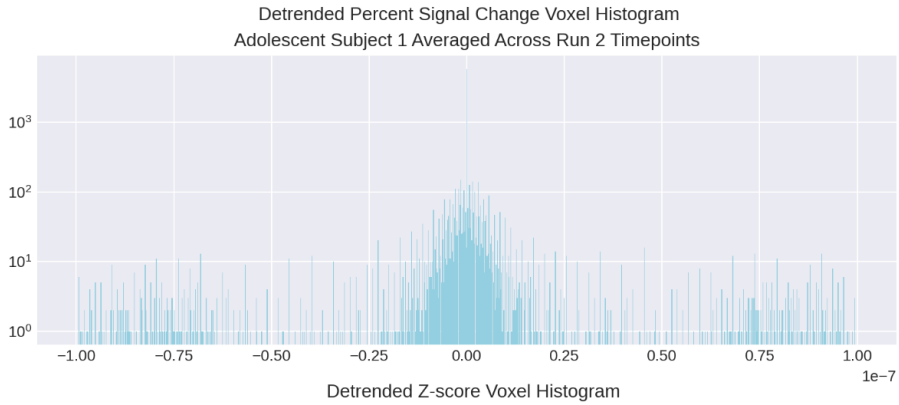
REFERENCES

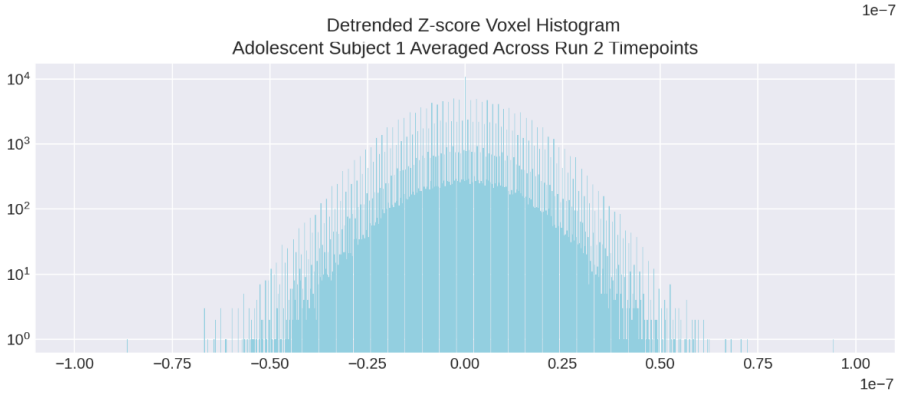
Wang, Xiaoxiao. “Decoding and mapping task states of the human brain via deep learning.” Hum Brain Mapp, vol. 41, no. 10, 2020, pp. 1505-1519. National Library of Medicine, https://arxiv.org/ftp/arxiv/papers/1801/1801.09858.pdf.

APPENDIX

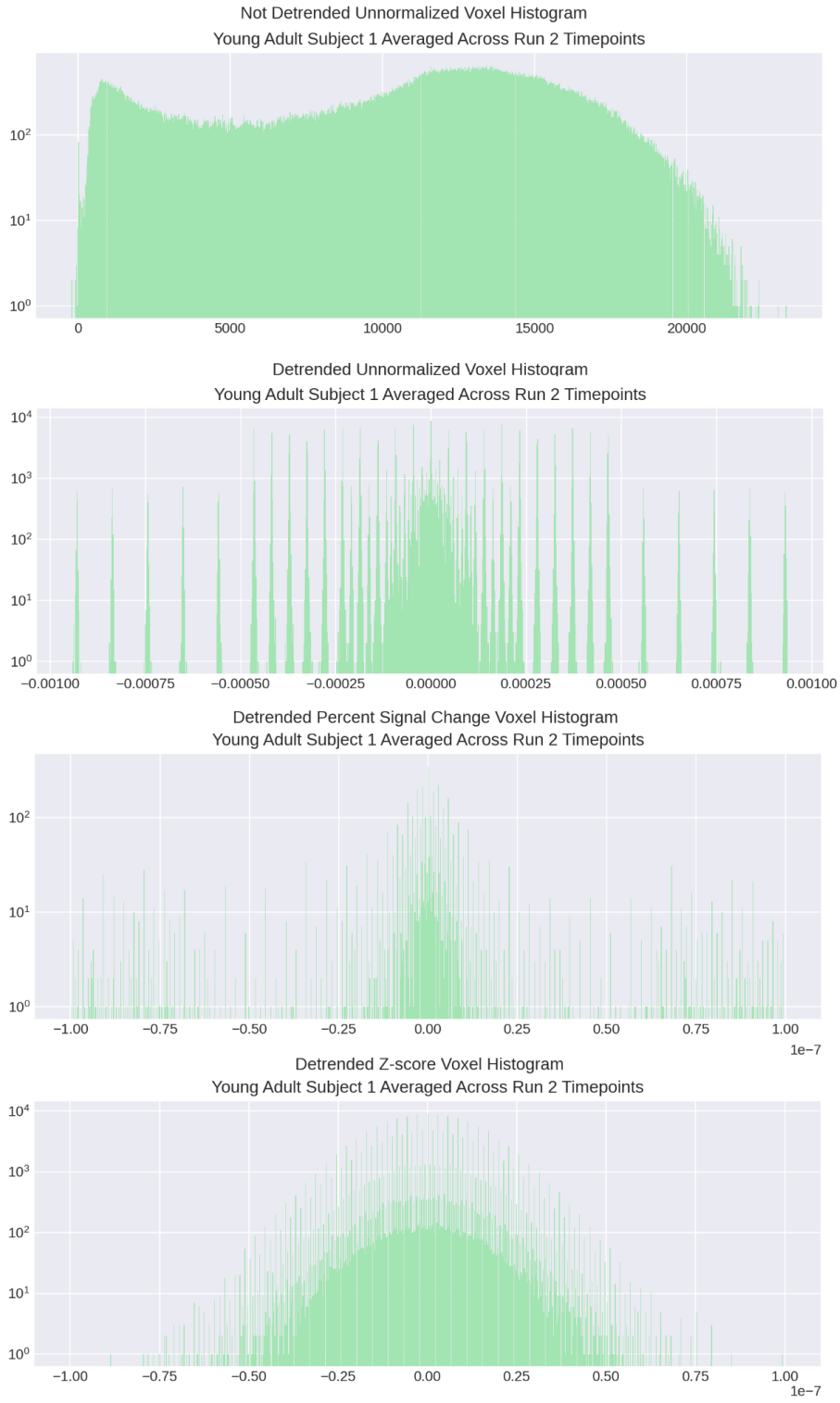

Not Detrended Unnormalized Voxel Histogram
Adolescent Subject 1 Averaged Across Run 2 Timepoints
10^3
10^2
10^1
10^0
0
5000
10000
15000
20000
25000


Detrended Unnormalized Voxel Histogram
Adolescent Subject 1 Averaged Across Run 2 Timepoints
10^4
10^3
10^2
10^1
10^0
−0.00100
−0.00075
−0.00050
−0.00025
0.00000
0.00025
0.00050
0.00075
0.00100


Detrended Percent Signal Change Voxel Histogram
Adolescent Subject 1 Averaged Across Run 2 Timepoints
10^3
10^2
10^1
10^0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1e−7

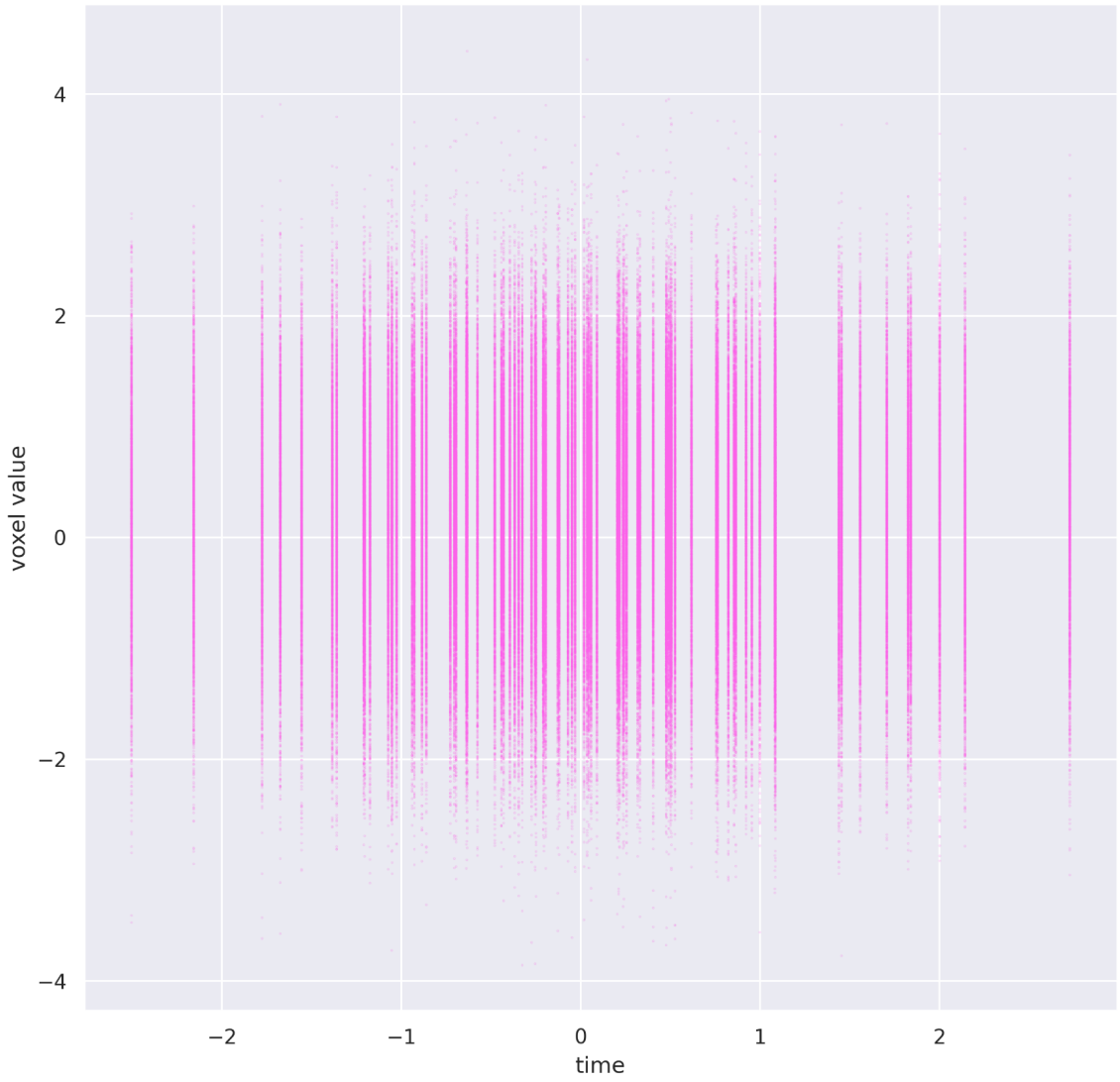

Detrended Z-score Voxel Histogram
Adolescent Subject 1 Averaged Across Run 2 Timepoints
10^4
10^3
10^2
10^1
10^0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1e−7

APPENDIX



APPENDIX : Voxel
Feature Space

Subject 1 on Run 1


4
2
0
−2
−4
voxel value
−2
−1
0
1
2
time