# Obtaining leaner deep neural networks for decoding brain functional connectome in a single shot

Sukrit Gupta[1], Yi Hao Chan[1], Jagath C. Rajapakse, and the Alzheimer's Disease Neuroimaging Initiative[2]

*School of Computer Science and Engineering, Nanyang Technological University, 639798, Singapore.*

**Abstract**

Neuroscientific knowledge points to the presence of redundancy in the correlations of brain's functional activity. These redundancies can be removed to mitigate the problem of overfitting when deep neural network (DNN) models are used to classify neuroimaging datasets. We propose an algorithm that removes insignificant nodes of DNNs in a layerwise manner and then adds a subset of correlated features in a single shot. When performing experiments with functional MRI datasets for classifying patients from healthy controls, we were able to obtain simpler and more generalizable DNNs. The obtained DNNs maintained a similar performance as the full network with only around 2% of the initial trainable parameters. Further, we used the trained network to identify salient brain regions and connections from functional connectome for multiple brain disorders. The identified biomarkers were found to closely correspond to previously known disease biomarkers. The proposed methods have cross-modal applications in obtaining leaner DNNs that seem to fit the data better. The corresponding code is available at `https://github.com/SCSE-Biomedical-Computing-Group/LEAN_CLIP`.

*Keywords:* Alzheimer's disease, attention deficit hyperactivity disorder, brain decoding, deep neural networks, feature selection, major depressive disorder, mild cognitive impairment

[1]These authors contributed equally.

[2]Data used in preparation of this article were obtained from the ADNI database (`adni.loni.usc.edu`). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete list of the ADNI investigators can be found at `http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf` Corresponding author: Jagath C. Rajapakse (ASJagath@ntu.edu.sg).

## 1. Introduction

Deep neural networks (DNN) have been successfully applied to a wide variety of classification problems, ranging from image recognition to language processing [21, 28]. In these applications, the choice of network architecture and number of trainable parameters affects network performance significantly. Larger networks learn more complex and accurate mappings, which comes with increased computational complexity and possibilities of overfitting [48]. On the other hand, having insufficient number of parameters limits the network's ability to learn the correct mapping [60]. Therefore, finding the optimal neural network architecture is an important problem. One possible solution involves performing elimination on a large neural network by removing redundant nodes and connections. This has been shown to produce comparable performance with a smaller network but with better generalization capability [1, 4].

Neural networks and machine learning techniques have also been successfully applied to neuroimaging data. Non-invasive neuroimaging techniques are used to characterize functional and structural anomalies in the brain and aid in better diagnosis and treatment [13]. Functional and/or structural connectivity derived from these techniques are used as features for the neural network models to classify diseased and normal subjects [7, 15]. However, such studies typically involve many more input features ($\sim 10,000$) than subject scans ($\sim 1,000$), making the neural networks prone to overfitting. Also, experimental noise introduces systematic connectivity [34], which should be ignored during the classification task. Most crucially, overfitting is exacerbated by the presence of redundancies: not all functional connectivity features are important for differentiating between scan samples from two subject groups.

In recent works [16], the authors used feature salience scores to remove less salient features. In order to find salient features, DeepLIFT [44] was used to compute salience scores of both input features and hidden layer nodes. DeepLIFT is one of the recent attempts (besides Integrated Gradients [50], Layerwise Relevance Propagation [5] and SHAP [30]) that circumvent the issues with gradient-based approaches (such as zero gradients or discontinuities). These approaches find the contribution of nodes at each layer by propagating the contributions from the output layer. A major differentiating factor of DeepLIFT is that it gives consideration to both negative and positive contributions and computes the relevance scores efficiently in a single pass. By recursively removing the least salient features, these recent works arrived at a much smaller model with comparable accuracy to the original model, reducing the problem of overfitting without compromising on classification accuracy.

However, one limitation of these previous approaches is efficiency: they involve multiple elimination and fine-tuning steps to arrive at a smaller neural network. In this paper, we propose an algorithm called Layerwise Elimination of Accessory Nodes (LEAN) that performs the elimination of accessory nodes in a single shot. Accessory nodes are defined as nodes which do not influence the classification task significantly - such nodes will be assigned a low importance score by a valid decoder. LEAN uses node salience scores derived from

a valid DNN decoder to eliminate accessory nodes in the network, producing a leaner, more computationally efficient, and more generalizable deep neural network that is derived in a single shot.

LEAN leads to a drastic reduction in parameters, including the removal of a large number of input features. While powerful and efficient, this also leads to the loss of whole groups of correlated features. In neuroimaging datasets, such correlations exist due to the *specialization* property of the brain, whereby distinct sub-systems in the brain perform specialized tasks [17]. To tackle this, we introduce another algorithm called Correlation-based eLimination of InPuts (CLIP) that identifies and retains a subset of correlated input features. When these features are combined with the remaining input features from LEAN, it leads to an improvement in the DNN performance as compared to LEAN. Combining LEAN and CLIP leads to a model that provides the best of both worlds: low number of parameters with minimal accuracy drop.

Besides the goal of getting an efficient classifier with the highest accuracy and generalization ability, a relevant and important research direction in neuroscience is to find the biological markers that are associated with a brain state (disease or cognitive task), which differentiate it from another brain state. We denote identification of brain regions and connection that are associated with a particular brain state as brain decoding. Traditional methods for brain decoding include multivariate pattern analysis [18], sparse networks based feature selection [38], latent Dirichlet allocation [40], and the use of nodal features [43]. Such methods are however based on simple or linear models. On the other hand, DNN-based approaches build deep and hierarchical models and represent key patterns underlying brain activation in its nodes and connection weights [38]. Deep learning techniques make no assumptions about application specific a priori knowledge and therefore give consistent and unbiased decoding based on neuroimaging data. Recent applications of neural networks on neuroimaging data have derived the importance of input features in the classification task to uncover biomarkers for neurological diseases [16, 19] or reveal task-related brain functional modulations [27]. Identifying disease-specific biomarkers aids in building models and classifying and predicting the progression or occurrence of unknown diseases in individuals. To do so, we use the salience scores given by DeepLIFT to determine salient brain connections and regions for Alzheimer's disease (AD), mild cognitive impairment (MCI), attention deficit hyperactivity disorder (ADHD), major depressive disorder (MDD), and Autism Spectrum Disorder (ASD) patients using their functional MRI scans.

In sum, we have made the following novel contributions in this work:

- We proposed an efficient layerwise node elimination approach, LEAN, that removes accessory input features and hidden layer nodes from a DNN in a single iteration. We further adapted it for brain decoding by proposing CLIP to add in a subset of correlated features, leading to a model with minimal number of connection while maintaining the accuracy.

- The proposed approach finds salient functional connections and identifies brain regions that are responsible of classification of brain disease.

- The proposed methods were applied on multiple brain disorders for classification of patients from healthy controls and identification of disease biomarkers.

## 2. Methods

### *2.1. Feedforward DNN*

Given a pair $(x, d)$ where $x = (x_i)$ denotes the input feature vector and $d \in D$ denotes the sample label, we used a feedforward neural network of $L$ layers with the first $L-1$ layers having rectified linear unit activation and a softmax layer at the end of the network. Let the weights and biases of the layer $l$ be given by $W_l$ and $b_l$, respectively. The output of layer $l \neq \{0, L\}$ is given by:

$$h_l = \text{ReLU}(W_l^T h_{l-1} + b_l) \tag{1}$$

For the input layer, $h_0 = x$. For the softmax layer, the output probabilities $y$ of the input $x$ belonging to class $k$ is given by:

$$P(y = k|x) = \text{softmax}(W_L^T h_{L-1} + b_L) \tag{2}$$

where $k \in \{1, ..., K\}$ represents the class label and the output layer weight $W_L = [w_{k,L}]$ and bias $b_L = (b_{k,L})$. To learn the parameters of the network, the cross-entropy cost is defined as:

$$J(\theta) = -E_x[\log\ P(y = d|x, \theta)] \tag{3}$$

where $E_x$ is the expectation taken over all scan samples $x$ and $\theta = \{(W_l, b_l)\}_{l=1}^{L}$ denotes all the parameters in the network.

### *2.2. Saliency of nodes in network layers*

Let $f$ be the neural network function mapping input $x$ to output $y$. A simpler *explanation model* $g$ is found such that $g$ is both interpretable and an approximation of the model $f$. Let the number of nodes in layer $l$ be $n_l$. Using an appropriate reference, let us assign to each neuron $i$ in layer $l$ a contribution $C_{\Delta h_{i,l} \Delta h_{k,l+1}}$ to the change in the output of neuron $k$ in layer $l+1$:

$$\sum_{i \leq n_l} C_{\Delta h_{i,l} \Delta h_{k,l+1}} = \Delta h_{k,l+1} \tag{4}$$

when $l \in \{0, \ldots, L-2\}$ and

$$\sum_{i \leq n_{L-1}} C_{\Delta h_{i,L-1} \Delta y} = \Delta y \tag{5}$$

when $l = L-1$, where $\Delta h_{i,l}$ is the change in the activation of the $i^{th}$ neuron of layer $l$ due to the input relative to the reference. $C_{\Delta h_{i,l} \Delta h_{k,l+1}}$ is computed from the Linear, Rescale and RevealCancel rule from [44].

Given the reference input $\bar{x}$ and the original input $x$, we substitute $\Delta y = f(x) - f(\bar{x})$ and $g(x) = f(x)$, giving us an equation for the model $g$:

$$g(x) = f(\bar{x}) + \sum_{i,l} C_{\Delta h_{i,l} \Delta y} \tag{6}$$

where the contribution of nodes in each layer $l$ to the output $y$ is given by $C_{\Delta h_{i,l} \Delta y}$ [44]:

$$C_{\Delta h_{i,l} \Delta y} = \Delta h_{i,l} \sum_{k,l} \frac{C_{\Delta h_{i,l} \Delta h_{k,l+1}}}{\Delta h_{i,l}} \frac{C_{\Delta h_{k,l+1} \Delta y}}{\Delta h_{k,l+1}} \tag{7}$$

Contribution of nodes in all layers $0 \leq l < L$ to the output $y$ is obtained by backpropagating contributions of layers to the output. For nodes in each layer $l \in \{0, \ldots, L-1\}$, we compute the *salience score vector* $c_l$ given by:

$$c_l = \left( C_{\Delta h_{i,l} \Delta y} \right)_{i < n_l} \tag{8}$$

where $C_{\Delta h_{i,l} \Delta y}$ is derived from Equation 7. The DeepLIFT method [44] computes the layer's salience scores based on change in the output from a reference, allowing information to propagate across the network layers even when the gradient is zero. We compute the average contributions for all layers over multiple test samples to get the final salience score for each feature.

### *2.3. LEAN: Layerwise elimination of accessory nodes*

A decoder can identify a subset of distinguishing input features and weights that are important to classify samples. We proposed a brain decoding strategy in [16], where we obtained the salience scores for input features and hidden layer nodes. In the proposed scheme, a fraction $\mu$ of nodes with the lowest salience scores $c_k$ from layers $k \in \{0, \ldots, L-1\}$ are removed iteratively and the pruned model is fine-tuned at each step. Although the proposed strategy improved the classification performance, the process involved multiple iterations of elimination and fine-tuning that is time-consuming and cumbersome. Thus, we propose an efficient strategy to find a layerwise salience score threshold, below which nodes from the layers (input and hidden) of the DNN are removed.

For the salience score vector $c_l$ in each layer, we determined the best fit distribution by computing the log likelihood of the model fit. We did not make any assumption about the best fit distribution and used multiple distributions (viz; power-law, log-normal, exponential and stretched exponential) to find the best fit for the salience score distribution at each layer. The power-law distribution is defined by the distribution of the salience scores $c_{il}$ for nodes $i < n_l$ in layer $l$ [2, 10]:

$$p_{\text{power}}(c_{il}) = k c_{il}^{-\alpha}$$

where $\alpha$ is the scaling parameter, $k = (\alpha - 1)\, c_{\min}^{\alpha - 1}$, and $c_{\min}$ is the minimum degree that obeys the power-law. The distribution parameters were computed using maximum likelihood estimation [10]. We compute statistical significance

for the salience scores of nodes in their respective layers and remove the nodes with $p - value \geq 0.95$. The complete overview of the layerwise elimination of nodes is given in Algorithm 1.

**Algorithm 1** LEAN: Layerwise elimination of accessory nodes

Input: DNN with layers $\{l_i\}_{i=0}^{L}$
Output: Reduced DNN with layers $\{l'_i\}_{i=0}^{L}$

Train DNN with layers $\{l_i\}_{i=0}^{L}$
**for** each layer $l_i$ in $\{l_i\}_{i=0}^{L-1}$ **do**
  $c_i \leftarrow C_{\Delta h_{*,l_i} \Delta y}$
  $l'_i \leftarrow \{c_{ik} \mid p - value(c_{ik}) \geq 0.95\}$

### *2.4. CLIP: Correlation-based elimination of inputs*

Besides the presence of a large number of correlated features, brain functional connectomes are also known to be modular in nature [17, 47] and such modularity or clusters give rise to correlated features. Different modules correspond to different sub-systems in the brain, performing a specific function. Examples of these modules are shown in Figure 1(a). Modules or clusters have stronger connectivity between nodes within the module and weaker connectivity with nodes outside the module. This gives rise to the intuition that the functional connectivity for some regions of interest (ROI) - especially ROIs within the same module - are correlated with each other.

Although some modules involved in cognition and decision-making vary across subjects, others involved in functions related to perception and motor control are stable across subjects [17, 32]. This gives rise to inter-subject relationships among functional connectivity features. While the extent of how high correlations is difficult to determine, the correlation between pairs of inter-cluster connections would likely be lower than the correlation between pairs of intra-cluster connections. This is because of the sparser and variable inter-cluster connections, and the relatively denser and more stable intra-cluster connections due to similar modularizations as shown in Figure 1(b) and (c). For example, the dense connections within the visual module [17, 32], which has a low inter-subject variability, should be highly correlated across subjects.

#### *2.4.1. Finding clusters of correlated features*

While there are other dimensionality reduction approaches such as Principal Component Analysis or Independent Component Analysis that reduce the problem of multicollinearity [22], such approaches require either prior knowledge of the number of components to be used, or extensive experimentation to arrive at the optimal number. Alternatively, a clustering-based approach presented below provides a more principled solution. Figure 2 provides a schematic of the

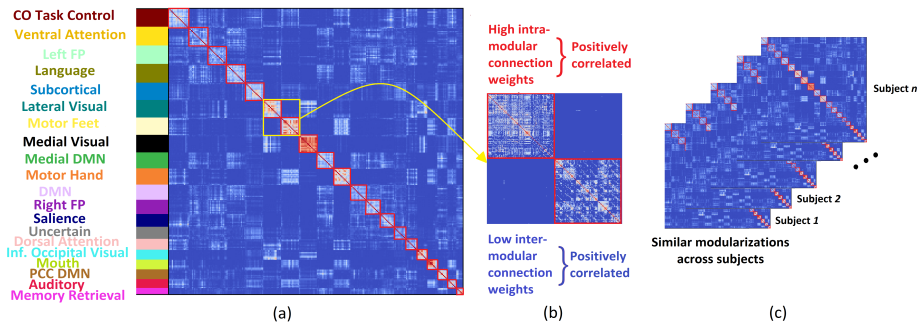



Figure 1: Visualization of the functional connectivity matrices. (a) The matrix has its rows rearranged such that nodes in the same functional clusters are placed consecutively. The labels of different functional modules are given by the left color bar. (b) A magnified view of a subset of the correlation matrix is provided. Cells colored white have a higher weight than the cells colored blue. The white patches along the left diagonal represent intra-cluster connections, while the blue patches along the right diagonal represents inter-cluster connections. Intra-cluster connections have high and inter-cluster connections have low values across subjects and are therefore positively correlated. (c) A schematic of how some of the clusters are similar across subjects.

intuition behind finding such clusters that are spread across the distribution of salience scores.

We formulate this problem in terms of finding clusters/communities of correlated features, where we compute the similarity matrix, $S = \{s_{ij}\}$, where $s_{ij}$ denotes the correlation between features $i$ and $j$ from the training data:

$$s_{ij} = \frac{\mathrm{Cov}(x_i, x_j)}{\sigma(x_i)\sigma(x_j)} \tag{9}$$

where Cov finds the covariance between two variables and $\sigma$ finds the standard deviation, and $x_k$ denotes the vector containing values of feature $k$ across the samples.

We perform clustering of the similarity matrix $S$ to obtain a cluster vector $m = (m_k)$ where each element $m_k$ corresponds to the cluster label assigned to each voxel $k$ from a set $\{1, 2, \ldots, M\}$ and $M$ of clusters. This is done by the minimization of the normalized cut cost given by:

$$\text{cut-cost}(S, M) = \frac{1}{M} \sum_{l=1}^{M} \left(1 - \frac{u_l^T S u_l}{u_l^T D^k u_l}\right) \tag{10}$$

where $D$ denotes the diagonal degree matrix of $S$ and $U = \{u_l\}_{l=1}^{M}$ is a set of binary matrices representing $m$, such that $l$th module is given by $u_l = (u_{li})_{i=0}^{|x|}$ where $u_{li} = 1(m_i = l)$ and $1(\cdot)$ denotes the identity function.

The minimization of the cut-cost in (10) is performed by multi-class spectral clustering [49]. The number of clusters is found by computing the elbow point [42] (i.e. the point of maximum curvature) of the scree plot of the eigenvalues of similarity matrix $S$. However, since the number of features $|x|$ are often

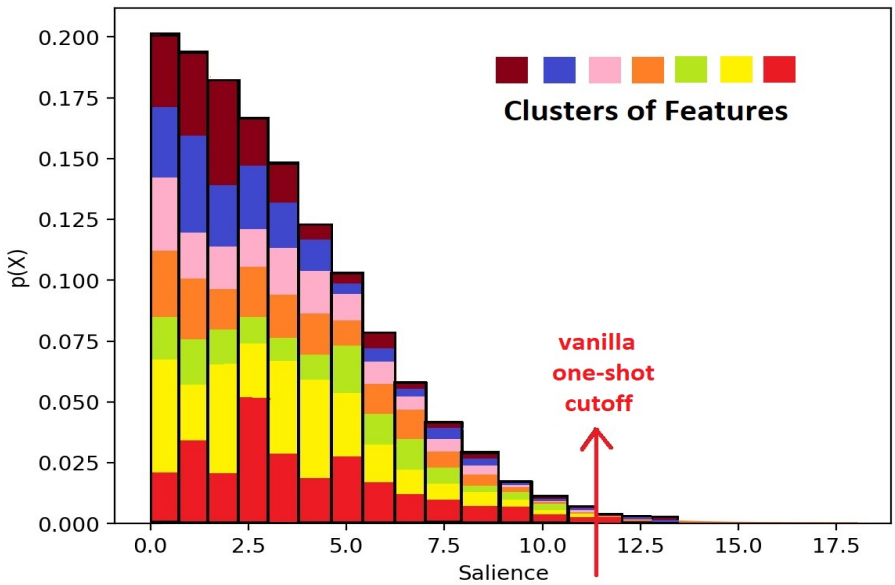



Figure 2: Schematic of how the features are distributed in different clusters throughout the distribution of salience scores and how salience scores consider only features from a few clusters.

large, the similarity matrix has a large size and computing its eigendecomposition becomes expensive. Therefore, we sparsified the matrix $S$ by thresholding correlations with values less than 0.3 (which is a recognised threshold for low correlation values [20]). Thereafter, we used the Implicitly Restarted Arnoldi Method [29] to perform eigendecomposition on the sparse matrix to obtain the eigenvalues.

### *2.4.2. Selecting subsets of correlated features*

For each of the computed clusters, a subset of features that have the highest correlation with features from the same cluster was selected. We do this by using the *intra-cluster degree*, $\alpha_i$, of feature $i$ given by

$$\alpha_i = \sum_{m_j = m_i} s_{ij} \tag{11}$$

We select a fraction of nodes with the highest $\alpha_i$ values from each cluster and include them as features in addition to the ones selected from one-shot elimination pruning. The whole process for removal of subsets of correlated features is summarized in Figure 3 and Algorithm 2.

## *2.5. Combining LEAN and CLIP*

We argue that the presence of correlated features leads to a sudden drop in accuracy with LEAN because while performing recursive elimination, we retain

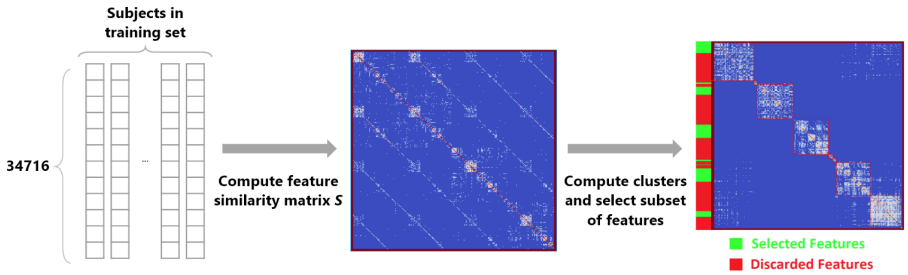



Figure 3: Overview of the feature selection process. To generate $S$, the vector of features representing each subject is concatenated and the similarity matrix is computed. Thereafter, clustering is performed on $S$ and a subset of nodes with the highest intra-cluster degree each cluster are retained, while the rest are discarded. The matrix on the right has been rearranged such that nodes in the same cluster are placed consecutively.

**Algorithm 2** CLIP: Correlation-based elimination of inputs

Input: Concatenated features from training data: $\{x\}$, k
Output: Reduced feature set $l'_0$

$S \leftarrow \frac{\text{Cov}(x_i, x_j)}{\sigma(x_i)\sigma(x_j)}$
$M \leftarrow elbow(S)$
$m \leftarrow$ minimize cut-cost$(S, M)$
$l'_0 \leftarrow l'_0 \cup$ top k% features from each module in $m$

a fraction of features from clusters of correlated features at different thresholds. However, with LEAN, entire clusters of correlated features are lost at once. This problem has been studied previously for various linear and non-linear classifiers [62] by using different feature selection methods [54]. For instance, in ordinary least square regression, the existence of multicollinearity leads to a larger standard error [14] and affects the interpretation of feature salience - a feature that would have been deemed as important is no longer significant when another correlated feature is present. We hypothesize that LEAN discards too many input features at once and using the additional subsets of correlated features (identified by Algorithm 2) will improve the model accuracy. Thus, while hidden layers nodes are eliminated only using LEAN (Algorithm 1), the input features are first eliminated using LEAN and then some are added back from the subsets of correlated features via CLIP (Algorithm 2).

### *2.6. Decoding the brain functional connectome associated with brain disease*

Brain decoding is defined as the identification of brain activity patterns that emerge from activations as well as interactions among specific brain regions and connections, which distinguish one brain state from another. Decoding the DNN trained on connectome features translates to identifying salient features of the

DNN, which correspond to biomarkers (i.e. key brain connections and ROI) associated with the brain state. We demonstrated this approach in [16] for the first time by using feedforward DNN. We successfully adopted DeepLIFT in decoding brain functional connectivity in [16], which efficiently computes salience scores for input features in a single pass and then recursively eliminate irrelevant. Such an approach only focuses on input features and does not optimize the DNN architecture. In this paper, we improve upon our previous work by proposing a combination of LEAN and CLIP for not only to decode the input feature but also finds leaner DNN model for efficient classification without the loss of accuracy. By using LEAN and CLIP on resting-state fMRI brain scans gathered in brain disease, we achieve state-of-art accuracies for disease classification with leaner DNN models and salient input features. The decoded input features correspond to brain connections that are associated with brain disease.

## 3. Results

### *3.1. Datasets*

We downloaded resting state functional MRI scans for AD and MCI from the Alzheimer's Disease Neuroimaging Initiative (ADNI); for ADHD from the International Neuroimaging Datasharing Initiative (INDI) [6]; for MDD from the data provided by the Creativity and Affective Neuroscience Lab in the Brain Imaging Center of Southwest University and for ASD from the Autism Brain Imaging Data Exchange (ABIDE). The details of the acquisition protocols, subjects, preprocessing pipelines are attached in the supplementary materials.

### *3.2. Features for classification*

We used the Power atlas [35] to obtain functionally diverse 264 ROIs spanning the entire cerebral cortex. Average time series were computed for voxels within a spherical radius of 2.5mm surrounding each ROI, and the functional connectivity matrix for each scan was generated by computing the Pearson correlation coefficient (without thresholding) for each pair of ROI. Since the matrices are symmetrical, we consider only the upper triangular connectivity matrix and flatten it into an input vector for the network. This resulted in an input vector with 34716 elements.

### *3.3. Classification for full feature set*

Both the encoder and decoder were implemented in Python using the Keras, Tensorflow and DeepLIFT libraries. For datasets with more than one scan per subject, we ensured that all the subject scans were either in the training or test set. For all datasets, a batch size of 8 was used along with a learning rate of $10^{-4}$.

Besides the feedforward neural networks (FFN), different convolution neural network (CNN) architectures [8, 31] and support vector machines (SVM) were implemented with different parameters. For CNNs, we gave the connectivity matrix as an input, and varied the number of filters and the number of

layers. The number of filters in each layer was varied based on the number of weights in the corresponding FFN such that the number of trainable parameters were same in both the architectures. The number of trainable weights were changed from $1.7 \times 10^4$ to $3.5 \times 10^7$. For SVMs, we varied the parameters ($C \in \{0.001, 0.01, 0.1, 1, 10\}$, $\gamma \in \{0.001, 0.01, 0.1, 1\}$) and tried different kernels such as linear, polynomial, RBF and sigmoid. For the FFN, the number of hidden layers and the number of neurons in each hidden layer were varied. The final architecture for the full feature set was obtained using grid search from accuracies obtained from stratified 5-fold cross validation. We added dropout of 0.1 to the hidden layers and imposed early stopping to prevent overfitting. The details of the different configurations for the DNNs can be found in the supplementary materials. We observed that the highest accuracies were still given by FFN (except in case of CN vs AD, where the difference was insignificant). The average accuracies along with the standard deviation for the SVM, CNN and FFN models are reported in table 1 and the corresponding architectures are reported in table S5 in the supplementary materials. These accuracies are obtained from repeating the experiments using 10 different seeds. For each seed, 5-fold cross validation was performed.

Table 1: Performances of the best models of different models on the neuroimaging datasets.

| Disorder | SVM | CNN | FFN |
|---|---|---|---|
| ASD | 68.0% ± 3.4% | 70.6% ± 2.9% | 70.9% ± 4.2% |
| AD | 71.3% ± 6.5% | 78.1% ± 5.3% | 77.4% ± 4.6% |
| ADHD | 54.8% ± 5.5% | 60.5% ± 4.5% | 61.1% ± 4.2% |
| MCI | 64.6% ± 4.3% | 66.5% ± 3.8% | 67.5% ± 4.1% |
| MDD | 73.3% ± 3.9% | 77.1% ± 3.6% | 78.3% ± 3.2% |

### *3.4. Classification with feature subsets from CLIP and LEAN*

We tried 3 different elimination approaches: 'LEAN', 'CLIP + LEAN (Inputs Only)' and 'LEAN + CLIP'. 'LEAN' (described in Algorithm 1) involves keeping only statistically significant features at each layer; 'CLIP + LEAN (Inputs Only)' includes subsets of correlated features (Algorithm 2) in addition to DeepLIFT features, but no elimination is performed for the hidden layers; and 'CLIP + LEAN' goes even further to prune the hidden layers, as described in Section 2.5. For CLIP, we generated the similarity matrix (described in Section 2.4.1) to derive a subset of correlated features that are combined with features obtained from LEAN. We did this for each of the seeds and folds and retained 5% of the top features for each cluster. Table 2 summarises the changes in classification accuracy and remaining number of trainable parameters (compared to the the original model) for different approaches on the 3 datasets.

As seen in table 2, we observe that relative to the full feature set, LEAN is able to reduce the model parameters drastically to around 0.5% to 2.1% of the original number of parameters. Intuitively, keeping the most important features

should lead to a model with minimal drop in accuracy. However, we found that the drop in accuracy is large (1.6% in ADHD to 3.1% in MDD) and thus LEAN cannot be directly used. One can argue that a different p-value threshold can be used to allow more features to be included, but it will take several iterations to arrive at the threshold which gives the optimal model. Furthermore, such a tuned threshold is unlikely to generalize to other situations.

We reduced the drop in accuracy by identifying subsets of correlated features in 'CLIP + LEAN (Inputs Only)'. Evidently, our approach leads to a smaller drop in accuracy as compared to the LEAN approach (and even lead to an increase in accuracy for CN/AD). Although we obtain similar classification performance when we only prune the inputs (and not the hidden layers), the number of remaining parameters (7% to 11% of the original) are much larger than LEAN. By extending the node elimination process to the hidden layers in 'CLIP + LEAN', we further reduced the number of parameters involved to levels quite close to the LEAN approach, with similar accuracies (and with consistently higher accuracies than LEAN).

Comparing the results of 'CLIP + LEAN (Inputs Only)' and 'CLIP + LEAN', it is seen that there is no significant trend in terms of change in accuracy. This shows that we can safely remove hidden nodes to obtain a leaner model. Thus, our subsequent analysis will be focused on LEAN and 'CLIP + LEAN'.

Table 2: Comparison of functional connectivity based classifiers. Params represent the number of parameters left in the reduced model, relative to the model that uses the full feature set.

| | All Features | LEAN | | CLIP + LEAN (Inputs Only) | | CLIP + LEAN | |
|---|---|---|---|---|---|---|---|
| Disorder | Accuracy | Acc | Params | Acc | Params | Acc | Params |
| ASD | 70.9% | 68.1% | 0.5% | 69.3% | 9.5% | 68.9% | 1.1% |
| AD | 77.4% | 75.8% | 0.7% | 78.1% | 7.7% | **78.1%** | 2.0% |
| ADHD | 61.1% | 59.0% | 2.1% | 59.3% | 9.9% | **59.4**% | 5.7% |
| MCI | 67.5% | 64.5% | 0.6% | 65.3% | 6.9% | **66.2%** | 2.0% |
| MDD | 78.3% | 75.2% | 1.2% | **76.4%** | 10.9% | 76.0% | 2.6% |

*3.4.1. Reduction in overfitting*

Overfitting happens during model training when the test loss increases after reaching a minimum [60]. If a model overfits, it fails to generalise and remembers only the training data. To evaluate overfitting, we use the difference between the test set loss and train set loss at the epoch as evaluated by cross-entropy when the best model was found. Because of the small size of the datasets, the difference of train and test losses renders a stable measure of overfitting [33].

The model was first trained with early stopping, such that the best model was chosen by looking at the test accuracies of each of the epochs. The best model is chosen at the epoch where test accuracy is highest. Table 3 shows the

Table 3: The differences of test and train losses computed using cross-entropy.

| Disorder | All Features | LEAN | CLIP + LEAN |
|---|---|---|---|
| ASD | 0.50 | 0.28 | 0.20 |
| AD | 0.34 | 0.14 | 0.13 |
| ADHD | 0.29 | 0.06 | 0.01 |
| MCI | 0.13 | 0.002 | 0.02 |
| MDD | 0.61 | 0.31 | 0.36 |

extent of overfitting for each of the datasets with cross-entropy being the metric used to compute the loss. The reported scores were calculated by averaging over 5 folds. As shown in the table, there is a clear reduction of overfitting across all datasets for both LEAN and 'CLIP + LEAN', after these algorithms were used to generate the leaner model.

### *3.4.2. Comparison with other redundancy removal methods*

Table 4: Accuracies of models produced by keeping X% most important features from SVM and logistic regression models. C + L = CLIP + LEAN.

| Disorder | FFN | C + L | | 1% | 5% | 10% | 100% |
|---|---|---|---|---|---|---|---|
| ASD | 70.9% | 68.9% | SVM | 53.0% | 59.0% | 67.7% | 68.0% |
| | | | Logreg | 63.1% | 67.0% | 66.6% | 68.1% |
| AD | 77.4% | 78.1% | SVM | 66.8% | 70.2% | 71.6% | 71.3% |
| | | | Logreg | 70.9% | 72.7% | 72.2% | 71.3% |
| ADHD | 61.1% | 59.4% | SVM | 53.2% | 51.6% | 52.0% | 53.1% |
| | | | Logreg | 52.5% | 51.9% | 52.1% | 53.2% |
| MCI | 67.5% | 66.2% | SVM | 64.6% | 62.9% | 60.1% | 58.7% |
| | | | Logreg | 56.1% | 57.9% | 58.4% | 58.7% |
| MDD | 78.3% | 76.0% | SVM | 69.4% | 72.5% | 72.9% | 73.3% |
| | | | Logreg | 70.5% | 72.6% | 72.9% | 73.3% |

We compared performances of 'CLIP + LEAN' with those of SVM and logistic regression, including their own feature importance methods. Experiments were conducted, keeping 1%, 5% and 10% of features as this is the same range of remaining features obtained from our proposed algorithms. For SVM, only the linear kernel is able to provide importance scores and for this comparison, the best model with a linear kernel was chosen for each dataset. As seen in table 4, as the number of features remaining increases, model accuracy for logistic regression generally increases but such a trend is less pronounced for SVM. Ultimately, 'CLIP + LEAN' still generally outperforms both the SVM and logistic regression models.

Additionally, since traditional models like SVM and logistic regression assume that the used features are uncorrelated, we used CLIP to retain a subset of uncorrelated features and compared the performance of the resulting models. As seen in table 5, the classification accuracy usually suffers when a feature subset is used. However, the accuracy is significantly lower ($p - value < 10^{-3}$) than the corresponding FFN in all the cases (except for MCI).

Table 5: Accuracies of baseline models before and after CLIP was applied.

| | SVM | | Logistic Regression | |
|---|---|---|---|---|
| Disorder | Before | After | Before | After |
| ASD | 68.0% ± 3.3% | 62.0% ± 2.9% | 67.9% ± 3.1% | 61.9% ± 3.1% |
| AD | 71.3% ± 6.5% | 67.6% ± 6.4% | 71.2% ± 6.6% | 70.7% ± 7.0% |
| ADHD | 54.8% ± 5.5% | 55.9% ± 4.8% | 53.2% ± 5.7% | 51.6% ± 5.4% |
| MCI | 64.5% ± 4.3% | 64.6% ± 4.3% | 58.7% ± 5.4% | 56.3% ± 5.5% |
| MDD | 73.3% ± 3.9% | 71.9% ± 5.0% | 73.3% ± 4.3% | 68.9% ± 3.7% |

### *3.4.3. Comparison of algorithm runtime*

Table 6: Average time taken per epoch (in seconds) for each variant of the proposed algorithms. mil = million.

| Disorder | Dataset Size | Initial Params | All Features | LEAN | CLIP + LEAN |
|---|---|---|---|---|---|
| ASD | 823 | 1.74 mil | 0.71 | 0.53 | 0.52 |
| AD | 299 | 1.74 mil | 0.36 | 0.24 | 0.24 |
| ADHD | 396 | 1.74 mil | 0.47 | 0.34 | 0.37 |
| MCI | 554 | 0.69 mil | 0.53 | 0.42 | 0.41 |
| MDD | 457 | 34.78 mil | 2.09 | 0.42 | 0.54 |

In order to evaluate time efficiency of different algorithms, average time taken per epoch to train the neural network were computed. As shown in table 6, all variants of our proposed redundancy reduction algorithms lead to a reduction of time taken - the increase being greater especially if the original 'All Features' model used was large like in the case of MDD. Here, we also specified the time taken for the one time operation to compute the feature clusters in CLIP. On average (over datasets and seeds), pre-processing took 135 minutes where the eigenvalue computation of the similarity matrix $S$ is the source of the bottleneck. This can be significantly reduced by lowering the number of eigenvalues to be computed - in our experiments, we defined it to be 1000. Also, importance score computation took, on average, a range of approximately 20s for AD to approximately 2 minutes for MDD (which starts off from a large model). All experiments were performed using 4 x NVIDIA Tesla P100 16GB on a Linux server with 72 cores. Comparing the proposed algorithms with each other, there

are slight variations in the timings but the differences are largely insignificant (except for the case of MDD) and both variants take approximately the same amount of time to run. We also concluded that the pruned models, LEAN and 'CLIP + LEAN', take a significantly shorter amount of time to run as compared to the full model ($p - value < 10^{-4}$). When a relatively small model is used, both variants take approximately the same amount of time to run.

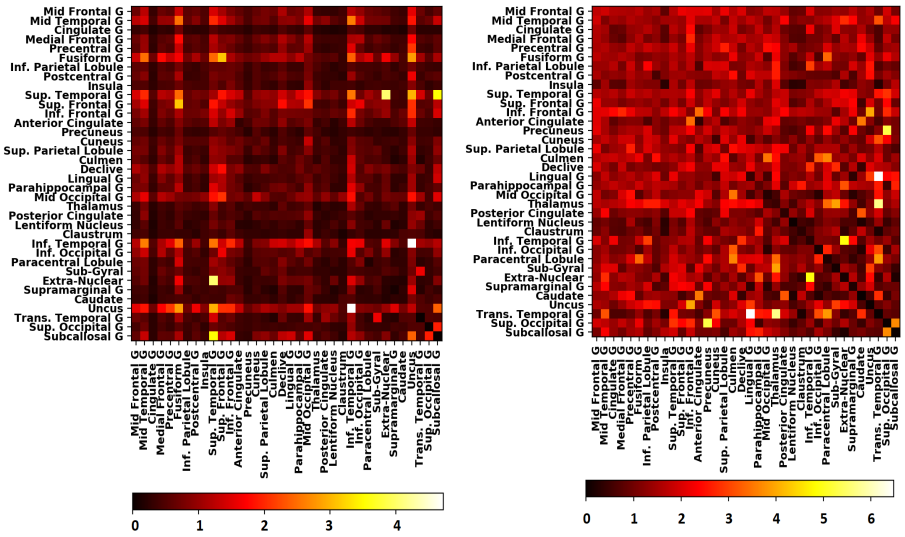
### *3.5. Salient brain features*

Thus far, we have seen how salience scores help to reduce the number of parameters involved, leading to a leaner model with minimal reduction in accuracy (or at times, even leading to an increase in model accuracy). We also use the salience scores to identify important features and ROI for the different diseases. The input fed into our neural network model is a 34716-dimensional vector and each element in the vector represents a correlation score between a pair of ROI (i.e. strength of a functional connection). We analyse the importance of the functional connections by using the salience scores and the importance of ROI by computing the sum of salience scores of the functional connections incident on the ROI. Since the Power atlas [35] does not provide anatomical labels, the Crossley atlas [12] (which has anatomical labels) was used to map ROI from the Power atlas on the basis of Euclidean distance. Then, ROI with the top 10% highest scores are visualised using Nilearn.

From Figure 4b, we found that the salient connections for MDD lie between the transverse temporal gyrus and lingual gyrus; the transverse temporal gyrus and thalamus; superior occipital gyrus and precuneus, and between extra-nuclear and inferior temporal gyrus. For AD, we found that the connections between the superior temporal and subcallosal; superior temporal and extra nuclear, and between the uncus and inferior temporal regions were salient (Figure 4a). For MCI, however, we found that only the connections between the uncus, subcallosal and uncus inferior temporal were salient (supplementary figure S2(b)). For ADHD, the salient connections were found between the paracentral and superior temporal; paracentral and inferior frontal; inferior temporal and superior occipital; supramarginal and sub-gyral; and supramarginal and extra nuclear regions (supplementary figure S2(a)). For ASD, salient connections were found between regions in the frontal inferior orbital to the putamen, frontal superior orbital, and frontal inferior operculum; the putamen and temporal pole superior; and cerebellum and the vermis (supplementary figure S2(c)).

For all the diseases, ROI in the inferior temporal gyrus and paracentral lobule were common among the salient regions. The other most distinctive ROI for MDD were located in the subcallosal gyrus, supplementary motor area, the medial frontal gyrus, the parahippocampal gyrus and the posterior cingulate (Figure 5b); for ADHD in the postcentral gyrus, inferior frontal gyrus, transverse temporal gyrus, superior occipital gyrus, the insula and the medial frontal gyrus (supplementary figure S3(a)). For both AD and MCI, the salient regions were found to be in the cerebellum, temporal pole middle, frontal superior medial, parahippocampal and fusiform gyrus (Figure 5a and supplementary figure

S3(b), respectively). For ASD, salient regions were found in the inferior temporal lobe, the rectus, heschl's gyrus, cerebellum, and the paracentral lobule (supplementary figure S3(c)).




(a) CN vs AD (b) CN vs MDD

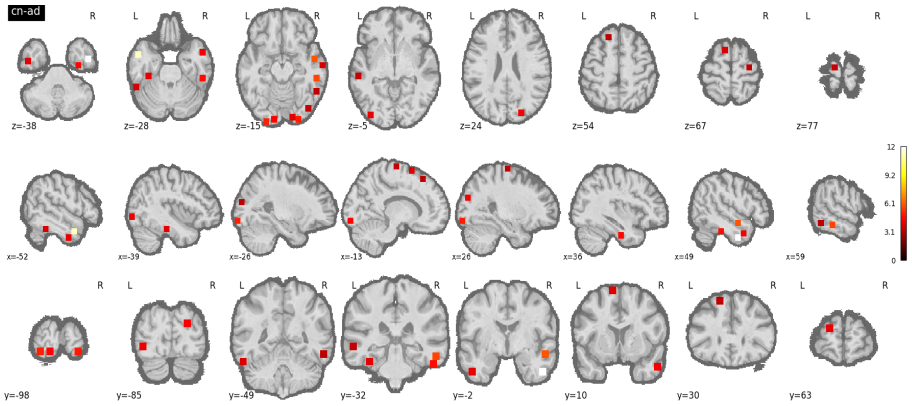
Figure 4: The salience scores of functional connections between brain regions derived while classifying normal and diseased participants for (a) Alzheimer's Disease; and (b) Major Depressive Disorder. Mid: Middle, Inf: Inferior, Sup: Superior, G: Gyrus

## 4. Discussion

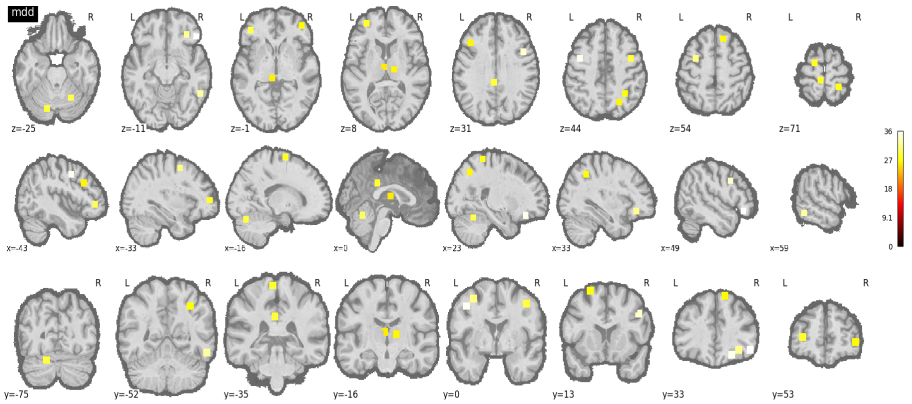
### *4.1. DNN models for full feature set*

As seen in table 1, the feedforward DNN using the entire feature set is generally able to achieve better classification accuracies than CNNs and SVMs. The rather high standard deviation is attributable to the size of the datasets [57], which typically do not contain more than a few hundred scans. The low accuracy for ADHD is attributable to the mismatch between the age of the subjects and the age of the subjects used to derive the Power atlas [35]. Also, more subtle connectivity differences with respect to CN subjects makes it harder to classify MCI as compared to AD.

We found that CNNs and DNNs perform consistently better than SVMs. However, the difference in performance between CNNs and feedforward DNNs are subtle: except for CN-AD, feedforward DNNs did better than CNNs but the difference is rarely more than 1%. From these results, CNNs - even the customised ones - do not seem to be getting any additional significant information for functional connectivity. Although CNNs have desirable features such




(a) CN vs AD




(b) CN vs MDD

Figure 5: The axial, sagittal and coronal views of top 10% salient ROI differentiating normal and diseased patients for (a) Alzheimer's Disease, and (b) Major Depressive Disorder.

as parameter sharing, they only capture information within their local receptive field (usually a small square-shaped subset, or a row or column along a matrix). Such information is not sufficient to capture the functional connectivity present in brains as these receptive fields do not consider global connectivity patterns. On the other hand, the fact that CNNs have fewer parameters than feedforward DNNs shows the possibility of redundancies in feedforward DNNs which makes it important to eliminate accessory nodes from feedforward DNNs. With LEAN, we are able to get a leaner model while methodically removing less salient features and nodes (instead of being constricted by the spatial limitations of local receptive fields).

*4.2. LEAN + CLIP: high accuracy with few parameters*

From the results in table 2, we observe that LEAN has the largest accuracy drop but the smallest number of remaining parameters; but, 'CLIP + LEAN (Inputs Only)' results in the small drop in accuracy but retains the largest number of parameters. However, 'CLIP + LEAN' gives the best of both worlds - the number of weights retained is close to levels from LEAN, but the accuracy drop is similar to that of 'CLIP + LEAN (Inputs Only)'. An implication of our results is the presence of correlated features in functional connectomes, which can be exploited to reduce input feature set in functional/structural connectomes. In our case, we observed that LEAN led to drastic reduction in the input feature set, thereby removing sets of correlated features entirely. Our results show that adding a subset of the correlated features in 'CLIP + LEAN' (and also in 'CLIP + LEAN (Inputs Only) led to an improvement in the performance of the classifier.

Crucially, our results show that there is a large amount of redundancy in neural network models. Our experiments with quantification of overfitting and time taken per epoch for the proposed model show that the proposed model not only leads to a reduction in overfitting but is also faster. The latter was expected since there is a significant reduction in the number of trainable network parameters. Feedforward networks have been used by researchers in the past [19, 25, 27] to perform classification on functional connectivity data. However, these works often use the full set of features and overfitting is a key limitation in such approaches. In our work, we have proposed a way to reduce the effects of overfitting and showed how even as it leads to a drastic reduction of parameters, the drop in accuracy is minimal. Thus, just a simple feedforward DNN is sufficient to capture crucial patterns in the data to classify healthy and diseased brains. Adding more nodes in the hidden layers does not improve model accuracy much (and in some cases such as CN-AD classification, doing so even leads to poorer generalization).

Although we have only tested our approaches on neuroimaging datasets, they are also widely applicable to other datasets. LEAN (Algorithm 1) is applicable in settings where the number of features greatly outnumber the size of the dataset, or in cases where the dataset is small and will benefit from using fewer features so as to increase generalizability of the model. CLIP (Algorithm 2) is applicable to any datasets where there are features that have strong correlations with each other.

*4.3. Identified salient features are clinically relevant*

For our algorithm to perform well, it is important that the features classified as salient by our decoder have clinical relevance to the respective condition. To verify this, we compare the results of our decoder with results from previous studies finding disease biomarkers.

For AD/MCI, previous studies have consistently pointed out to changes in the hippocampal and medial temporal lobe [9, 11, 52]. Accordingly, we found alterations in the inferior temporal, parahippocampal [58], fusiform [9], and paracentral lobule [52]. We observed that the most salient connections were from

the uncus for both AD and MCI, an anterior extremity of the parahippocampal gyrus known for disruption during both AD [59] and MCI [65]. Multiple functional networks such as the fronto-parietal task-control network, involved in attention and emotion regulation; the default mode network, for internal medication; the dorsal attention network, for directing external attention; and the salience network, which helps in emotion processing or monitoring salient events have been implicated in MDD [23, 46]. Deficits in cognitive control can be traced to the anomalies in the fronto-parietal task-control network, whereas too much internal rumination (and lesser engagement with the external world) may be reflected in the aberrant connectivity of the default mode network with other goal-directed networks. The amygdala [26, 36] (part of the subcallosal region), regions in the inferior temporal [56], the medial frontal lobe [56], posterior cingulate cortex [63] and supplementary motor area [64] have been implicated in MDD. In [61], disruption between the connections between regions in the thalamus and the transverse temporal gyrus were reported, which our consistent with our results.

Multiple studies have reported anomalies in the default-mode and dorsal attention networks in children and adults suffering with ADHD suggesting altered functional connectivity with attention and cognitive processing [45, 51, 53, 55]. The postcentral gyrus and paracentral lobule areas involved in motor functioning [45, 51], medial and inferior frontal gyrus [39, 45], superior occipital gyrus [55], and the insula [53] have been found to be different for children/young adoloscents having ADHD, which are consistent with our results.

Likewise, the frontal orbital regions and Heschl's gyrus are involved in sensory integration, speech processing and decision-making [24, 37], the putamen responsible for focusing attention [41], the cerebellar (and vermis) involved in motor regulation [3] have been previously identified as biomarkers for ASD.

## 5. Conclusion

In summary, we have proposed two algorithms - LEAN and CLIP - to reduce overfitting in DNNs making them more generalizable than models that use the entire feature set. Our approach leads us to an optimal neural network architecture in a single shot that is more efficient than previous methods that relied on recursive removal of features and nodes. Furthermore, via CLIP, the approach is customised to reduce the effects of correlated features that are present in neuroimaging datasets. Our experiments show that using both 'LEAN + CLIP' took into account both redundancy and correlation in input features and gave the best balance between drop in accuracy and reduction in trainable weights. We successfully applied the proposed algorithms on 4 datasets (and 5 different neurological disorders), showing its application in brain decoding and biomarker identification. The proposed approach has applications to both structural and functional neuroimaging datasets and to investigate brain disease.

**Acknowledgement**

This work was partially supported by AcRF Tier 1 grant RG 149/17 of Ministry of Education, Singapore. Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012), the International Neuroimaging Datasharing Initiative (INDI) and the Creativity and Affective Neuroscience Lab in the Brain Imaging Center of Southwest University. We would like to thank Dr. Jiang Qiu and Dr. Dongtao Wei from the Creativity and Affective Neuroscience Lab, Brain Imaging Center of Southwest University for providing us the dataset for MDD.

**References**

[1] Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.T., 2012. Learning from data. volume 4. AMLBook New York, NY, USA:.

[2] Albert, R., Barabási, A.L., 2002. Statistical mechanics of complex networks. Reviews of modern physics 74, 47.

[3] Allen, G., Courchesne, E., 2003. Differential effects of developmental cerebellar abnormality on cognitive and motor functions in the cerebellum: an fmri study of autism. American Journal of Psychiatry 160, 262–273.

[4] Anwar, S., Hwang, K., Sung, W., 2017. Structured pruning of deep convolutional neural networks. ACM Journal on Emerging Technologies in Computing Systems (JETC) 13, 32.

[5] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W., 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one 10, e0130140.

[6] Bellec, P., Chu, C., Chouinard-Decorte, F., Benhajali, Y., Margulies, D.S., Craddock, R.C., 2017. The neuro bureau ADHD-200 preprocessed repository. Neuroimage 144, 275–286.

[7] Birn, R.M., 2012. The role of physiological noise in resting-state functional connectivity. Neuroimage 62, 864–870.

[8] Brown, C.J., Kawahara, J., Hamarneh, G., 2018. Connectome priors in deep neural networks to predict autism, in: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), IEEE. pp. 110–113.

[9] Chan, D., Fox, N.C., Scahill, R.I., Crum, W.R., Whitwell, J.L., Leschziner, G., Rossor, A.M., Stevens, J.M., Cipolotti, L., Rossor, M.N., 2001. Patterns of temporal lobe atrophy in semantic dementia and Alzheimer's disease. Annals of neurology 49, 433–442.

[10] Clauset, A., Shalizi, C.R., Newman, M.E., 2009. Power-law distributions in empirical data. SIAM Review 51, 661–703.

[11] Convit, A., De Asis, J., De Leon, M., Tarshish, C., De Santi, S., Rusinek, H., 2000. Atrophy of the medial occipitotemporal, inferior, and middle temporal gyri in non-demented elderly predict decline to alzheimer's disease. Neurobiology of aging 21, 19–26.

[12] Crossley, N.A., Mechelli, A., Vértes, P.E., Winton-Brown, T.T., Patel, A.X., Ginestet, C.E., McGuire, P., Bullmore, E.T., 2013. Cognitive relevance of the community structure of the human brain functional coactivation network. Proceedings of the National Academy of Sciences 110, 11583–11588.

[13] Deco, G., Kringelbach, M.L., 2014. Great expectations: using whole-brain computational connectomics for understanding neuropsychiatric disorders. Neuron 84, 892–905.

[14] Dormann, C.F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., Marquéz, J.R.G., Gruber, B., Lafourcade, B., Leitão, P.J., et al., 2013. Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. Ecography 36, 27–46.

[15] Du, Y., Fu, Z., Calhoun, V.D., 2018. Classification and prediction of brain disorders using functional connectivity: promising but challenging. Frontiers in neuroscience 12.

[16] Gupta, S., Chan, Y.H., Rajapakse, J.C., Initiative, A.D.N., et al., 2019. Decoding brain functional connectivity implicated in AD and MCI, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 781–789.

[17] Gupta, S., Rajapakse, J.C., 2019. Iterative consensus spectral clustering improves detection of subject and group level brain functional modules. Scientific Reports (Under Review) -, –.

[18] Hebart, M.N., Baker, C.I., 2018. Deconstructing multivariate decoding for the study of brain function. Neuroimage 180, 4–18.

[19] Heinsfeld, A.S., Franco, A.R., Craddock, R.C., Buchweitz, A., Meneguzzi, F., 2018. Identification of autism spectrum disorder using deep learning and the ABIDE dataset. NeuroImage: Clinical 17, 16–23.

[20] Hinkle, D.E., Wiersma, W., Jurs, S.G., 2003. Applied statistics for the behavioral sciences. volume 663. Houghton Mifflin College Division.

[21] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., et al., 2012. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal processing magazine 29.

[22] Hyvärinen, A., Oja, E., 2000. Independent component analysis: algorithms and applications. Neural networks 13, 411–430.

[23] Kaiser, R.H., Andrews-Hanna, J.R., Spielberg, J.M., Warren, S.L., Sutton, B.P., Miller, G.A., Heller, W., Banich, M.T., 2014. Distracted and down: neural mechanisms of affective interference in subclinical depression. Social cognitive and affective neuroscience 10, 654–663.

[24] Kana, R.K., Keller, T.A., Cherkassky, V.L., Minshew, N.J., Just, M.A., 2009. Atypical frontal-posterior synchronization of theory of mind regions in autism during mental state attribution. Social neuroscience 4, 135–152.

[25] Kim, J., Calhoun, V.D., Shim, E., Lee, J.H., 2016. Deep neural network with weight sparsity control and pre-training extracts hierarchical features and enhances classification performance: Evidence from whole-brain resting-state functional connectivity patterns of schizophrenia. Neuroimage 124, 127–146.

[26] Kong, L., Chen, K., Tang, Y., Wu, F., Driesen, N., Womer, F., Fan, G., Ren, L., Jiang, W., Cao, Y., et al., 2013. Functional connectivity between the amygdala and prefrontal cortex in medication-naive individuals with major depressive disorder. Journal of psychiatry & neuroscience: JPN 38, 417.

[27] Koyamada, S., Shikauchi, Y., Nakae, K., Koyama, M., Ishii, S., 2015. Deep learning of fMRI big data: a novel approach to subject-transfer decoding. arXiv preprint arXiv:1502.00093 .

[28] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, pp. 1097–1105.

[29] Lehoucq, R.B., Sorensen, D.C., Yang, C., 1998. ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods. volume 6. Siam.

[30] Lundberg, S.M., Lee, S.I., 2017. A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, pp. 4765–4774.

[31] Meszlényi, R.J., Buza, K., Vidnyánszky, Z., 2017. Resting state fMRI functional connectivity-based classification using a convolutional neural network architecture. Frontiers in Neuroinformatics 11, 61.

[32] Mueller, S., Wang, D., Fox, M.D., Yeo, B.T., Sepulcre, J.e.a., 2013. Individual variability in functional connectivity architecture of the human brain. Neuron 77, 586–595.

[33] Poggio, T., Kawaguchi, K., Liao, Q., Miranda, B., Rosasco, L., Boix, X., Hidary, J., Mhaskar, H., 2017. Theory of deep learning iii: explaining the non-overfitting puzzle. arXiv preprint arXiv:1801.00173 .

[34] Power, J.D., Barnes, K.A., Snyder, A.Z., Schlaggar, B.L., Petersen, S.E., 2012. Spurious but systematic correlations in functional connectivity mri networks arise from subject motion. Neuroimage 59, 2142–2154.

[35] Power, J.D., Cohen, A.L., Nelson, S.M., Wig, G.S., et al., 2011. Functional network organization of the human brain. Neuron 72, 665–678.

[36] Ramasubbu, R., Konduru, N., Cortese, F., Bray, S., Gaxiola, I., Goodyear, B., 2014. Reduced intrinsic connectivity of amygdala in adults with major depressive disorder. Frontiers in psychiatry 5, 17.

[37] Rausch, A., Zhang, W., Haak, K.V., Mennes, M., Hermans, E.J., van Oort, E., van Wingen, G., Beckmann, C.F., Buitelaar, J.K., Groen, W.B., 2016. Altered functional connectivity of the amygdaloid input nuclei in adolescents and young adults with autism spectrum disorder: a resting state fmri study. Molecular autism 7, 13.

[38] Rosa, M.J., Portugal, L., Hahn, T., Fallgatter, A.J., Garrido, M.I., Shawe-Taylor, J., Mourao-Miranda, J., 2015. Sparse network-based models for patient classification using fMRI. Neuroimage 105, 493–506.

[39] Rubia, K., Halari, R., Christakou, A., Taylor, E., 2009. Impulsiveness as a timing disturbance: neurocognitive abnormalities in attention-deficit hyperactivity disorder during temporal processes and normalization with methylphenidate. Philosophical Transactions of the Royal Society B: Biological Sciences 364, 1919–1931.

[40] Rubin, T.N., Koyejo, O., Gorgolewski, K.J., Jones, M.N., Poldrack, R.A., Yarkoni, T., 2017. Decoding brain activity using a large-scale probabilistic functional-anatomical atlas of human cognition. PLoS computational biology 13, e1005649.

[41] Sato, W., Kubota, Y., Kochiyama, T., Uono, S., Yoshimura, S., Sawada, R., Sakihama, M., Toichi, M., 2014. Increased putamen volume in adults with autism spectrum disorder. Frontiers in human neuroscience 8, 957.

[42] Satopaa, V., Albrecht, J., Irwin, D., Raghavan, B., 2011. Finding a" kneedle" in a haystack: Detecting knee points in system behavior, in: 2011 31st international conference on distributed computing systems workshops, IEEE. pp. 166–171.

[43] Sen, B., Chu, S.H., Parhi, K.K., 2019. Ranking regions, edges and classifying tasks in functional brain graphs by sub-graph entropy. Scientific reports 9, 7628.

[44] Shrikumar, A., Greenside, P., Kundaje, A., 2017. Learning important features through propagating activation differences, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR.org. pp. 3145–3153.

[45] Smith, A.B., Taylor, E., Brammer, M., Halari, R., Rubia, K., 2008. Reduced activation in right lateral prefrontal cortex and anterior cingulate gyrus in medication-naïve adolescents with attention deficit hyperactivity disorder during time discrimination. Journal of Child Psychology and Psychiatry 49, 977–985.

[46] Snyder, H.R., 2013. Major depressive disorder is associated with broad impairments on neuropsychological measures of executive function: a meta-analysis and review. Psychological bulletin 139, 81.

[47] Sporns, O., Betzel, R.F., 2016. Modular brain networks. Annual Review of Psychology 67, 613–640.

[48] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15, 1929–1958.

[49] Stella, X.Y., Shi, J., 2003. Multiclass spectral clustering, in: Proc. of International Conference on Computer Vision, IEEE. pp. 313–319.

[50] Sundararajan, M., Taly, A., Yan, Q., 2017. Axiomatic attribution for deep networks, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR.org. pp. 3319–3328.

[51] Suskauer, S.J., Simmonds, D.J., Caffo, B.S., Denckla, M.B., Pekar, J.J., Mostofsky, S.H., 2008. fmri of intrasubject variability in adhd: anomalous premotor activity with prefrontal compensation. Journal of the American Academy of Child & Adolescent Psychiatry 47, 1141–1150.

[52] Thiyagesh, S.N., Farrow, T.F., Parks, R.W., Accosta-Mesa, H., Young, C., Wilkinson, I.D., Hunter, M.D., Woodruff, P.W., 2009. The neural basis of visuospatial perception in Alzheimer's disease and healthy elderly comparison subjects: an fMRI study. Psychiatry Research: Neuroimaging 172, 109–116.

[53] Tian, L., Jiang, T., Wang, Y., Zang, Y., He, Y., Liang, M., Sui, M., Cao, Q., Hu, S., Peng, M., et al., 2006. Altered resting-state functional connectivity patterns of anterior cingulate cortex in adolescents with attention deficit hyperactivity disorder. Neuroscience letters 400, 39–43.

[54] Toloşi, L., Lengauer, T., 2011. Classification with correlated features: unreliability of feature ranking and solutions. Bioinformatics 27, 1986–1994.

[55] Tomasi, D., Volkow, N.D., 2012. Abnormal functional connectivity in children with attention-deficit/hyperactivity disorder. Biological psychiatry 71, 443–450.

[56] Townsend, J.D., Eberhart, N.K., Bookheimer, S.Y., Eisenberger, N.I., Foland-Ross, L.C., Cook, I.A., Sugar, C.A., Altshuler, L.L., 2010. fmri activation in the amygdala and the orbitofrontal cortex in unmedicated subjects with major depressive disorder. Psychiatry Research: Neuroimaging 183, 209–217.

[57] Varoquaux, G., 2018. Cross-validation failure: small sample sizes lead to large error bars. Neuroimage 180, 68–77.

[58] Wang, L., Zang, Y., He, Y., Liang, M., Zhang, X., Tian, L., Wu, T., Jiang, T., Li, K., 2006. Changes in hippocampal connectivity in the early stages of Alzheimer's disease: evidence from resting state fMRI. Neuroimage 31, 496–504.

[59] Wang, Z., Zhang, M., Han, Y., Song, H., Guo, R., Li, K., 2016. Differentially disrupted functional connectivity of the subregions of the amygdala in alzheimer's disease. Journal of X-ray Science and Technology 24, 329–342.

[60] Weigend, A., 1994. On overfitting and the effective number of hidden units, in: Proceedings of the 1993 connectionist models summer school, pp. 335–342.

[61] Xue, S.W., Wang, D., Tan, Z., Wang, Y., Lian, Z., Sun, Y., Hu, X., Wang, X., Zhou, X., 2019. Disrupted brain entropy and functional connectivity patterns of thalamic subregions in major depressive disorder. Neuropsychiatric Disease and Treatment 15, 2629.

[62] Yan, K., Zhang, D., 2015. Feature selection and analysis on correlated gas sensor data with recursive feature elimination. Sensors and Actuators B: Chemical 212, 353–363.

[63] Yao, Z., Wang, L., Lu, Q., Liu, H., Teng, G., 2009. Regional homogeneity in depression and its relationship with separate depressive symptom clusters: a resting-state fMRI study. Journal of affective disorders 115, 430–438.

[64] Zhang, B., Li, M., Qin, W., Demenescu, L.R., Metzger, C.D., Bogerts, B., Yu, C., Walter, M., 2016. Altered functional connectivity density in major depressive disorder at rest. European archives of psychiatry and clinical neuroscience 266, 239–248.

[65] Zhao, Z., Lu, J., Jia, X., Chao, W., Han, Y., Jia, J., Li, K., 2014. Selective changes of resting-state brain oscillations in aMCI: an fMRI study using ALFF. BioMed Research International 2014.