Computing directed connectivity by predicting fMRI signals using attention based artificial neural networks

Jost Triller

Abstract

In this work, I explore a method to extract directed connectivity information from fMRI resting state signals. As an alternative to, for example, correlation and Granger causality based approaches, this new method is based on an artificial neural network (ANN) model using a combination of dense fully connected layers and attention mechanisms. The ANN is trained to predict future fMRI signals, and the attention matrix during inference is used to describe connectivity between regions. The resulting attention connectivity matrices compare competitively to Granger causality and Pearson and partial correlation on fingerprinting and on predicting individual behavior. The code for this project is available on GitHub.[1]

1 Introduction

Functional connectivity (FC), as a way of mapping how parts of the brain work together, has been an active area of research. One common approach is to use fMRI to capture blood oxygenation levels in the brain as an indirect measure of brain activity (Buckner et al., 2013). These spatio-temporal signals can be analyzed using statistical methods for correlations between signals of different brain regions. The resulting information (usually a square matrix whose value at (a, b) represents how connected two regions a and b are) can, for example, be used to infer the existence of distinct large-scale networks (Fox et al., 2005) or to predict various mental diseases or cognitive and behavioral traits and assign importance weights to edges between specific regions regarding these different mental properties (Shen et al., 2017).

A central question is if and how the underlying causal mechanisms between neural units of the brain are represented in the functional connectivity calculated from fMRI signals and said statistical methods. Commonly used correlation based methods, such as Pearson correlation, result in undirected connectivity graphs, meaning that it is very hard to make statements about causal inference. There are however approaches to extract directed information from brain signals, such as Granger causality or dynamic causal modeling (Stephan & Roebroeck, 2012).

To expand in this direction, in this work, I propose a new method to compute directed functional connectivity, i.e., an algorithm that takes signals of regional brain activity over time as input, and produces a non-symmetric square matrix as output. The intention is that the value at (a, b) in the matrix represents how strong the influence of the past activity of region b is on the future activity of region a.

[1] https://github.com/tsoj/fmri_attention_connectivity

The fundamental idea this new method is motivated by is the hypothesis that if knowing the past of region b helps predicting the future activity of region a, then it is likely that regions a and b are in some capacity connected. If at the same time knowing the past of a helps less to predict b, it might mean that in the ground truth system that produces the signal, b is less influenced by a than a is influenced by b, or in other words, we could make assumptions about the flow of information between brain regions.

The new method is implemented using an artificial neural network (ANN) model and machine learning methods to train the model to predict future fMRI signals from past signals. The connectivity information is extracted from a layer of the model that implements a form of the attention mechanism. This mechanism allows the model for each region a to dynamically select from other input regions the ones which might be most useful for predicting the future of region a. The model is trained either on multi-subject data to learn to predict the population average brain signal, or it is trained on single subject data to learn to predict brain patterns of a specific subject.

The rest of this report is structured as follows:

- **Relevant work**: An introduction to various functional connectivity mapping methods, work relevant to predicting fMRI time series, and previous approaches using attention mechanisms for fMRI analysis.
- **Method**: Description of the data used and the new algorithm.
- **Evaluation**: Description of the fingerprinting and predictive modeling of behavioral data test approaches, and results compared to Pearson and partial correlation, and a linear model similar to Granger causality.
- **Further results**: Future time step prediction power, hyperparameter effects, synthetic dataset with ground truth connectivity.

2 Relevant work

2.1 Attention mechanism

Since the advent of the transformer architecture (Vaswani et al., 2017), attention based ANNs have seen a large rise in popularity because of their strong performance across many fields (Brown et al., 2020; Dosovitskiy et al., 2020; Jumper et al., 2021; Rombach et al., 2021).

The attention mechanism is used if different entities (e.g., brain regions or the embedding vectors of different brain regions) in the network architecture need to share information. To search for the most useful information other regions provide, a region a creates a query vector (Q_a). At the same time, all other regions (e.g., region b) generate a key and a value vector (K_b and V_b). The key K_b is the vector that "publicizes" what kind of information V_b region b provides; in other words, K_b is a key for accessing V_b. To find out how "important" information from another region b is for region a, we compute the dot product of the query of a and the key of b: $\text{importance}_{ab} = Q_a K_b^T$. Now we sum the value vectors of all regions together, weighted by their importance: $\text{aggregated\_information}_a = \sum_{b \in \text{regions}} (Q_a K_b^T V_b)$. The most common variant of attention additionally uses softmax and a dimension specific scaling factor.

2.2 Predicting fMRI time series

There has been some previous work on predicting future time steps of fMRI data. The classical example is the VAR for Granger causality. More recently, machine learning using artificial neural networks (ANNs) for fMRI time series prediction has been a popular topic

of research. The models usually tend to be deep learning architectures (i.e., multiple hidden layers), such as graph neural networks (GNN) (Wein et al., 2022), recurrent neural networks (RNNs) (Sobczak et al., 2020), and more recently transformers (Zheng et al., 2024), (Sun et al., 2024), (Sun et al., 2025). Because of their deep architectures, these models are usually hard to interpret. (Wein et al., 2021) extract (potential) causal connectivity via perturbation analysis (perturbing the input at selected regions and observing for which regions the output changes). They show plausible causal connectivity related to the vestibular cortex. Additionally there are other neural network based approaches, for example, to improve Granger causality with neural networks to predict brain activity time series (Mamoon et al., 2025). Instead of trying to predicting future time steps using a linear and static coefficients, here a neural network is trained to produce dynamic, non-linear coefficients. They achieve very good results on autism and mild cognitive impairment classification tasks. Other recent work, CaLLTiF, is a constraint-based causal discovery method (Arab et al., 2025). They find very good performance on simulated fMRI from the macaque connectome and the learned connectomes show similar connections across individuals. Because of this large body of existing work, I will focus on basic simple and widely used approaches. Detailed comparisons with these more advanced connectivity methods is out of scope for this paper, especially because of the heterogeneity of frameworks and evaluations; direct comparisons are left for future work.

2.3 Existing functional connectivity methods

2.3.1 Pearson correlation

Pearson correlation is a very commonly used method to estimate functional connectivity. It is easy to understand and implement. It is effectively a normalized version of the covariance. The covariance between two signals A and B can be computed with:

$$\mathrm{cov}(A, B) = \sum_i ((A_i - \mu_A)(B_i - \mu_B))$$

where A_i is the i-th value of signal A, and μ_A is the mean value of A (likewise for B)

In other words, it is the dot product of two mean-subtracted signals. Now, to calculate the Pearson correlation, before we put the signals through the covariance formula, we z-normalize them individually:

$$A' = \frac{A - \mu_A}{\sigma_A}, B' = \frac{B - \mu_B}{\sigma_B}$$
$$\mathrm{Pearson}(A, B) = \mathrm{cov}(A', B')$$

Covariance and variants like the Pearson correlation have been used to theorize the existence of large-scale networks (Greicius et al., 2003), (Fox et al., 2005), for the creation of brain parcellations (Shen et al., 2013; Thomas Yeo et al., 2011), and for predictive modeling (Shen et al., 2017).

2.3.2 Partial correlation

The idea behind partial correlation is to only correlate the part of two signals with each other that can't be explained by the rest of all other available signals. Concretely, this means that given two signals A and B which we want to correlate, and a set of other signals Z, we first compute the signals $\hat{A} = Z\beta_A, \hat{B} = Z\beta_B$ with β_A and β_B being optimized via ordinary least squares ($\min_{\beta_K}\left((K - Z\beta_K)^2\right)$, with $K \in \{A, B\}$). Then we can compute the partial correlation between A and B by computing Pearson correlation between $A - \hat{A}$ and $B - \hat{B}$, in other words, we correlate the residuals of what can't be linearly predicted by Z.

Partial correlation instead of the simple covariance has been used in the attempt to uncover direct links between brain regions, which are obscured by the high correlation between non-direct links (e.g., common drivers) when using covariance measures (Marrelec et al., 2006) A note for clarity: *Direct* connection means that there are no other links between two regions. *Directed* on the other hand means that there is directionality to the connection. Partial correlation based connectivity seems to align well with biological similarity networks and performs well on identifiability and for predicting cognition and behavioral traits (Liu et al., 2025). However, compared to covariance based measures, partial correlation performs poorly on test-retest reliability tests (Liu et al., 2025).

2.3.3 Granger causality

To uncover directed causal links between brain regions, Granger causality inspired methods have been proposed. For Granger causality, it is assumed that information we collect from the causing region at the time of the cause can be used to predict the future of the effect at another region. As an example, given signals A, B, C one way to find out if B Granger-causes A, is to minimize the following two losses:

$$L_{ABC} = \min_\beta \left(\left(A_i - \sum_{j \geq 1} \left(\beta_{A_j} A_{i-j} + \beta_{B_j} B_{i-j} + \beta_{C_j} C_{i-j} \right) \right)^2 \right)$$

$$L_{AC} = \min_\beta \left(\left(A_i - \sum_{j \geq 1} \left(\beta_{A_j} A_{i-j} + \beta_{C_j} C_{i-j} \right) \right)^2 \right)$$

(notably the B part is missing in the second loss equation).

These linear models are sometimes called vector autoregression (VAR) models. Now, if $L_{ABC} < L_{AC}$ (i.e. the error when optimizing with B is lower) we can assume that B contains *unique* information useful for predicting A's future. Thus, we say that B Granger-causes A. Importantly, this can be different from a true causal relationship. Also, this example ignores issues of overfitting and noise for brevity.

The understanding of effective connectivity (directed functional connectivity) has been a main application of Granger causality in neuroscience. Examples are the attempt to uncover influence of selected regions on each other during task based fMRI (Roebroeck et al., 2005), classifying multiple sclerosis using a directed brain network (Azarmi et al., 2019), and, similarly to the correlation based human connectome, the creation of a directed connectome of the brain (Duggento et al., 2018).

It should be noted that Granger causality might be sensitive to heterogeneous haemodynamic response functions (HRF) across the brain, and thus results should be interpreted with caution (Deshpande & Hu, 2012; Smith et al., 2012). However, there is some evidence that the likelihood of false positive connections being found by Granger causality analysis because of HRF distortions is low (Novelli et al., 2025).

2.4 Attention based connectivity

Since attention lets the ANN decide on its own (during training) which regions are important for other regions, this mechanism presents itself as a natural method to extract connectivity information from data.

One such approach is DICE, which has been exploring the use of spatial attention (region to region) as a way to extract FC by training on tasks such as predicting schizophrenia, autism, age, or gender from fMRI data. The model architecture takes in fMRI data, uses RNNs and temporal attention to create per-region embeddings that are then used to create a spatial (cross-region) attention matrix. This matrix is used to predict the target.

The model is trained end-to-end, i.e., the spatial attention matrix is learned to be specifically optimized for the task at hand (e.g., classifying autism). After the model is trained, the spatial attention matrix can be used to infer important connectivity specifically for the trained task. This, for example, reveals that for classifying dementia and gender, sensorimotor and default-mode connectivity is important (Mahmood et al., 2022).

Another model architecture with a similar approach is DSAM. It is also intended to be a task specific model, trained end-to-end on tasks such as predicting gender. The architecture consists first of a network block that combines temporal (per region) convolutional and attention layers to select important points from the fMRI time series. These per-region important points are used to build a connectivity matrix using spatial attention. Finally, this connectivity matrix is used as an input for a GNN, which is used to classify a target (e.g., gender). Similarly to DICE, the DSAM connectivity matrix can be interpreted in the context of being learned specifically for the given task (Thapaliya et al., 2025).

(Nauta et al., 2019) describe an attention mechanism to capture causal relationships in time series: A network is trained to predict future timesteps using an attention architecture. After training, the trained attention weights are used to intepret, which part of which timeseries is used to predict a future value. It is in principle similar to my approach, however, it hasn't been applied on real fMRI data, and details in the architecture training differ. Importantly, here I use the trained model on unseen data to uncover directed connectivity; therefore it is possible to apply the trained model dynamically.

2.4.1 Benchmarking functional connectivity methods

Since even beyond the examples of (effective) functional connectivity mentioned above, there are many other approaches for building a connectome, the question arises, which of these many approaches is best for a given task. One approach used to answer that question is to build an artificial system, modelling the brain to some degree. This provides a ground truth connectivity that we can compare the outputs of different functional connectivity methods to (Wang et al., 2014). However, the effectiveness of this approach depends on the decision of simulation model. A more empirically grounded way to determine the usefulness of FC methods is to evaluate them on objective measures of quality on real fMRI data. Examples for such objective measures are fingerprinting accuracy, brain-behavior prediction, test-retest reliability, or sensitivity to motion artifacts (Liu et al., 2025; Mahadevan et al., 2021). Good scores on these measures imply a decent likelihood for an FC method to capture true underlying brain functions (though how to interpret them is another question).

2.5 Comparing functional connectivity methods

To motivate the attention connectivity method, let's compare some selected features of the presented functional connectivity measures. A summary can be seen in Table 1.

Directed connectivity: Pearson and partial correlation are both undirected. The connectivity matrices are symmetric and thus provide no causal information, e.g., about whether region a is more strongly influenced by b than a is by region b. Granger causality (using a VAR), DICE, DSAM, and attention connectivity all provide some kind of directed information, though whether this is indeed causal is a question for future work.

Not necessarily task specific: All methods except DICE and DSAM support functional connectivity estimates from fMRI data without further labels. DICE and DSAM need to train on task specific targets, such as autism classification or age regression, to build their connectivity matrices.

Works without training: Pearson and partial correlation both don't need to train any models to work. Granger causality and attention connectivity need to be trained/fitted autoregressively to provided fMRI data. As noted above, DICE and DSAM need to be trained in a supervised manner on a combination of fMRI input data and classification/ regression labels.

Future time step prediction: Granger causality and attention connectivity do next time step prediction, which potentially requires them to model certain aspects of how the brain works over time. It is left for future work to explore how much this hypothesis turns out to be true, or which factors external to brain function (e.g., spatially and temporally heterogeneous haemodynamic response) are the main influences for future time step predictions (Smith et al., 2012). DICE and DSAM do non-temporal, task-based predictions, and Pearson and partial correlation don't do any predictions.

Non-linear modeling: Non-linear models far exceed the representation power of linear models. It is very likely that the brain exhibits patterns that are non-linear, thus it might be of use for the FC method to be able to model non-linear relationships Of the functional connectivity models that do prediction tasks, DICE, DSAM, and attention connectivity are able to do so due to their non-linear activation functions (e.g. ReLU, sigmoid). The most commonly used Granger causality model is based on VARs, i.e., a linear combination of the input time series.

Can be used dynamically: The change of functional connectivity over finer time-scales than whole sessions has been an area of interest (Allen et al., 2012; Romanello et al., 2022). Compared to looking at the aggregated connectivity over a whole session, dynamic connectivity might reveal information regarding the frequency and nature of short-term brain states. Pearson correlation and partial correlation can be easily adjusted for dynamic use by applying them on smaller windows. Attention connectivity inherently supports dynamic attention, and thus also dynamic directed connectivity. Granger causality VARs only support dynamic connectivity insofar as needs to be enough training data in each window to avoid overfitting. For DICE and DSAM it is not clear if non-trivial changes would make them available for dynamic analysis.

Works on subject-level: One goal of functional connectivity analysis is to provide individualistic insights that might help patients. In that context it is useful for an FC method to be applicable not only the group level (i.e., providing a group average connectivity), but also on just one individual. All methods allow subject level analysis (at least if I correctly understood how DICE and DSAM extract their attention matrices, which I am not 100% sure of).

Can detect anti-correlations: One interesting aspect that came out of functional connectivity research is that certain large-scale networks are anti-correlated (Fox et al., 2005). Of the methods compared here, only the correlation based methods (Pearson and partial) allow for the detection of anti-correlated signals by looking at negative correlation values (by definition). The other methods model how *important* connections are, and since anti-correlations have an equal information as an equivalent correlated signal, both signals would be deemed equally important.

Criterion	Pearson correlation	Partial correlation	Granger causality	**Attention connectivity**	DICE/DSAM
Directed connectivity	No	No	Yes	**Yes**	Yes
Not necessarily task specific	Yes	Yes	Yes	**Yes**	No
Works without training	Yes	Yes	No	**No**	No
Future time step prediction	No	No	Yes	**Yes**	No
Non-linear modeling	No	No	No	**Yes**	Yes
Can be used dynamically	Yes	Yes	Maybe	**Yes**	No
Works on subject-level	Yes	Yes	Yes	**Yes**	Yes
Detect anti-correlations	Yes	Yes	No	**No**	No

Table 1: Comparison of functional connectivity analysis methods across selected criteria

3 Method

3.1 Data

The attention connectivity model is trained and evaluated on resting state data from the *HCP-Young Adult 2025 Release* (Van Essen et al., 2013). Per subject, four runs are captured with 1200 volumes each, using a 3 T scanner with a *TR* of 0.72 s and a resolution of 2 mm isotropic. The preprocessed, surface-aligned version of the data is used, for which the HCP 2025 rfMRI preprocessing pipeline had been applied. The four runs per subject are concatenated. Additionally, the retest rfMRI dataset for 45 of the HCP S1200 subjects is used for fingerprinting analysis. Next, the data is parcellated using the *Schaefer 2018* atlas with 100 regions (Schaefer et al., 2017). For that, surface voxels falling into a single region are averaged. Compared to previous work (such as (Wein et al., 2021) or (Sun et al., 2025)), the resulting data is not not filtered to exclude higher frequencies, since it is not clear that higher frequencies include only noise (Chen & Glover, 2015). Effects of signal filtering are left for future work.

3.2 Model architecture

A summary of the architecture is shown in Figure 1. The model takes as input a short window of the signal for each region (in the order of 10 steps, at an *TR* of 0.72 s that would amount to roughly 7 seconds). Using an input embedding block consisting of a series of multi-layer perceptrons (MLPs, sometimes also referred to as feedforward or fully connected neural networks), for each region, a key, value, and query vector are generated. Using these, each region attends to each other region (i.e., computes importance), and aggregates the values from all other regions depending on this importance. Then the per-region aggregated values are again put through MLPs to finally produce a single value, which is the prediction for the BOLD value of the given region in the timestep following directly the input window.

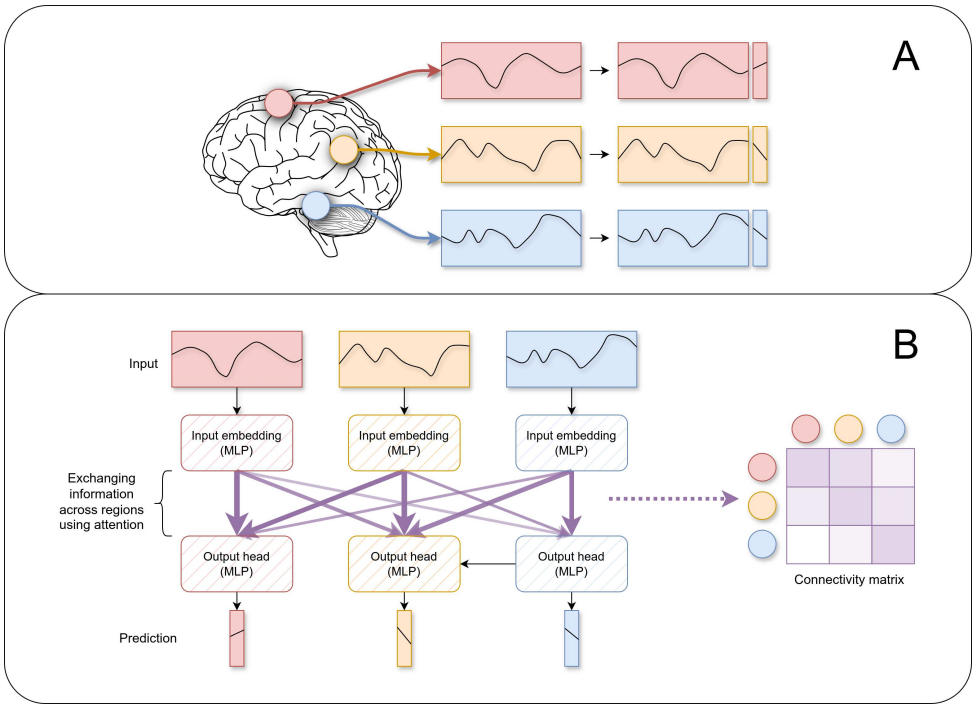

Figure 1:
A: The fMRI signal of various brain regions is split into windows, and for each window the next time step is the prediction target.
B: The signals are per-region embedded into vector representations using simple MLPs. Then, via a custom attention mechanism, the regions can share info about themselves with each other. Finally, the aggregated information is used to predict the next time step for each region. The information exchange weights are extracted and interpreted as a connectivity matrix.

3.2.1 Details

A custom attention mechanism, differing from the commonly used one (Vaswani et al., 2017), is used here. The inputs to our custom attention per region a are the vector representing the windowed BOLD signal for this region. These BOLD signals are used to generate region-to-region specific queries $Q_{ab}, a, b \in \text{regions}$ and region specific keys and values K_a, V_a (note that here BOLD_a, Q_{ab}, K_a, and V_a are all 1-dimensional vectors):

$$Q_{ab} = \text{MLP\_in}_{Q_{ab}}(\text{BOLD}_a)$$
$$K_a = \text{MLP\_in}_K(\text{BOLD}_a)$$
$$V_a = \text{MLP\_in}_V(\text{BOLD}_a)$$

$\text{MLP\_in}_{Q_{ab}}$ here being an MLP with weights (trainable parameters) that are specific to both the region asking the query and the region receiving the query. The reason for this is to allow a region to ask more region specific questions. E.g., what a brain region is interested in depends on if it is asking the visual cortex or the motor cortex. MLP_in_K and MLP_in_V share their weights for all source and target regions. The reason for not specializing these is to avoid having, for example, the specialized MLP_in_{V_a} learn a very useful function early during the training, which would then result in many regions preferably attending this region, even though all things being equal, other regions would be more useful for certain regions.

The attention score A_{ab} for region a attending region b is then computed like this:

$$A_{ab} = \max\left(0, Q_{ab} K_b^T\right)$$

The clamping to zero is there to make it easier to interpret attention scores. The attention score matrix A is built from the entries A_{ab} and row normalized to sum to one. This is the matrix which will later be extracted to build a connectivity matrix. Since the scores depend on the input BOLD values, the attention matrix is dynamic, even during inference after having finished training. After the attention score matrix has been built, it is used to aggregate the values:

$$\text{aggregated\_values}_a = \sum_{b \in \text{regions}} (V_b A_{ab})$$

Finally, to get the predicted BOLD value for region a, the aggregated values are put through a region specific MLP:

$$\text{predicted}_a = \text{MLP\_out}_a(\text{aggregated\_values}_a)$$

The Python code for this model architecture can be found on GitHub[2] or in the appendix (Section 8.1).

3.2.2 Hyperparameters

There are various hyperparameters of this model architecture. Luckily, the model seems to be mostly robust to changes to them, as later described in Section 5.3.

Input window size: This specifies how large the window of the BOLD signal should be. Here, I focus on short windows of less than 10 seconds; default configs are set to use an input window size of 5 (3.6 seconds).

Hidden dimension: This is the dimension which is used for all hidden MLP layers, including key and query vectors. It is set to 64 for group-level and subject-level models. There is one important exception where this hidden dimension is not used, which is the value vector, see the next point.

Bottleneck dimension: The value vector is bottlenecked to a lower dimension than other hidden layers, usually set to 8. The reasoning behind doing so is largely a precaution: Theoretically, the model might learn to encode all information of all 100 regions (of the Schaefer 2018 atlas) into a sufficiently large aggregated value vector. Since the model wouldn't have the pressure to select only the most valuable regions, this would make the attention score matrix much less useful. So to avoid this, the value vector is projected to a low dimension just before aggregation, and then after the weighted sum, the aggregated value is up-projected to the general hidden dimension again.

Number of layers: The input (before the attention) and output (after the attention) MLPs can be configured to have different number of layers. 4 input layers and 4 output layers are used for the subject-level model, for the group-level model, 3 layers are used for each.

Attention dropout: Optionally, a dropout function is used on the attention score matrix before applying the matrix on the value aggregation (Srivastava et al., 2014). The intent behind a higher dropout probability here is to nudge the network to learn not to rely on the most prominent sources of information, but also learn to pay attention to regions which might provide duplicate information or information of lower importance. Empirically a high dropout (0.5 or even 0.9) makes a big difference in the structure of the connectivity matrices. Though, best performance (R^2) is achieved with the dropout probability set to zero. For subject-level models, a dropout of zero is used; for group-level models, a dropout of 0.9 is used.

[2] https://github.com/tsoj/fmri_attention_connectivity

3.3 Training

There are two ways to train the model: Group-level and subject-level. For the group-level training, the training data consists of a training split (10% of subjects are excluded from training and left for testing, this includes the 45 test-retest subjects) of the preprocessed data, which amounts to a large number of subjects (~900). The resulting model can be interpreted as being a model that learned to predict brain patterns that are common across many people. For the subject-level training, the model is trained from scratch (i.e. no pre-training) exclusively on fMRI data from a single subject. Here, the train-test split is made by using the first 90% of the fMRI time series of the subject as training data, and the last 10% as test data. The time series are z-normalized per region using the mean and standard deviation computed from exclusively the training data.

The optimization method used is a variant of stochastic gradient descent using the AdamW algorithm (Loshchilov & Hutter, 2019) with a learning rate of 0.001 and a batch size of 256. For the group-level training, the model is trained for 1 epoch since training losses are showing diminishing returns already at the end of just one epoch. For the subject-level training, the model is trained for 10 epochs. This number has been chosen after unsystematic observations on the behavior of one subject (this subject not being part of the 45 test-retest subjects later used for fingerprinting evaluation). For the loss function, the mean-squared error was used.

The training is done using a single AMD RX 7900 XT GPU and usually takes less than an hour for the group-level training.

3.4 Extracting connectivity

After the model is trained, to extract connectivity information, we need to look at the attention score matrix A. To do that, we run the model on the given fMRI data set (e.g., of a specific subject we want to know connectivity for). At each time step of the fMRI time series, given the input window (of roughly size 5 to 15), the model will try to predict the next time step value for each region. However, we will ignore the prediction and instead take the attention score matrix A that was computed with this input. Since A depends on the input (i.e., depending on the input values for a specific region (e.g. progressing from high BOLD activation to low BOLD activation), that region might choose to attend other regions, than if the input value would look different (e.g. progressing from low to high BOLD activation)), this attention score matrix will look different at each timestep. To get the summary connectivity information for the entire fMRI time series, the attention score matrices of all time steps are averaged. How these connectivity matrices look like, can be seen in Figure 2.

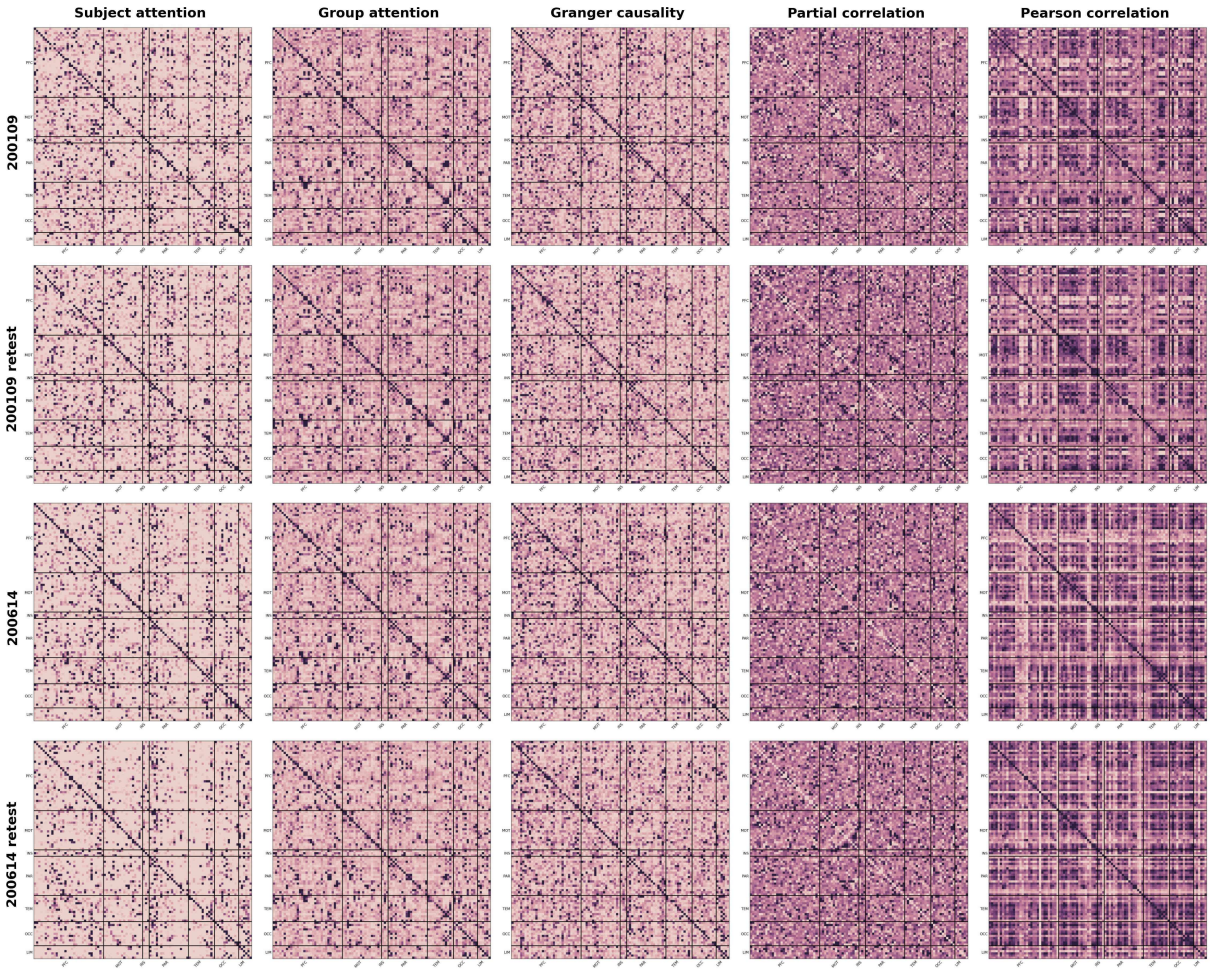

Figure 2: A visual comparison of connectivity matrices over multiple methods and two subjects (200109 and 200614) and their retest sessions. For subject attention, no dropout is used, for group attention, a dropout of 0.9 is used. For a comparison to a group-level model without dropout, see Figure 10. For the directed methods, the rows are the targets (the regions to be predicted), and the columns are the sources (the regions providing information). For example, if a single column were very connected, that would mean that the region corresponding to this column is used by every other region to predict next time steps.

4 Evaluation

To evaluate the new method, two benchmarks commonly used to test FC measure are being run: Fingerprinting and prediction of behavioral and brain structural properties. Attention connectivity is being compared to Pearson correlation, partial correlation, and Granger causality. Both for fingerprinting and behaviour prediction, it has been tested that the accuracy is not better than chance if the labels (subject IDs or behavioral targets) are randomly shuffled.

4.1 Granger causality implementation

For Granger causality, to determine the causal strength of region a, a VAR with an input window of 2 (larger windows lead to overfitting) is fitted using ridge regression to a subject's fMRI data with all regions as input and the next time step of a as target. Then, for each region b, a new VAR is fitted to predict a, except this time, region b is not part of the input. The difference in predictive performance between the full VAR to predict a, and the b ablated VAR to predict a is the causal influence of b on a. The multiplier for the

L2 term in the ridge regression model and the window size are optimized using an inner train-test split of the full training data.

4.2 Fingerprinting

For estimating fingerprinting performance, the recipe of (Finn et al., 2015) is largely followed. The HCP test-retest dataset of 45 subjects is used. These 45 subjects (of which many are twins or related) took the standard two fMRI sessions and additionally another two sessions some time later. The problem statement of fingerprinting now is, given sessions for a single subject from the test dataset, select the sessions from the retest dataset belonging to the same subject. To do that, the two sessions for each subject in test and retest datasets are concatenated into a single "big session". Then, for each big session, a connectivity matrix is calculated using the method that we want to benchmark (e.g., Granger causality, Pearson correlation, ...). The connectivity matrices are then edge wise z-normalized over the test dataset of 45 subjects (and same approach for the retest dataset). To compare the similarity between two big sessions, the Pearson correlation between the two respective connectivity matrices is computed. The performance of a method now can be measured either by using accuracy (for how many subjects is the similarity score between the two same-subject big sessions bigger than the similarity of all other possible cross-subject pairs that include this subject), or by the effect size using Cohen's d of the same-subject vs cross-subject similarity score distributions.

4.2.1 Fingerprinting results

Partial correlation shows the best fingerprinting performance (see Figure 3), identifying 100% of subjects, followed by Granger causality, which identifes 86% correctly, and then the subject-level attention, which identifies 84% of subjects correctly. The lowest performance is shown by Pearson correlation and other variants of the attention connectivity, with an accuracy of just over 70%. Notably, the fingerprinting accuracy is still reasonably high for group-level attention, which hasn't been trained on any of the subjects in the fingerprinting dataset. And importantly, the fingerprinting accuracy is also surprisingly good when computing the attention connectivity matrices for the test set with the subject-level model, and the retest dataset with the group-level model, i.e., computing the similarity between subject and group-level connectivity for the same subjects.

The effect sizes are all very high and are in line with the identification accuracy results. In Figure 4 the distributions for cross-subject and same-subject similarity scores can be seen. Partial correlation shows the lowest overlap between cross-subject and same-subject distributions. Pearson correlation has a rather large variance in its cross-subject distribution. The subject-level attention connectivity seems to show a comparatively long tail towards high similarity scores for same-subject similarity scores.

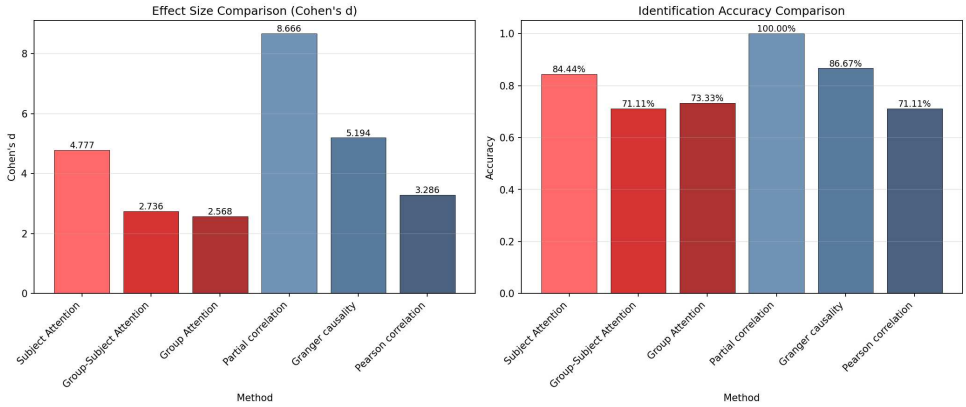

Figure 3: Comparing the fingerprinting performance for FC methods. Subject-level attention and Granger causality are very close, maybe because of the similarities in how they work (trained/fitted to predict a single subject). Results for when requiring identification from the combined set of retest and test sessions can be found in the appendix (Section 8.3).

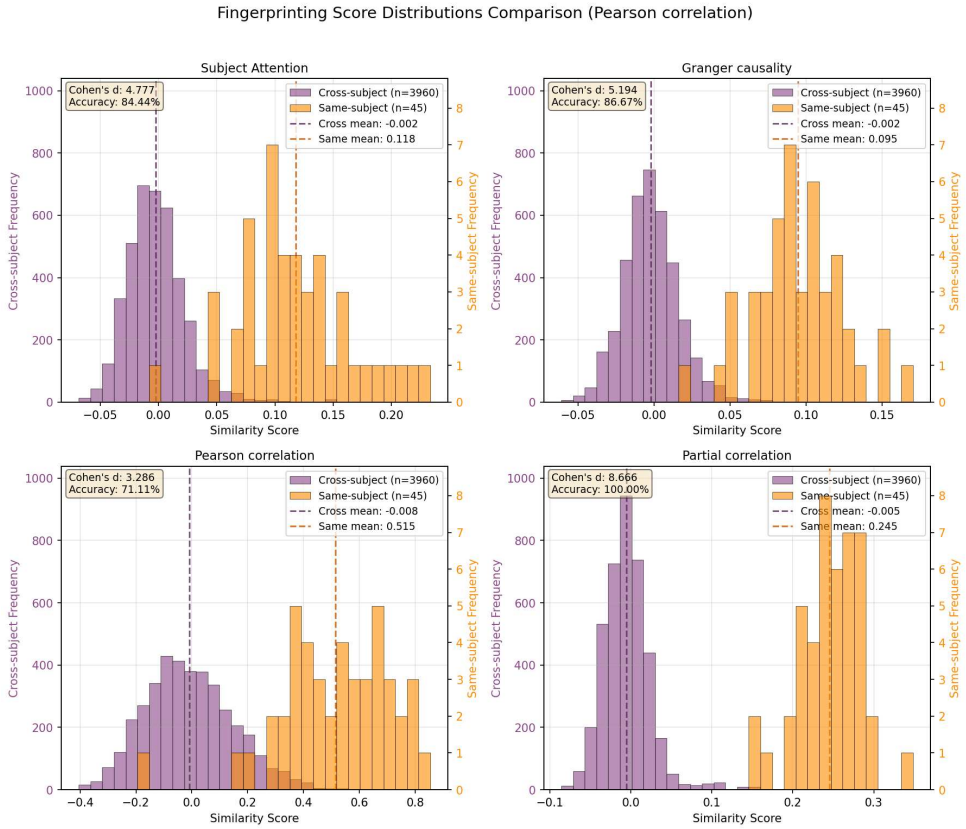

Figure 4: All fingerprinting methods show a clear separation between cross-subject and same-subject similarity distributions.

4.3 Prediction

Prediction: We can use the connectivity matrices for subjects to train a regression model for predicting various properties/traits of a subject. To do that, the open access clinical, and behavioral data mapped to fMRI subjects of the HCP project is used. I tried getting the *connectome-based predictive modeling* (CPM) approach by (Shen et al., 2017) to work, however, because of limited time, I decided to keep it simple and train a linear regressor, which is something I am more familiar with. Instead of ridge regression, the approach to limit overfitting (necessary, given the limited dataset and large input space (100×100

input features) was to train using stochastic gradient descent with early stopping based on a validation set (which was distinct from the test set). Train, validation, and test split are selected using 10-fold cross-validation, where 8 folds are used for training, 1 fold for validation (early stopping), and 1 fold for testing. Unfortunately, there was not a lot of time left to check the various details of this method, or compare it to a proper implementation of above-mentioned CPM.

4.4 Prediction results

Prediction power of the different connectivity measures differs a lot over the different possible targets (Figure 5). The method that most frequently performs best at predicting cognitive and behavioral traits is partial correlation: It has the highest R^2 score for 39% of targets. Subject-level attention predicts best for 22% and group-level for 19% of behavioral targets. Taken together, the attention based methods are better than even partial correlation, predicting 42% of cases best. Notably, in most cases, partial and group-level attention have very similar R^2 scores. However, for many targets where subject-level attention predicts very well, other methods are close to having no predictive power whatsoever. The connectivity methods are additionally tested by how well they can be used to predict various brain structural variables. Here group-level attention is the most performant method (Figure 6).

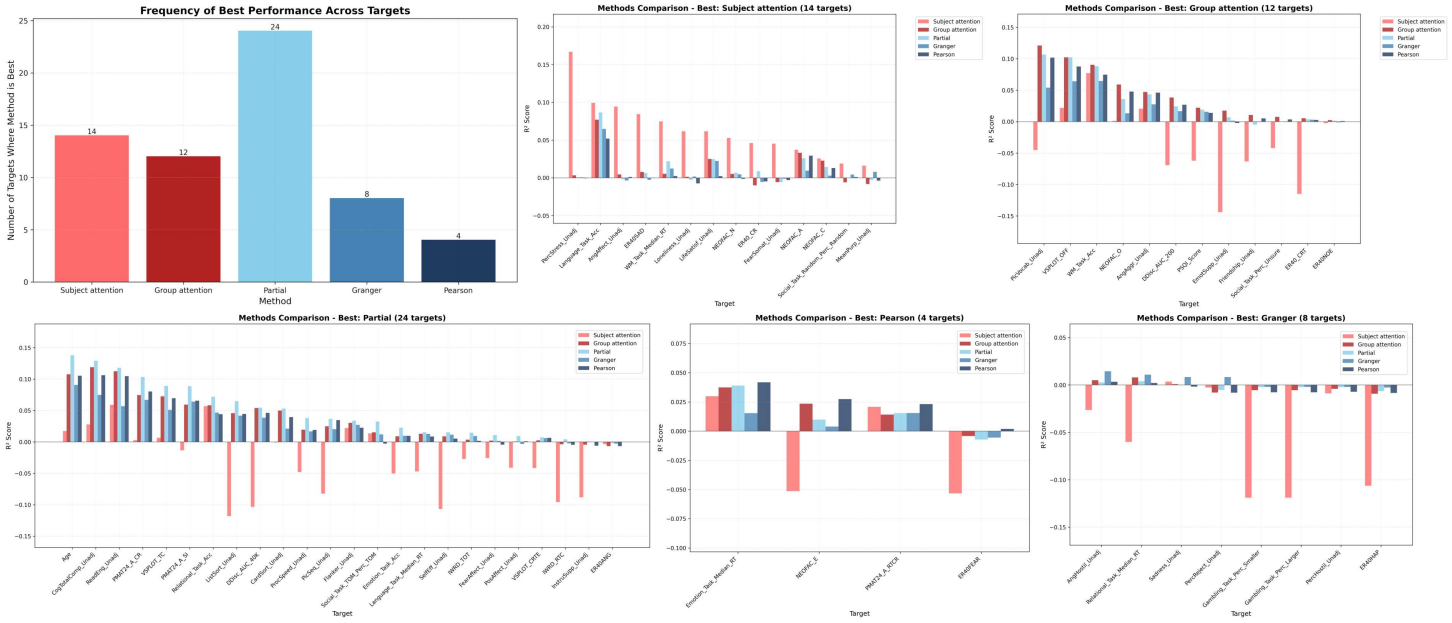

Figure 5: Comparison of prediction performance between different FC methods. Partial correlation and group-level attention have similar scores across most targets, though partial correlation is usually is slightly better. Subject-level attention excels at targets where other methods have large difficulties, such as *PercStress_Unadj* or *AngAffect_Unadj*. Though in general, the R^2 scores are rather low for all targets across all methods.

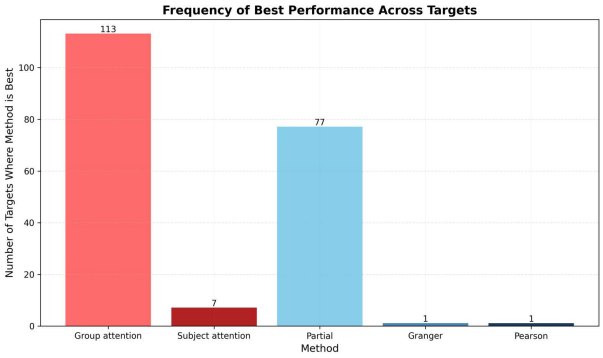

Figure 6: Prediction of brain structural variables. These include FreeSurfer statistics, volume segmentation, surface thickness, and surface area. For a lack of anatomical knowledge, some of these variables might not be part of the cerebral cortex, and thus somewhat outside the scope of the cortical surface connectivity. The R^2 scores for both group-level attention and partial correlation are rather uniformly distributed from 0 to 0.4. Detailed scores can be seen in the appendix (Section 8.4).

5 Further results

5.1 Future time step prediction

Both Granger causality and attention connectivity are trained to predict future time steps. For the extraction of connectivity information, the prediction performance is of second priority. However, as a sanity check, comparing the R^2 scores between these two methods makes sense. The expectation is that the attention model outperforms the VAR Granger causality model, since it can model non-linear relationships. However, it turns out that the attention model is only very slightly better in the mean than the Granger causality model. In Figure 7 an interesting phenomenon can be observed: If a subject is predictable, i.e., both Granger causality and attention model have high prediction accuracy, the attention model consistently outperforms the Granger model. However, if a subject is hard to predict, then the Granger causality model is usually better than the attention model. The reasons for this behaviour are not entirely clear, but closer investigation is left for future work.

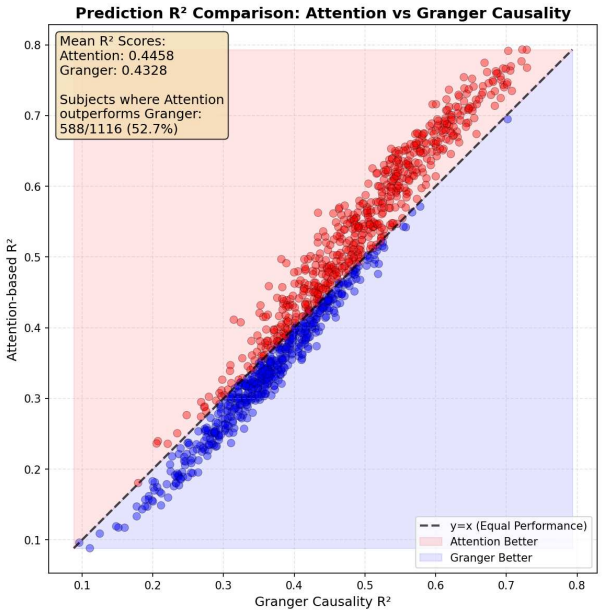

Figure 7: Each scatter point represents a subject, the place where it falls in the plot describes the achieved R^2 for the attention model on the y-axis and for Granger causality on the x-axis. If points fall on the diagonal, that means that both Granger causality and attention model are equally good at predicting this subject. If the point falls below the diagonal into the blue shaded region, then Granger causality predicts this subject better, and vice versa, if the point falls into the red shaded region, the attention model performs better. Noticeable is the trend that low-predictable subjects (low R^2 scores for either method) are better predicted by Granger causality, while for more predictable subjects, attention performs better.

Examples of how input, target, and prediction look like for an attention model, see Section 8.2.

5.2 Simulation with ground truth connectivity

To make sure that the attention model is at least in principle capable of detecting ground truth connectivity, a very simple synthetic data scheme has been tested.

The input to the synthetic data generation is a connectivity matrix A_{in}, and an input window size l. For each region pair (a, b), a random two-layer MLP_{ab} is created. This MLP takes two vectors of length l as input and produces a single value as output. Then for each training sample, a random input vector input_a of length l for each region is generated. Now, the prediction target for region a is computed the following way:

$$\text{target}_a = \sum_{b \in \text{regions}} \text{MLP}_{ab}(\text{input}_a, \text{input}_b) \cdot A_{\text{in}_{ab}}$$

Since the ground truth connectivity depends both on the random weights of MLP_{ab} and $A_{\text{in}_{ab}}$, the ground truth connectivity matrix is built by sampling a large number of random inputs and observing the average value of MLP_{ab} and multiplying this with $A_{\text{in}_{ab}}$. In practice, this true connectivity is usually very similar (correlation larger than 0.9) to the input matrix A_{in}.

When training the attention model on a dataset generated this way, the resulting attention connectivity (see Figure 8 for an example) has correlation values of around 0.5. However, it should be noted that this only serves as a sanity check, and not as proof that the attention connectivity model can extract ground truth connectivity in other cases (specifically fMRI).

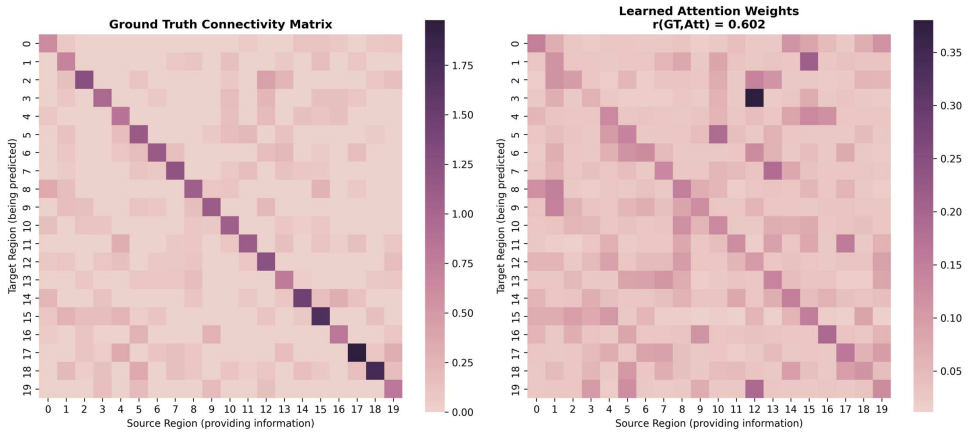

Figure 8: On the left, the ground truth connectivity of a simple simulated system. On the right, the estimated connectivity from an attention model trained on a synthetic dataset generated from said simulated system.

5.3 Hyperparameter study

In general, the attention model is rather robust regarding changes of its hyperparameters. The largest impact both on predictive performance (test loss) and on the visual appearance of the extracted connectivity matrices has the attention dropout parameter (Figure 10). Low dropout probability leads to a lower loss (i.e., better predictive performance) and a more sparse connectivity matrix. Both intuitively make sense: The higher the dropout probability, the less the model is able to focus its attention on the regions that matter the most. The reason dropout was used despite its negative impact on performance is because visually it was easier to identify differences in resulting attention connectivity matrices. However, investigating the wider effects of using or not using dropout in the attention layer is left for future work.

Noteworthy is that the performance measurably improves as the input window increases in size. This shows that the model not only uses information immediately before the next time step, but also information about the state of the brain multiple seconds before.

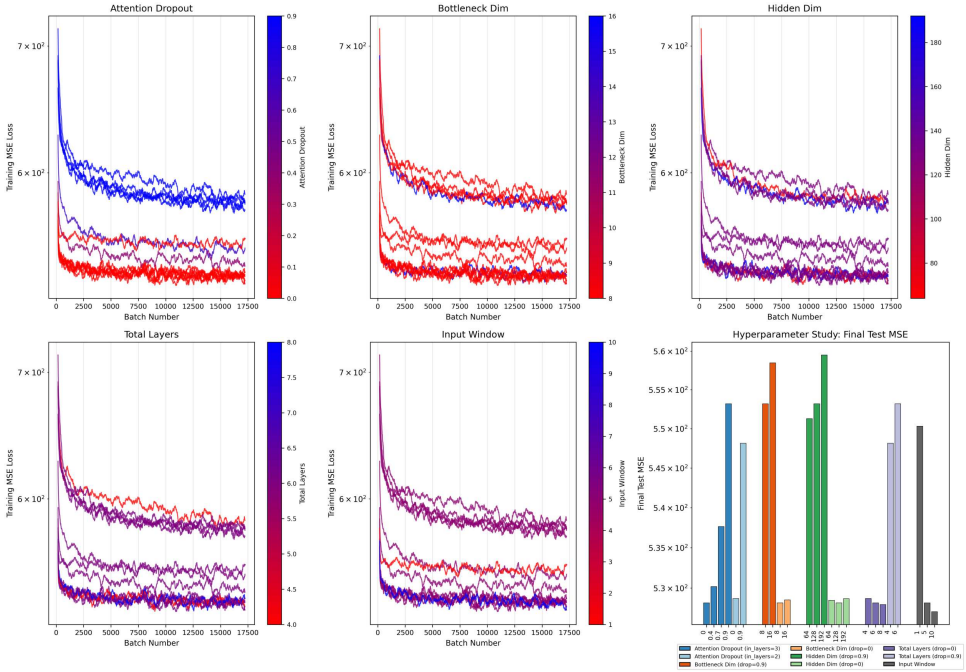

Figure 9: The bottom right plot shows the test performance as various hyperparameters are varied. The other plots show the same training loss progression of various hyperparameter configurations. The difference in the plots is after which hyperparameter the lines are shaded in blue and red.

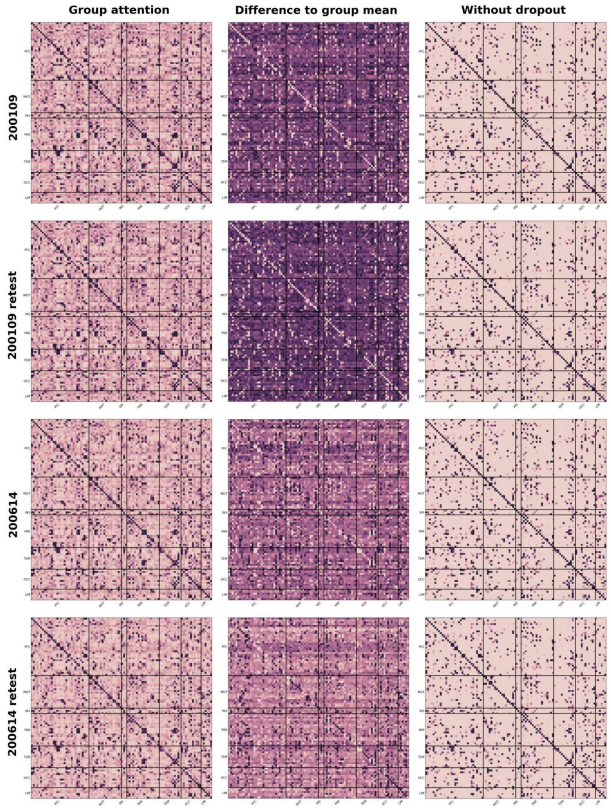

Figure 10: Comparing the visually different connectivity matrices between the default group-level model configuration (on the left), the default group-level model configuration, but visualized with the mean connectivity subtracted (middle), and a group-level model configuration without dropout (right).

6 Discussion and conclusion

The attention connectivity method, while not always beating partial correlation, shows promising results in a first set of experiments. Fingerprinting results are not outstanding, though comparable to Granger causality and improving over Pearson correlation. However, fingerprinting results show that it is likely that attention connectivity can measurably capture individual differences, even when the model architecture and training procedure change between test and retest connectivity extraction. Thus, it is likely that attention connectivity captures some of the true underlying nature of brain patterns instead of just picking up statistical outliers. This also shows in the performance on predicting behavioral and brain structural properties. While partial correlation as a single measure still outperforms attention connectivity on many prediction targets, there are a significant number of exceptions, where attention connectivity improves the prediction performance over all other compared methods.

However, there are a large number of limits to this work. First and foremost, the missing ground truth: All claims to the true causal nature of the brain are merely hypotheses that need to be tested using other methodologies. From a practical perspective, the implementations of the used benchmarks are not tested very well, and might be improved, and should be compared to other methods to estimate fingerprinting accuracy and behavioral prediction (e.g., CPM). Furthermore, Pearson correlation, partial correlation and Granger causality are only a subset of (effective) FC methods used in the field. To determine if the attention connectivity provides any added value, it might need to be compared to other existing work. In general, there are many open points left for future work: The effect of preprocessing (e.g. low-pass filtering) use of other datasets than the HCP, the effect of dropout on fingerprinting and behavioral prediction, the usefulness of the value dimension bottleneck, the influence of input window parameters (how many values over how much time), or the investigation of effects of heterogenous HRFs.

In summary, I think that this new method's advantages (primarily directedness and nonlinear modeling) could make it an interesting subject for further research.

7 Acknowledgments

Data were provided by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

Large parts of the code have been generated using LLMs such as Claude, ChatGPT, or Gemini.

Bibliography

Allen, E. A., Damaraju, E., Plis, S. M., Erhardt, E. B., Eichele, T., & Calhoun, V. D. (2012). Tracking Whole-Brain Connectivity Dynamics in the Resting State. *Cerebral Cortex*, *24*(3), 663–676. https://doi.org/10.1093/cercor/bhs352

Arab, F., Ghassami, A., Jamalabadi, H., Peters, M. A. K., & Nozari, E. (2025). Whole-brain causal discovery using fMRI. *Network Neuroscience*, *9*(1), 392–420. https://doi.org/10.1162/netn_a_00438

Azarmi, F., Miri Ashtiani, S. N., Shalbaf, A., Behnam, H., & Daliri, M. R. (2019). Granger causality analysis in combination with directed network measures for classification of MS patients and healthy controls using task-related fMRI. *Computers in Biology and Medicine*, *115*, 103495. https://doi.org/https://doi.org/10.1016/j.compbiomed.2019.103495

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. *Corr*. https://arxiv.org/abs/2005.14165

Buckner, R. L., Krienen, F. M., & Yeo, B. T. T. (2013). Opportunities and limitations of intrinsic functional connectivity MRI. *Nature Neuroscience*. https://doi.org/10.1038/nn.3423

Chen, J. E., & Glover, G. H. (2015). BOLD fractional contribution to resting-state functional connectivity above 0.1Hz. *Neuroimage*, *107*, 207–218. https://doi.org/https://doi.org/10.1016/j.neuroimage.2014.12.012

Deshpande, G., & Hu, X. (2012). Investigating Effective Brain Connectivity from fMRI Data: Past Findings and Current Issues with Reference to Granger Causality Analysis. *Brain Connectivity*, *2*(5), 235–245. https://doi.org/10.1089/brain.2012.0091

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *Corr*. https://arxiv.org/abs/2010.11929

Duggento, A., Passamonti, L., Valenza, G., Barbieri, R., Guerrisi, M., & Toschi, N. (2018). Multivariate Granger causality unveils directed parietal to prefrontal cortex connectivity during task-free MRI. *Scientific Reports*, *8*(1), 5571. https://doi.org/10.1038/s41598-018-23996-x

Finn, E. S., Shen, X., Scheinost, D., Rosenberg, M. D., Huang, J., Chun, M. M., Papademetris, X., & Constable, R. T. (2015). Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity. *Nature Neuroscience*, *18*(11), 1664–1671. https://doi.org/10.1038/nn.4135

Fox, M. D., Snyder, A. Z., Vincent, J. L., Corbetta, M., Essen, D. C. V., & Raichle, M. E. (2005). The human brain is intrinsically organized into dynamic, anticorrelated functional networks. *Proceedings of the National Academy of Sciences*, *102*(27), 9673–9678. https://doi.org/10.1073/pnas.0504136102

Greicius, M. D., Krasnow, B., Reiss, A. L., & Menon, V. (2003). Functional connectivity in the resting brain: A network analysis of the default mode hypothesis. *Proceedings*

of the National Academy of Sciences, *100*(1), 253–258. https://doi.org/10.1073/pnas.0135058100

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., … Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2

Liu, Z.-Q., Luppi, A. I., Hansen, J. Y., Tian, Y. E., Zalesky, A., Yeo, B. T. T., Fulcher, B. D., & Misic, B. (2025). Benchmarking methods for mapping functional connectivity in the brain. *Nature Methods*, *22*(7), 1593–1602. https://doi.org/10.1038/s41592-025-02704-4

Loshchilov, I., & Hutter, F. (2019,). *Decoupled Weight Decay Regularization*. https://arxiv.org/abs/1711.05101

Mahadevan, A. S., Tooley, U. A., Bertolero, M. A., Mackey, A. P., & Bassett, D. S. (2021). Evaluating the sensitivity of functional connectivity measures to motion artifact in resting-state fMRI data. *Neuroimage*, *241*, 118408. https://doi.org/https://doi.org/10.1016/j.neuroimage.2021.118408

Mahmood, U., Fu, Z., Ghosh, S., Calhoun, V., & Plis, S. (2022). Through the looking glass: Deep interpretable dynamic directed connectivity in resting fMRI. *Neuroimage*, *264*, 119737. https://doi.org/https://doi.org/10.1016/j.neuroimage.2022.119737

Mamoon, S., Xia, Z., Alfakih, A., & Lu, J. (2025). Dynamic brain effective connectivity network for identifying neurological disorders. *Applied Intelligence*, *55*(12), 850.

Marrelec, G., Krainik, A., Duffau, H., Pélégrini-Issac, M., Lehéricy, S., Doyon, J., & Benali, H. (2006). Partial correlation for functional brain interactivity investigation in functional MRI. *Neuroimage*, *32*(1), 228–237. https://doi.org/https://doi.org/10.1016/j.neuroimage.2005.12.057

Nauta, M., Bucur, D., & Seifert, C. (2019). Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, *1*(1), 19.

Novelli, L., Barnett, L., Seth, A. K., & Razi, A. (2025). Minimum-Phase Property of the Hemodynamic Response Function, and Implications for Granger Causality in fMRI. *Human Brain Mapping*, *46*(10), e70285. https://doi.org/https://doi.org/10.1002/hbm.70285

Roebroeck, A., Formisano, E., & Goebel, R. (2005). Mapping directed influence over the brain using Granger causality and fMRI. *Neuroimage*, *25*(1), 230–242. https://doi.org/https://doi.org/10.1016/j.neuroimage.2004.11.017

Romanello, A., Krohn, S., von Schwanenflug, N., Chien, C., Bellmann-Strobl, J., Ruprecht, K., Paul, F., & Finke, C. (2022). Functional connectivity dynamics reflect disability and multi-domain clinical impairment in patients with relapsing-remitting multiple sclerosis. *Neuroimage: Clinical*, *36*, 103203. https://doi.org/https://doi.org/10.1016/j.nicl.2022.103203

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2021). High-Resolution Image Synthesis with Latent Diffusion Models. *Corr*. https://arxiv.org/abs/2112.10752

Schaefer, A., Kong, R., Gordon, E. M., Laumann, T. O., Zuo, X.-N., Holmes, A. J., Eickhoff, S. B., & Yeo, B. T. T. (2017). Local-Global Parcellation of the Human Cerebral Cortex from Intrinsic Functional Connectivity MRI. *Cerebral Cortex*, *28*(9), 3095–3114. https://doi.org/10.1093/cercor/bhx179

Shen, X., Finn, E. S., Scheinost, D., Rosenberg, M. D., Chun, M. M., Papademetris, X., & Constable, R. T. (2017). Using connectome-based predictive modeling to predict individual behavior from brain connectivity. *Nature Protocols*. https://doi.org/10.1038/nprot.2016.178

Shen, X., Tokoglu, F., Papademetris, X., & Constable, R. (2013). Groupwise whole-brain parcellation from resting-state fMRI data for network node identification. *Neuroimage*, *82*, 403–415. https://doi.org/https://doi.org/10.1016/j.neuroimage.2013.05.081

Smith, S., Bandettini, P., Miller, K., Behrens, T., Friston, K., David, O., Liu, T., Woolrich, M., & Nichols, T. (2012). The danger of systematic bias in group-level FMRI-lag-based causality estimation. *Neuroimage*, *59*(2), 1228–1229. https://doi.org/https://doi.org/10.1016/j.neuroimage.2011.08.015

Sobczak, F., He, Y., Sejnowski, T. J., & Yu, X. (2020). Predicting the fMRI Signal Fluctuation with Recurrent Neural Networks Trained on Vascular Network Dynamics. *Cerebral Cortex*, *31*(2), 826–844. https://doi.org/10.1093/cercor/bhaa260

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, *15*(1), 1929–1958.

Stephan, K. E., & Roebroeck, A. (2012). A short history of causal modeling of fMRI data. *Neuroimage*, *62*(2), 856–863. https://doi.org/https://doi.org/10.1016/j.neuroimage.2012.01.034

Sun, Y., Cabezas, M., Lee, J., Wang, C., Zhang, W., Calamante, F., & Lv, J. (2024,). *Predicting Human Brain States with Transformer*. https://arxiv.org/abs/2412.19814

Sun, Y., Chahine, D., Wen, Q., Liu, T., Li, X., Yuan, Y., Calamante, F., & Lv, J. (2025,). *Voxel-Level Brain States Prediction Using Swin Transformer*. https://arxiv.org/abs/2506.11455

Thapaliya, B., Miller, R., Chen, J., Wang, Y. P., Akbas, E., Sapkota, R., Ray, B., Suresh, P., Ghimire, S., Calhoun, V. D., & Liu, J. (2025). DSAM: A deep learning framework for analyzing temporal and spatial dynamics in brain networks. *Medical Image Analysis*, *101*, 103462. https://doi.org/https://doi.org/10.1016/j.media.2025.103462

Thomas Yeo, B. T., Krienen, F. M., Sepulcre, J., Sabuncu, M. R., Lashkari, D., Hollinshead, M., Roffman, J. L., Smoller, J. W., Zöllei, L., Polimeni, J. R., Fischl, B., Liu, H., & Buckner, R. L. (2011). The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of Neurophysiology*, *106*(3), 1125–1165. https://doi.org/10.1152/jn.00338.2011

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., & Ugurbil, K. (2013). The WU-Minn Human Connectome Project: An overview. *Neuroimage*, *80*, 62–79. https://doi.org/https://doi.org/10.1016/j.neuroimage.2013.05.041

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *Corr*. http://arxiv.org/abs/1706.03762

Wang, H. E., Bénar, C. G., Quilichini, P. P., Friston, K. J., Jirsa, V. K., & Bernard, C. (2014). A systematic framework for functional connectivity measures. *Frontiers in Neuroscience*. https://doi.org/10.3389/fnins.2014.00405

Wein, S., Malloni, W. M., Tomé, A. M., Frank, S. M., Henze, G.-I., Wüst, S., Greenlee, M. W., & Lang, E. W. (2021). A graph neural network framework for causal inference in brain networks. *Scientific Reports*, *11*(1), 8061. https://doi.org/10.1038/s41598-021-87411-8

Wein, S., Schüller, A., Tomé, A. M., Malloni, W. M., Greenlee, M. W., & Lang, E. W. (2022). Forecasting brain activity based on models of spatiotemporal brain dynamics: A comparison of graph neural network architectures. *Network Neuroscience*, *6*(3), 665–701. https://doi.org/10.1162/netn_a_00252

Zheng, W., Bao, C., Mu, R., Wang, J., Li, T., Zhao, Z., Yao, Z., & Hu, B. (2024). Frequency-specific dual-attention based adversarial network for blood oxygen level-dependent time series prediction. *Human Brain Mapping*, *45*(14), e70032. https://doi.org/https://doi.org/10.1002/hbm.70032

8 Appendix

8.1 The python code for the attention model

```
import numpy as np
import torch
import torch.nn as nn


class CustomAttention(nn.Module):
    """Attention mechanism with shared K/V and source-target aware queries"""

    def __init__(self, dim, bottleneck_dim, num_regions, attention_dropout_rate):
        super(CustomAttention, self).__init__()
        assert dim > 0, f"Dimension must be positive, got {dim}"
        assert bottleneck_dim > 0, f"Bottleneck dimension must be positive, got {bottleneck_dim}"
        assert bottleneck_dim ⩽ dim, f"Bottleneck dimension ({bottleneck_dim}) must be ⩽ dim ({dim})"
        assert num_regions > 0, f"Number of regions must be positive, got {num_regions}"
        assert 0 ⩽ attention_dropout_rate ⩽ 1, f"Dropout rate must be in [0,1], got {attention_dropout_rate}"

        self.num_regions = num_regions
        self.dim = dim
        self.bottleneck_dim = bottleneck_dim

        # Shared projections for keys and values
        self.shared_key_proj = nn.Linear(dim, dim)
        self.shared_value_proj = nn.Linear(dim, bottleneck_dim)

        # Source-target specific query projections as a parameter tensor
        # Shape: (num_regions_source, num_regions_target, input_dim, input_dim)
        self.query_weight = nn.Parameter(torch.randn(num_regions, num_regions, dim, dim))
        self.query_bias = nn.Parameter(torch.randn(num_regions, num_regions, dim))

        # Attention dropout for regularization
        self.attention_dropout = nn.Dropout(attention_dropout_rate)

        self.final_value_reproj = nn.Linear(bottleneck_dim, dim)

    def forward(self, q_embed, kv_embed):
        """
        Args:
            q_embed: (batch_size, num_regions, dim) - region-specific embeddings for queries
            kv_embed: (batch_size, num_regions, dim) - shared embeddings for keys and values

        Returns:
            attended_outputs: (batch_size, num_regions, dim)
            attention_weights: (batch_size, num_regions, num_regions)
        """
        batch_size, num_regions_q, dim_q = q_embed.shape
        batch_size_kv, num_regions_kv, dim_kv = kv_embed.shape

        assert batch_size == batch_size_kv, f"Batch size mismatch: q={batch_size}, kv={batch_size_kv}"
        assert num_regions_q == self.num_regions, (
            f"Query regions mismatch: expected {self.num_regions}, got {num_regions_q}"
        )
        assert num_regions_kv == self.num_regions, (
            f"Key/value regions mismatch: expected {self.num_regions}, got {num_regions_kv}"
        )
        assert dim_q == self.dim, f"Query dimension mismatch: expected {self.dim}, got {dim_q}"
        assert dim_kv == self.dim, f"Key/value dimension mismatch: expected {self.dim}, got {dim_kv}"

        # Compute shared keys and values for all regions
        K = self.shared_key_proj(kv_embed)  # (batch_size, num_regions, dim)
        V = self.shared_value_proj(kv_embed)  # (batch_size, num_regions, bottleneck_dim)

        V_normalized = nn.functional.normalize(V, dim=-1)  # Normalize to keep the effect of all values the same

        # Compute source-target specific queries vectorized
        # Q[i,j] = query from source region i to target region j
        # Using einsum for efficient computation:
        #     (batch, source, dim) * (source, target, dim, dim) → (batch, source, target, dim)
        Q = torch.einsum("bsi,stid→bstd", q_embed, self.query_weight) + self.query_bias.unsqueeze(
            0
        )  # unsqueeze to account for batch dim

        # Compute attention scores vectorized: Q[i,j] · K[j] for each source i and target j
        # Using einsum: (batch, source, target, dim) * (batch, target, dim) → (batch, source, target)
        attention_weights = torch.einsum("bstd,btd→bst", Q, K)

        # Only keep positive weights and normalize attention weights for each source region to sum to 1.0
        attention_weights = attention_weights.relu()
        attention_weights_sum = 1e-8 + attention_weights.sum(dim=-1, keepdim=True)
        attention_weights = attention_weights / attention_weights_sum

        # Apply dropout to attention to nudge model to try to use a variety of other regions
        attention_weights = self.attention_dropout(attention_weights)

        # Compute attended outputs: weighted sum of values
        attended_outputs = torch.bmm(attention_weights, V_normalized)
        attended_outputs = self.final_value_reproj(attended_outputs)
```

```
        return attended_outputs, attention_weights


class HCPAttentionModel(nn.Module):
    """Time series prediction model using symmetric attention mechanism"""

    def __init__(
        self,
        num_regions,
        input_window,
        output_window,
        hidden_dim,
        num_input_layers,
        num_prediction_layers,
        bottleneck_dim,
        attention_dropout_rate,
        mean,
        stddev,
    ):
        super(HCPAttentionModel, self).__init__()

        # Validate all input parameters
        assert num_regions > 0, f"Number of regions must be positive, got {num_regions}"
        assert input_window > 0, f"Input window must be positive, got {input_window}"
        assert output_window > 0, f"Output window must be positive, got {output_window}"
        assert hidden_dim > 0, f"Hidden dimension must be positive, got {hidden_dim}"
        assert num_input_layers > 0, f"Number of input layers must be positive, got {num_input_layers}"
        assert num_prediction_layers > 0, f"Number of prediction layers must be positive, got {num_prediction_layers}"
        assert bottleneck_dim > 0, f"Bottleneck dimension must be positive, got {bottleneck_dim}"
        assert bottleneck_dim <= hidden_dim, (
            f"Bottleneck dimension ({bottleneck_dim}) must be <= hidden dimension ({hidden_dim})"
        )
        assert 0 <= attention_dropout_rate <= 1, (
            f"Attention dropout rate must be in [0,1], got {attention_dropout_rate}"
        )
        assert len(mean) == num_regions, f"Mean array length ({len(mean)}) must match num_regions ({num_regions})"
        assert len(stddev) == num_regions, f"Stddev array length ({len(stddev)}) must match num_regions ({num_regions})"
        assert np.all(np.array(stddev) > 0), f"All standard deviations must be positive, got min: {np.min(stddev)}"

        # Store all configuration parameters as instance variables
        self.num_regions = num_regions
        self.input_window = input_window
        self.output_window = output_window
        self.hidden_dim = hidden_dim
        self.num_input_layers = num_input_layers
        self.num_prediction_layers = num_prediction_layers
        self.bottleneck_dim = bottleneck_dim
        self.attention_dropout_rate = attention_dropout_rate

        # Store normalization parameters as buffers in reshaped form for broadcasting
        # Shape: (1, num_regions, 1) for broadcasting with (batch_size, num_regions, input/output_window)
        mean_tensor = torch.tensor(mean, dtype=torch.float32).view(1, -1, 1)
        stddev_tensor = torch.tensor(stddev, dtype=torch.float32).view(1, -1, 1)
        self.register_buffer("mean", mean_tensor)
        self.register_buffer("stddev", stddev_tensor)

        # Shared embedding for keys and values (same for all regions)
        self.shared_kv_input_projection = nn.Linear(input_window, hidden_dim)
        self.shared_kv_blocks = self._build_layers(hidden_dim, num_input_layers - 1)

        # Per region input embedding layers for the query
        self.query_input_projection = nn.ModuleList([nn.Linear(input_window, hidden_dim) for _ in range(num_regions)])
        self.query_blocks = nn.ModuleList(
            [self._build_layers(hidden_dim, num_input_layers - 1) for _ in range(num_regions)]
        )

        # Per region output heads after the attention
        self.post_attention_blocks = nn.ModuleList(
            [self._build_layers(hidden_dim, num_prediction_layers - 1) for _ in range(num_regions)]
        )
        self.output_projection_layers = nn.ModuleList(
            [nn.Linear(hidden_dim, output_window) for _ in range(num_regions)]
        )

        # Custom attention module
        self.attention_module = CustomAttention(hidden_dim, bottleneck_dim, num_regions, attention_dropout_rate)

    def _build_layers(self, dim, num_layers):
        num_layers = max(0, num_layers)
        layers = []
        for i in range(num_layers):
            layers.extend([nn.ReLU(), nn.Linear(dim, dim)])

        return nn.Sequential(*layers)

    def forward(self, x, return_attention=False):
        """
        Args:
            x: (batch_size, num_regions, input_window)
            return_attention: Whether to return attention weights

        Returns:
```

```
    predictions: (batch_size, num_regions, output_window)
    attention_weights: (batch_size, num_regions, num_regions) if return_attention=True
"""
batch_size, num_regions, input_window = x.shape
assert num_regions == self.num_regions, (
    f"Input regions ({num_regions}) must match model regions ({self.num_regions})"
)
assert input_window == self.input_window, (
    f"Input window ({input_window}) must match model input window ({self.input_window})"
)
assert batch_size > 0, f"Batch size must be positive, got {batch_size}"

# Apply per-region z-normalization to input
# x shape: (batch_size, num_regions, input_window)
# mean and stddev shape: (1, num_regions, 1) - already reshaped for broadcasting
x_normalized = (x - self.mean) / (self.stddev + 1e-8)

# Initial projection and query-specific processing for all regions
query_embed = [
    self.query_blocks[i](self.query_input_projection[i](x_normalized[:, i, :])) for i in range(self.num_regions)
]
query_embed = torch.stack(query_embed, dim=1)

# Compute shared key-value embeddings (optimized batch processing)
x_reshaped = x_normalized.view(batch_size * self.num_regions, self.input_window)
kv_projected = self.shared_kv_input_projection(x_reshaped)
key_value_embed = self.shared_kv_blocks(kv_projected).view(batch_size, self.num_regions, self.hidden_dim)

# Apply attention: region-specific queries, shared keys and values
attended_outputs, attention_weights = self.attention_module(q_embed=query_embed, kv_embed=key_value_embed)

# Post-attention processing and output projection for all regions
region_pred = [
    self.output_projection_layers[i](self.post_attention_blocks[i](attended_outputs[:, i, :]))
    for i in range(self.num_regions)
]
region_pred = torch.stack(region_pred, dim=1)

# Un-normalize predictions (undo the z-normalization)
# region_pred shape: (batch_size, num_regions, output_window)
# mean and stddev shape: (1, num_regions, 1) - already reshaped for broadcasting
region_pred_unnormalized = region_pred * self.stddev + self.mean

assert list(attention_weights.shape) == [batch_size, self.num_regions, self.num_regions], (
    f"Unexpected attention weights shape: {attention_weights.shape}"
)
assert list(region_pred_unnormalized.shape) == [
    batch_size,
    self.num_regions,
    self.output_window,
], f"Unexpected final output shape: {region_pred_unnormalized.shape}"

if return_attention:
    return region_pred_unnormalized, attention_weights

return region_pred_unnormalized
```

8.2 Examples of inputs and prediction time series

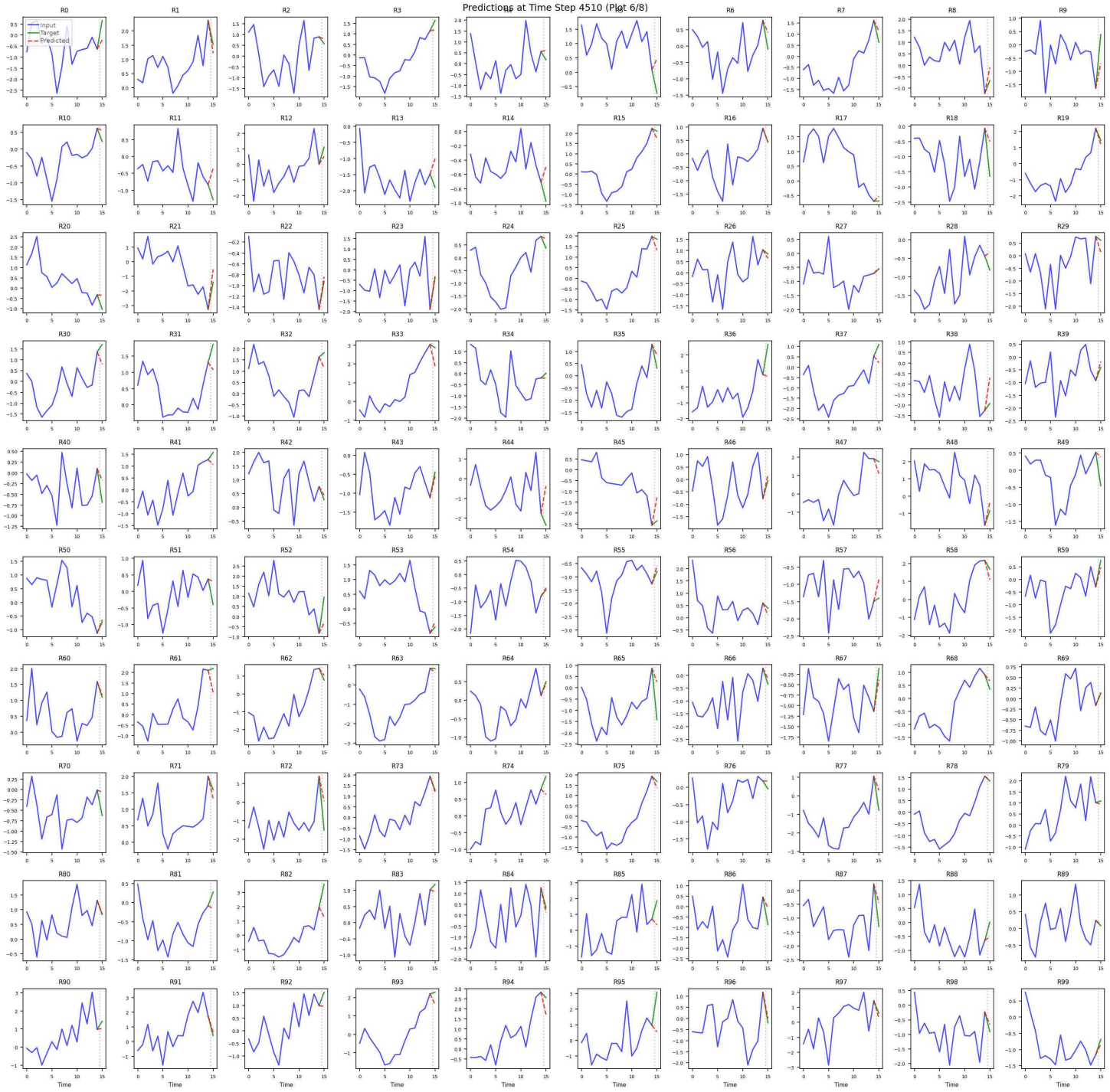

Figure 11: The input (blue), ground truth continuation (green), and predicted continuation using an attention model (red dashed). All 100 Schaefer 2018 regions shown at the same time step.

8.3 Extra fingerprinting plots

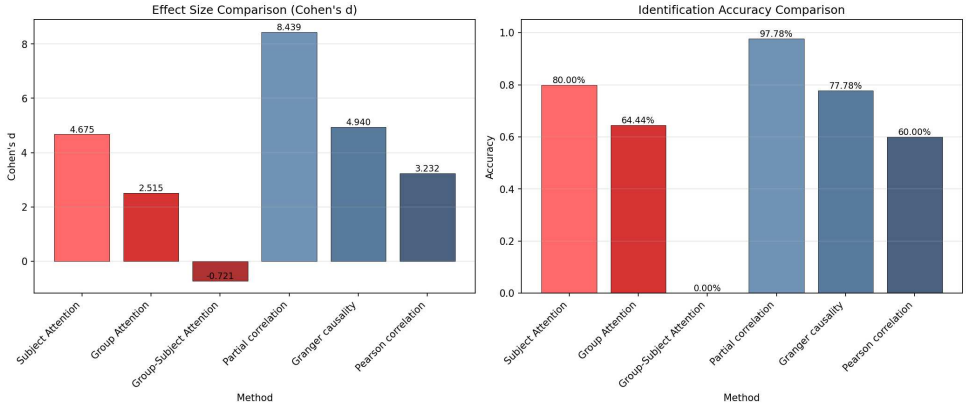

Figure 12: When we make the fingerprinting problem a little harder by using all cross subject pair even inside just the test set (or just the retest set respectively), we see very similar results, except for the group-subject version. This is because when a subject-level model is used to create the test attention connectivity matrices, and the group-level model is used to create the retest connectivity matrices, the matrices inside the test set are much more similar to each other, than compared to any (even include same-subject) matrices from the retest set which used a different model.

8.4 Extra brain structural prediction

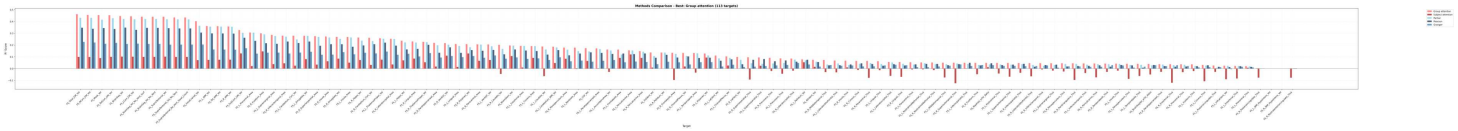

Figure 13: Brain structural properties best predicted by group-level attention connectivity.

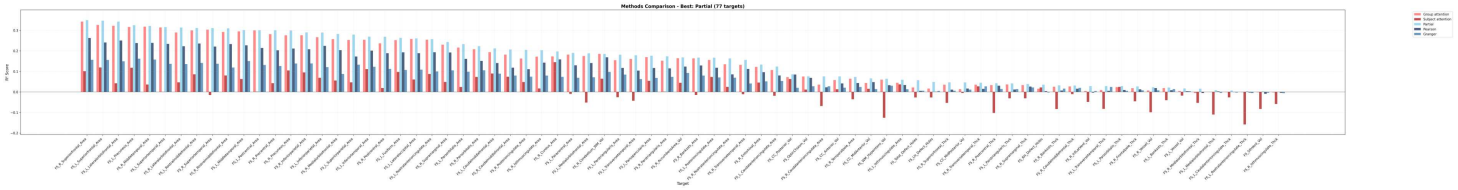

Figure 14: Brain structural properties best predicted by partial correlation.