Metanets: uncovering functional brain networks with artificial neural networks

Nicholas Tolley: nicholas_tolley@brown.edu
Annabeth Stokely: annabeth_stokely@brown.edu
Robert Zielinski: robert_zielinski1@brown.edu
David Chu: david_chu1@brown.edu

Introduction

Functional magnetic resonance imaging (fMRI) is a fundamental neuroimaging technique used to non-invasively measure whole-brain activity. A primary research goal with fMRI is to characterize statistical relationships in the activity between brain regions, i.e. functional connectivity. Unlike structural connectivity which describes physical axonal connections between neurons, functional connectivity is a data-driven description of co-active brain areas. Currently the leading techniques to identify functional brain networks largely depend on correlative measures. While these techniques have successfully identified robust brain networks such as the default mode network (DMN), there are potentially more complex functional connectivity patterns that are shared across individuals, and reveal interactions between regions that are missed by correlation based measures.

The goal of this project was to perform unsupervised learning with state-of-the-art deep learning architectures to identify fMRI based functional networks.

Methods

Data

We chose to use the developmental fMRI dataset from the python package nilearn, sourced from the paper “Development of the social brain from age three to twelve years” (Richardson et al., 2018). This dataset contains the fMRI results from patients of varying ages after they watched the movie short “Partly Cloudy”. 155 subjects in total were recorded, 122 children between the ages of 3 and 12 and 33 adults. We used the preprocessing available with the nilearn package. Raw fMRI blood oxygen level dependent images were skull-stripped, and normalized for head movement, intensity, and time. The raw voxels of the scan were then mapped to an anatomical atlas of 39 brain regions. This kind of preprocessing is standard for neuroscience datasets. The resulting data represents scans of all 155 subjects across about 150 time points. We loaded the data via the package, and then split it for training, testing, and validation purposes across subjects rather than across sections of the movie.

Architecture

For this project, we focused on two tasks: 1) learning low dimensional representations of fMRI data, and 2) comparing these representations–functional networks of brain activity– across the latent dimensions of each architecture. For these tasks we chose to use autoencoders with differing encoder/decoder structures.

In choosing the candidate models to be compared for this project, we considered several factors. First, all the models selected had a demonstrated applicability to fMRI data for evaluating functional connectivity in some population. This ensured that we were able to obtain a reasonable level of performance with that model. Second, each model had distinct attributes in the model architecture that justify inclusion in the comparison. These were not simple changes in hyperparameters, but rather more fundamental differences in the modeling techniques or calculations used.

Several variants of autoencoders were included. The critical distinction between each model was the architecture of the encoder/decoder layers. Since fMRI consists of multidimensional time series data, the architectures we assessed included 1) feed forward networks, 2) LSTM, 3) Mamba/SSM's (Gu & Dao, 2023).

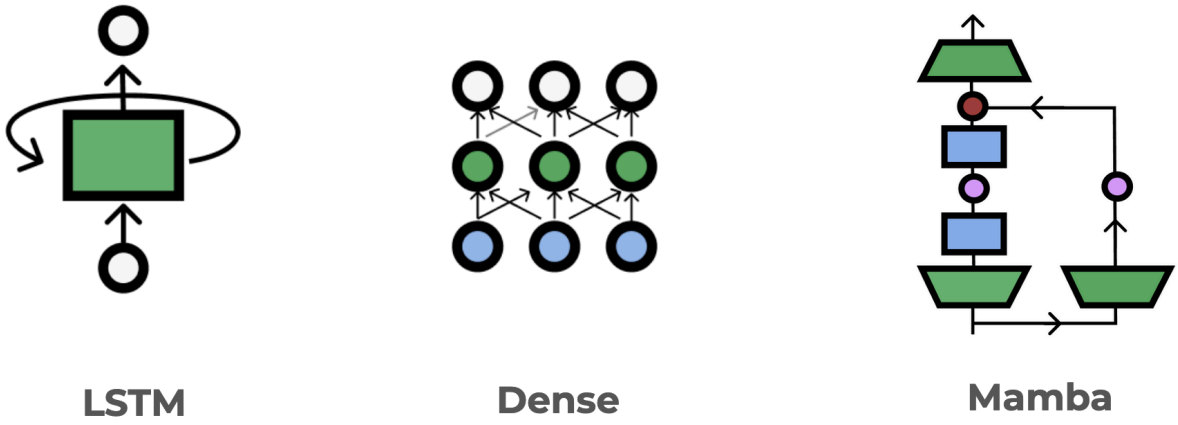

Figure 1: Model architecture overview
This diagram displays a visualization of one encoder/decoder block of each of the architectures we evaluated.

Dense

We used our dense layer model as a baseline for comparison to the performance of our other models. Our autoencoder had one encoder block with three hidden layers of sizes 100, 50, and 25, and one decoder block with increasing sizes of the same step.

LSTM

We chose to test an LSTM architecture because of its superior performance on time-series data compared to dense neural networks. Our LSTM architecture had one encoder block and one decoder block, each with two layers.

Mamba/SSM

We sourced our Mamba architecture from the paper “Mamba: Linear-Time Sequence Modeling with Selective State Spaces” (Gu & Dao, 2023). This architecture seeks to combine the benefits of the selective state space model (SSM) and the multi-layer perceptron block of a transformer to improve performance and training time on sequence data. Its main advantage is the ability to handle very long sequences of data in comparison to previous architectures via memory efficiency. Our Mamba architecture had one encoder block and one decoder block. Each encoder/decoder block includes two linear projections. The first linear projection leads into a convolution layer, followed by a sigmoid activation, followed by a state space model. The second linear projection leads to a sigmoid activation. These two paths are combined via multiplication before being linearly projected back to the original dimensions. Our SSM state expansion factor was 16, our local convolution width was 4, and our block expansion factor was 2.

For further comparison, we tested a range of values for the learning rate, hidden layer size, and weight decay of this model. Plots of these tests can be found in Figure 5.

Results

Our first goal was to apply the previously established PCA based methods for extracting brain (eigen) networks from fMRI data (Friston et al., 2014). The brain networks identified by PCA correspond to the loadings on the principal components. Figure 1 shows an example of the full 39 dimensional fMRI time series, where each dimension corresponds to a distinct brain region.

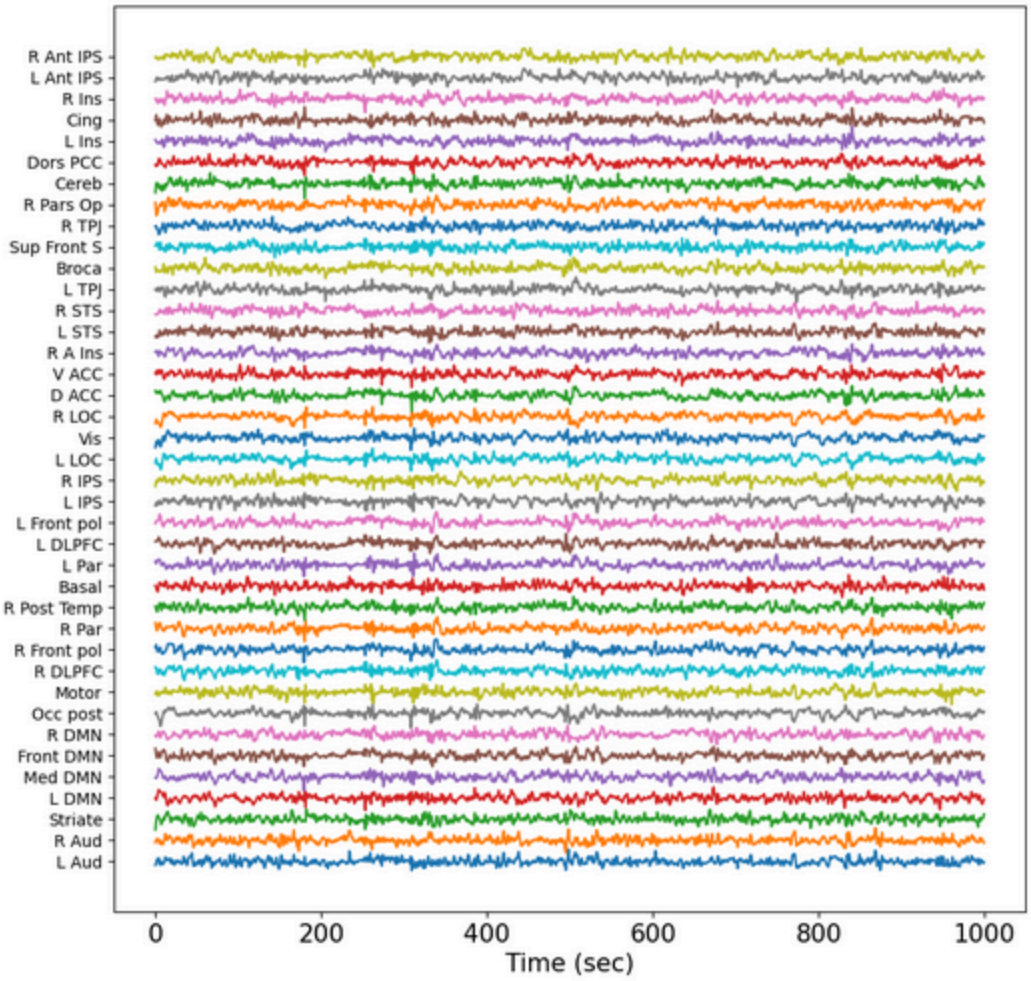

Figure 1: Resting state fMRI time series taken from developmental dataset
An example of the fMRI time series used in this study is shown. Raw voxel activity from every subject in the developmental dataset was mapped to 39 predefined brain regions using a reference anatomical atlas.

After applying PCA to the full dataset, we assessed the degree to which the dimensionality of the dataset could be reduced, providing a baseline to compare our autoencoder architectures to. As shown in Figure 2, the first 16 principal components account for 80% of the variance explained. Given that the recordings correspond to a 39 dimensional time series, this suggests that the dataset is not easily reduced to a lower dimensional representation through correlation based methods like PCA.

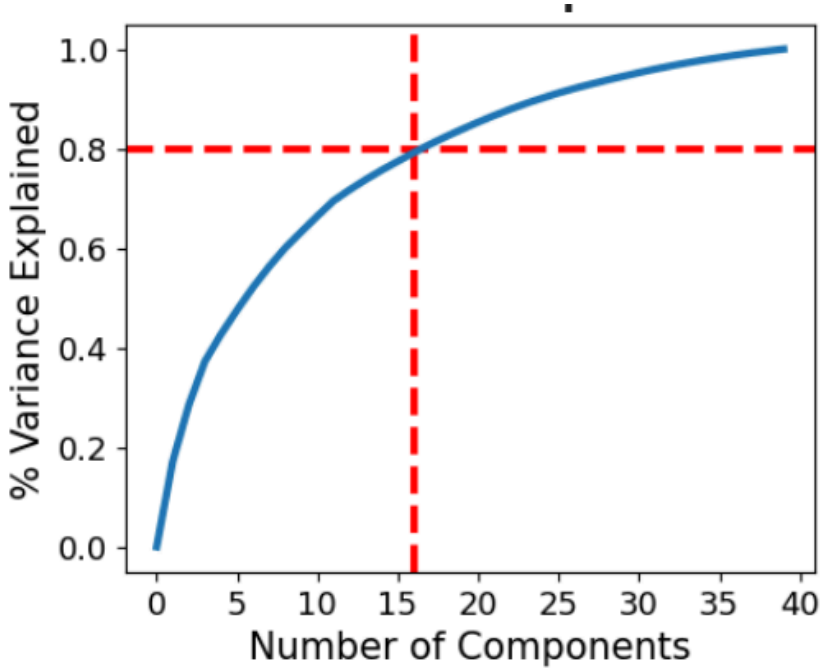

Figure 2: Dimensionality reduction of fMRI time series by PCA
Applying PCA to the full developmental dataset demonstrates that 80% of the variance explained is accounted for by 16 principal components.

Figure 3 shows the eigen networks extracted from the first 2 principal components. The first eigen network (Figure 3 left) explains the highest amount of variance in the dataset, and indicates a brain wide network with nodes primarily in the parietal and prefrontal cortex. Further, all of the nodes are positively (red) correlated with each other, as indicated by the edges. This is contrasted with the second principal components (Figure 3 right), where there are two distinct clusters in visual cortex, and prefrontal cortex. Within the clusters, the brain regions are positively correlated with each, but across the clusters the brain regions are negatively (blue) correlated.

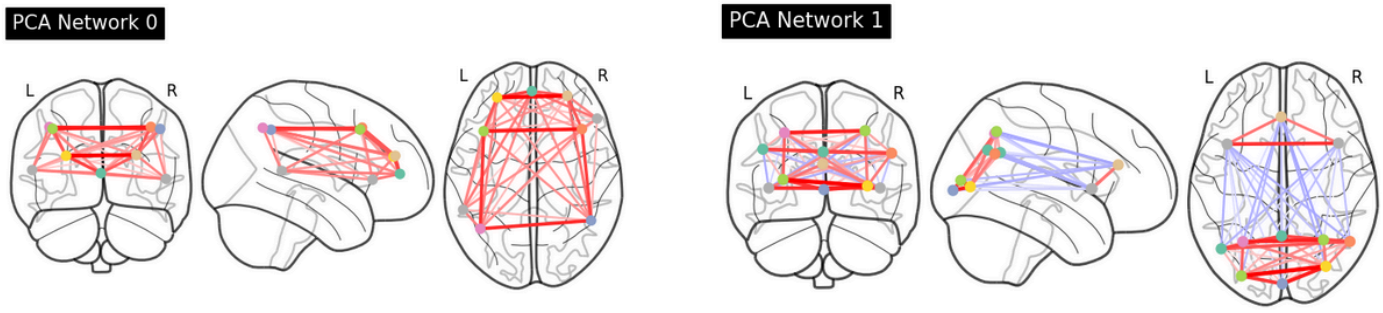

Figure 3: Eigen networks extracted from PCA
The eigen networks corresponding to the first two principal components PC1 (left) and PC2 (right).

With the PCA baseline established, we then compared the autoencoder architectures by assessing their reconstruction error with respect to the bottleneck dimensionality. As shown, none of the deep learning architectures beat PCA with respect to dimensionality reduction, with the LSTM architecture performing substantially worse. The Mamba autoencoder did achieve marginally lower reconstruction errors on the training set, however, it performed identically as PCA on the test set (assessed via 5 fold cross validation with an 80/20 train/test split).

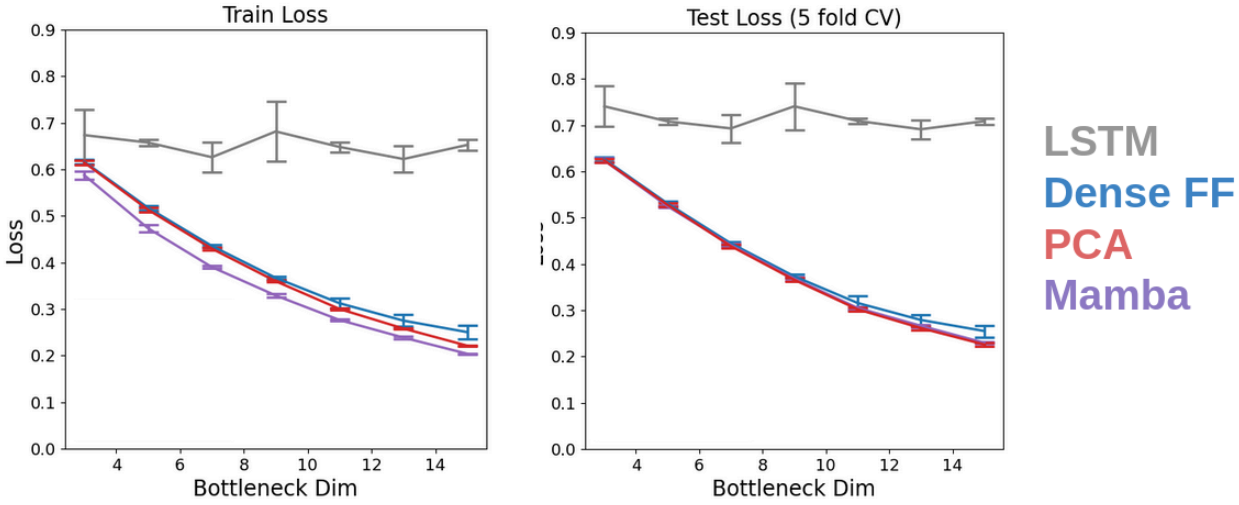

Figure 4: Comparison of autoencoder architectures on reconstruction loss

Given the high performance of the mamba architecture, we then tuned the hyperparameters to identify the best performing model before extracting brain networks from the learned representations. As shown in Figure 5, the optimal hyperparameters were a learning rate of 10e-4, a weight decay of 0.0, and a hidden size of 100 units.

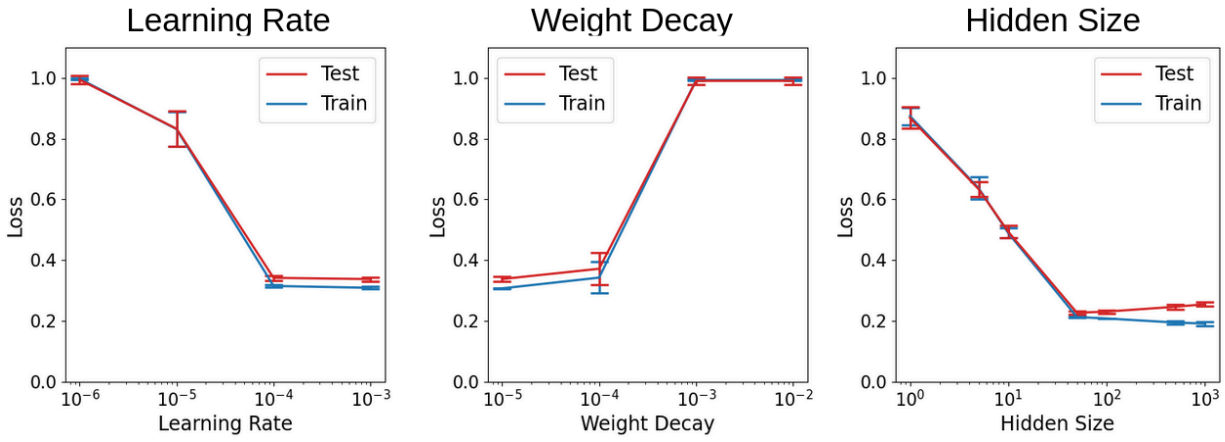

Figure 5: Hyperparameter tuning on Mamba autoencoder

Next we extracted brain networks from the mamba autoencoder to compare directly to the eigen networks identified via PCA. Specifically, we took the decoder network from the autoencoder and probed each hidden unit in the bottleneck layer separately. The variance of the outputs was used to assess which brain regions were sensitive to changes in each bottleneck unit, i.e. the brain network encoded by that dimension of the bottleneck layer. To compare the brain networks between mamba and PCA, we calculated the correlation coefficient between the eigen network's values (loading on each PC), and the sensitivity (variance) of each brain region to the Mamba decoder's bottleneck dimension. As shown in Figure 7 (left), the two approaches learn largely dissimilar brain networks, with the highest correlation between two networks being r=0.165. Figure 7 (right) plots the most similar brain networks identified by PCA and Mamba. As shown, both networks exhibit a cluster of positively correlated regions in the visual cortex, with a left lateralized collection of brain regions in the prefrontal and sensorimotor cortex.

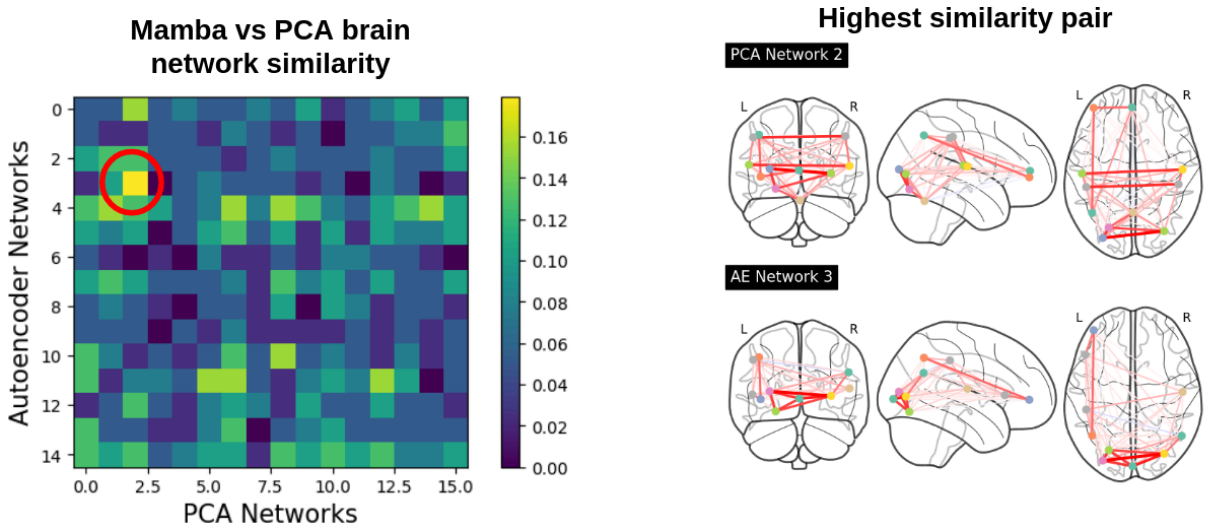

Figure 7: Comparison between brain networks identified by Mamba autoencoders and PCA

Challenges: What has been the hardest part of the project you've encountered so far?

The challenges we encountered during the project largely fell into three broad categories. First, finding an appropriate dataset was a nontrivial task. Due to the cost of collecting large amounts of fMRI data and the potential privacy concerns associated with the use of health data, freely available fMRI datasets that include enough data to train deep learning models are relatively scarce. We originally chose a dataset with preprocessed BOLD data that was collected from an interesting experiment related to movie watching. However, this dataset was compressed using a deprecated python package, making it incompatible with the other tools we were using to run the analysis. We ultimately decided to use data made available through the nilearn python package which includes BOLD data collected from 155 subjects.

The next challenge we faced was fully understanding the steps needed to appropriately summarize the fMRI data in terms of a set number of brain regions. While many of the necessary steps, such as skull stripping the images, had already been taken, there were several

steps required to process the voxel-level data into region-specific time series data. Each of these steps, like registering the data to a common template, involves decisions that determine which transformations are applied to the data. Individual judgment is needed to ensure that the signals present in the unprocessed data are still present at the end of the preprocessing pipeline. This required substantial efforts to make sure that, when considering the models we were using, we were undertaking the steps that would yield reasonable results.

A third set of challenges we encountered related to the computational demands of other models we were interested in implementing. Specifically, we intended to compare the models for which results are shown with variational autoencoder architectures that would be applied to the voxel-level data instead of the region-level data used for the other models (Qiang et al., 2021). Because of the level of summarization present in the region-level data, these alternative models could have the benefit of retaining more of the information that was originally present in the data, which could allow us to uncover networks present between subregions that we would not be able to identify with the region-level data. Additionally, using the region-level data makes the implicit assumption that summarizing based on brain regions is appropriate in this setting. While there are strong reasons to believe that this will lead to viable results, the use of voxel-level data would allow the model more flexibility in identifying regions most relevant to functional connectivity, which could yield interesting results.

Ultimately, implementing the architectures based on voxel-level data presented computational needs that we could not meet using the tools available to us, even when taking steps to minimize the model size and batch size. It is possible that some intermediate processing steps could have been taken that would reduce the computational burden of training these models without relying on complete region-level summarization, but exploring this option was left for future work.

Discussion:

We have a fully trained autoencoder, one with fully connected network encoders/decoders as well as Mamba based encoders/decoders. We've split our data such that our validation set has completely different subjects to our training and test data, and our models are still able to make correlations. This indicates we're finding connections between entirely different brains, which is encouraging. Further, quantitative comparison between autoencoder based and correlation (PCA) based connectivity identifies a small number of overlapping networks. However, these two approaches appear to primarily identify disjoint sets of functional networks, potentially suggesting that the autoencoder is learning nonlinear interactions unaddressed by correlation based approaches.

Reflection

Overall, our project was a success, *potentially* surpassing our stretch goal of identifying novel functional networks. For our baseline, our Mamba autoencoder successfully identified low-dimensional representations of fMRI data. Using this autoencoder, we successfully

characterized functional networks, discovering a set that is distinct from those found using PCA. This *intriguing result* suggests that the autoencoder is learning novel networks.

However, we were equally surprised to find that PCA performed just as well as our best autoencoder in terms of loss and dimensionality reduction. This could be because the dataset/model size was insufficient for a deep learning task, or more complex relationships did not exist in the data (or we removed important information in the preprocessing step). In terms of our biggest surprises, we were most surprised by the difficulty of the preprocessing step. We found it difficult to successfully train a Variational Autoencoder using raw voxel-level data as it would require substantial computing costs. However, in preprocessing, we were potentially removing meaningful information from the data.

Our approach mainly changed in our choice of dataset. At first, we used a dataset with preprocessed BOLD data that was compressed using Python2 (Busch et al., 2021), which prompted us to change the dataset to a dataset from the nilearn package. If we could do this project again (and given more time/computation), we would experiment with training a VAE on voxel-level data without the extensive preprocessing.

Undertaking this project was an incredibly valuable experience for us. It provided a high-level understanding of the challenges associated with fMRI data and the practical applications of identifying functional networks. At a more detailed level, it allowed us to gain firsthand experience of the complexities of data preprocessing and the instances where deep neural networks may not outperform simpler methods like PCA. This project has undoubtedly enriched our knowledge and skills in the field.

References

Busch, E. L., Slipski, L., Feilong, M., Guntupalli, J. S., Castello, M. V. di O., Huckins, J. F., Nastase, S. A., Gobbini, M. I., Wager, T. D., & Haxby, J. V. (2021). Hybrid hyperalignment: A single high-dimensional model of shared information embedded in cortical patterns of response and functional connectivity. *NeuroImage*, *233*, 117975. https://doi.org/10.1016/j.neuroimage.2021.117975

Friston, K. J., Kahan, J., Razi, A., Stephan, K. E., & Sporns, O. (2014). On nodes and modes in resting state fMRI. *NeuroImage*, *99*, 533–547. https://doi.org/10.1016/j.neuroimage.2014.05.056

Gu, A., & Dao, T. (2023). *Mamba: Linear-Time Sequence Modeling with Selective State Spaces* (arXiv:2312.00752). arXiv. https://doi.org/10.48550/arXiv.2312.00752

Richardson, H., Lisandrelli, G., Riobueno-Naylor, A., & Saxe, R. (2018). Development of the

social brain from age three to twelve years. *Nature Communications*, *9*(1), 1027. https://doi.org/10.1038/s41467-018-03399-2

Qiang, N. *et al*., (2021). Deep Variational Autoencoder for Mapping Functional Brain Networks. *IEEE Transactions on Cognitive and Developmental Systems*, 13(4), pp. 841-852, https://doi.org/10.1109/TCDS.2020.3025137.