Application of Bayesian Networks to EEG Data

Nicolas Etessami - ne47@st-andrews.ac.uk

Supervisor: Dr V. Anne Smith

External Collaborator: Dr Anastasia Klimovich-Gray

30th August 2025

University of
St Andrews

Abstract

Bayesian networks (BNs) provide a flexible way to model probabilistic directed relationships between variables but have rarely been applied to electroencephalography (EEG) data. Here, we present a new approach that uses BNs to represent neural connectivity patterns from 64-channel EEG recordings. For each recording, we construct a BN to capture directed interactions over time between channels and use the resulting edges as interpretable descriptors of neural information flow. We apply this method to EEG data taken from subjects listening to an audiobook under three different task conditions: *no task* (listening casually), *semantic* (listening carefully), and *auditory* (focusing on speaker accents). We show that the connectivity patterns derived by BNs preserve meaningful differences between brain states during these different tasks, and that machine learning classification models can successfully differentiate between these states. This work demonstrates the potential of BNs as a tool for EEG analysis and lays foundations for further development of this methodology.

1 Introduction

1.1 Bayesian networks

A **graph** is a mathematical object consisting of a set of **nodes** (also known as vertices) and a set of **edges** (also known as links) between these nodes. Graphs can be visualised as points connected by lines. A graph is known as a directed graph if its edges encode a direction between nodes — this is represented by including arrowheads on the lines.

A **Bayesian network** (BN) is a directed graph that is acyclic (without loops), known as a directed acyclic graph (DAG). In a BN, the nodes represent variables, and the directed edges represent probabilistic dependence relationships between these variables. The node to which an edge points is known as a child node of the parent node at which the edge starts (Figure 1).

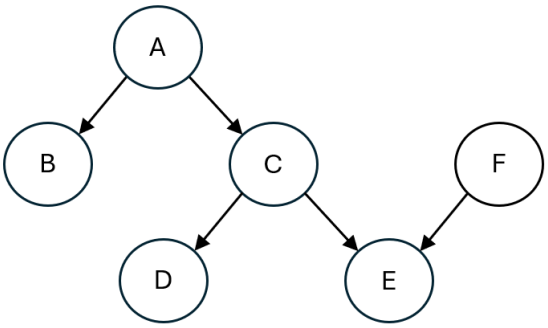

Figure 1: An example BN. Node A is a parent of B and C. Node C is a parent of D and E. Node F is a second parent of E.

A BN with nodes corresponding to the random variables $X_1, ..., X_n$ describes a particular factoring of the joint probability distribution $P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i|Pa(X_i))$ where $Pa(X_i))$ is the parents of X_i in the BN [13]; this describes the probability of the random variables taking on possible combinations of values [9].

The restriction that BNs must be acyclic can be an issue, as biological systems often contain feedback loops [7]. However, when time-series data (sets of values measured across time) is used, the model can be modified to include all variables at different time slices. This creates a **dynamic Bayesian network** (DBN). The most basic version of a DBN contains two time slices, each representing variables at two different points in time. These are time t and $t - \Delta t$, where Δt represents the time interval between the two time slices. We make the assumption that edges can only point forward in time — this allows for loops to appear over time while the underlying BN remains acyclic. Additionally, this means that edges can be interpreted as providing causal information between variables in a DBN, rather than only correlations, as in a BN (Figure 2). If we represent each electrode from the EEG data as a node in a DBN, then the edges between them represent the flow of neural information over time.

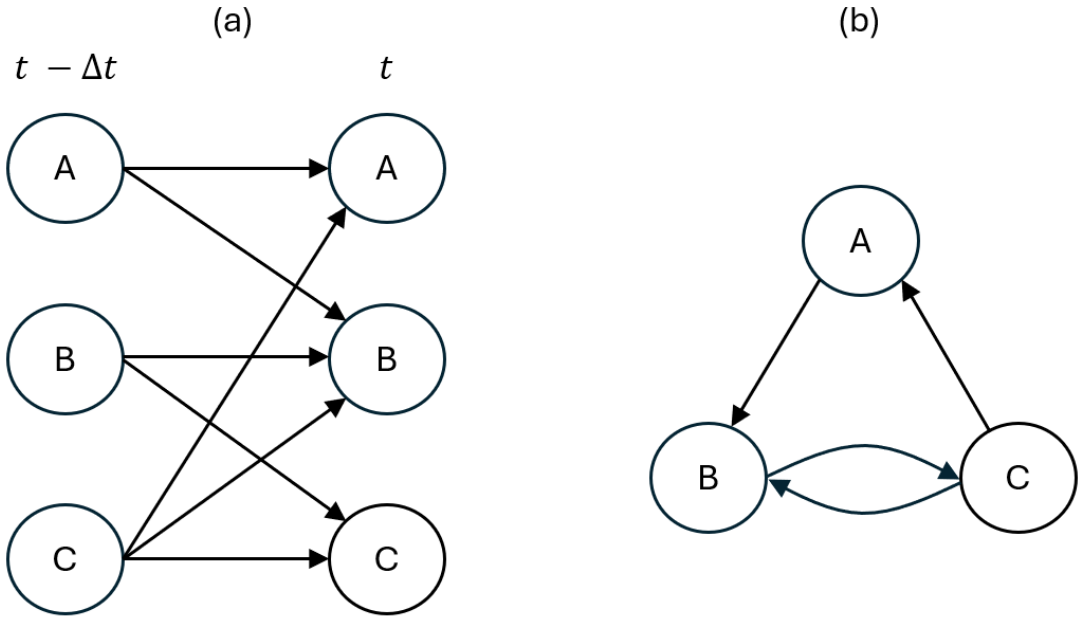

Figure 2: An example DBN. (a) Underlying acyclic BN containing 6 nodes; 1 for each variable at each point in time. (b) equivalent DBN over time containing a feedback loop between B and C.

The time interval Δt is determined by the **Markov lag**. A DBN with Markov lag of 1 uses values from 1 time sample before the time slice at time t as the time slice $t - \Delta t$. The Markov lag can be more complex, for example a Markov lag of 2 involves edges pointing 2 time slices into the future. Multiple different Markov lags can even be used in the same DBN, resulting in variables at time t being dependent on variables in multiple different time slices in the past.

There are a few methods that can be used to find the BN that best fits a set of data, however, the one that is used most in biological systems, and hence also in this work, is the search and score method [19]. This involves calculating a score for a BN given the data, which acts as a measure of how well the BN fits the data. One way of doing this is by estimating the probability of the graph given the data, $P(G|D)$. The log of this is usually taken because of the very small probabilities involved when dealing with large numbers of possible graphs [9]. Applying Bayes' rule, we obtain:

$$log(P(G|D)) = log(P(D|G)) + log(P(G)) - log(P(D)).$$

Since $P(D)$ is the same for all graphs, it can be ignored as a constant. $P(G)$ is the prior over graphs and can also be ignored when there is no reason to prefer some structures over others, so what remains is $P(D|G)$. One way of calculating this is known as the Bayesian Dirichlet equivalence (BDe) score, which has again widely been used in biological systems and in this work too. Details of the score and its derivation are omitted here, but can be found in Heckerman [10] [19].

There are too many possible graphs to simply search through each one to find the highest-scoring BN, so instead a heuristic search is used. This involves exploring the search space (space of all possible BNs) by taking steps determined by a set of rules known as the searching algorithm [9]. Provided

with sufficient searching time, this will usually find one of the highest, if not the highest, scoring BNs in the search space.

One of the more basic searching algorithms is a **greedy search**. This starts at a random location in the search space, of which the score is calculated. A single change to the BN is then made, such as the addition or removal of an edge, and the score of the resulting new BN is then calculated. Different single changes to the original BN are attempted until the score improves, at which point the algorithm steps onto the new BN with a higher score. This process repeats until no single change can improve the BN score, which means that a local peak has been found in the search space. Greedy searches can often get stuck in local peaks, so they are often run multiple times at different starting points to cover more of the search space (Figure 3a) [9].

A slightly more advanced, but useful, searching algorithm is **simulated annealing**. This is similar to a greedy search in that it takes small steps across the search space; however, steps to a lower scoring BN are possible and are taken with a probability known as the temperature. The temperature decreases over the duration of the search, meaning a final local peak is reached before the search terminates as before (Figure 3b).

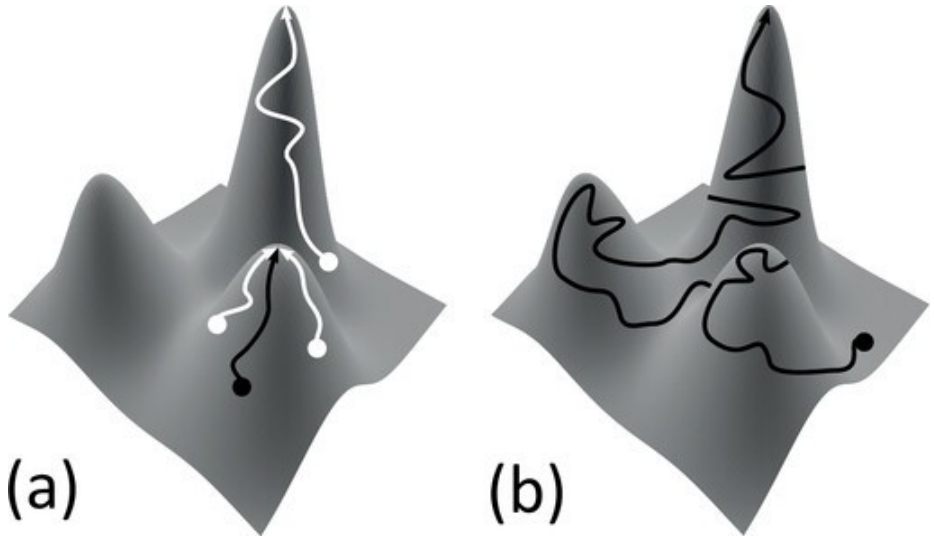

Figure 3: Visualisations of searching algorithms by Hammond and Smith [9]. The search space is simplified to a surface where height represents BN score, and adjacent points on the surface represents networks that are separated by a single change. (a) A greedy search. The search is performed multiple times to cover more of the search space. (b) Simulated annealing. Steps downward become less common as the search continues and the temperature decreases.

1.2 EEG

BNs have been used in many areas of biology over the last few decades. Common applications have been in molecular biology [22], ecology [14], and also neuroscience [26]. In neuroscience, DBNs can be used to infer neural information flow between brain regions by learning DBNs from neural activity data [9]. While many mathematical models are capable of this, most of them assume that there is a linear relationship between the neural activity levels in different regions, yet these relationships

are known to be non-linear [15]. However, DBNs are able to model such non-linear temporal relationships, making them a good choice to explore further in this context [15]. Another advantage of DBNs is that they also capture the directionality of the neural flow. Most models used in the related literature do not have such a directionality feature while also being compatible a large space of possibilities.

There are a variety of ways to source neural activity data for this task. DBNs have been shown to be useful when combined with electrophysiological data [21], and have also had some success with fMRI data [17]. However, the latter has a weakness because it has a very low temporal resolution, which means that a lot of rapid neural activity would not be captured by the DBNs.

Electroencephalography (EEG) is a technique used to record electrical activity in the brain using electrodes placed on the scalp. It is a powerful low-cost tool with high temporal resolution, which means that it can capture very fast neural activity, unlike fMRI [3]. Although there is some literature on the combination of BNs with EEG data [15], the topic has remained under-explored, and potential benefits are still largely unclear [9].

This work attempts to explore further whether BNs can be useful in helping to make inferences about neural flow from EEG data. EEG data from 33 human subjects was collected while the subjects listened to roughly 5 minute-long audiobooks about Greek mythology. The subjects each did this three times and each time were given a different task, each expected to generate different neural connectivity patterns:

- No task — subjects were told to relax and listen casually (this serves as a baseline).
- Semantic task — subjects were told to listen carefully and prepare to answer questions on the contents of the audiobook afterwards.
- Auditory task — subjects were told to pay attention to the accent of the speaker and how they pronounce words. The speaker changes during the audiobook for this task.

The subjects always performed no task first, and the subsequent order of the semantic and auditory tasks was randomised for each.

If DBNs are effective in learning meaningful structures from EEG data, the learned DBNs would show consistent task-specific patterns across subjects, and we would be able to determine which of the three tasks a subject was doing from their learned DBN with a relatively high accuracy. This is what this work achieves.

2 Methodology

2.1 Setup

The 64-channel EEG data, collected for this work with a sampling rate of 512 Hz, was pre-processed to reduce noise. For each subject and task, the EEG data was *re-referenced* to the average electrode. This can remove electromagnetic noise contributed by the surrounding environment and is effective if this affects each electrode similarly [16]. *Independent components analysis* (ICA) was then performed to identify and remove, based on correlation with the ocular electrodes, the underlying signals related to blink artifacts. Any channels found to still be noisy after this were further manually inspected

and interpolated where necessary. Finally, the data was *band-pass filtered* between 0.2–40 Hz with a zero phase FIR filter. This was done using the MNE library in Python [8]. Of the 33 subjects, 6 experienced various data collection issues; therefore, only the data of 27 subjects was ultimately used.

The BANJO software application, provided by Duke University, was used to search for the highest scoring BNs [21][24]. It is one of the few BN search frameworks that can search for DBNs and also has many customisable search parameters, making it a good choice for this work.

The initial steps in this project involved fine-tuning the setting parameters for the searches. It was immediately apparent that simulated annealing found consistently higher scoring networks than a greedy search, suggesting that the search spaces are complex and have multiple local peaks; hence, it was decided to use the simulated annealing searching algorithm for the remainder of the work. It was also decided to downsample the data to a sampling rate of 200 Hz, from the original 512 Hz. This was done because with 64 nodes, the limited computing power this project had access to struggled with the amount of data provided with 512 Hz, whereas this was not an issue at 200 Hz. It is also known that 200 Hz is still a high enough sampling rate to retain important information [4].

The discretisation level refers to how many different levels (e.g. high, medium, low) the EEG data is split into. BANJO requires discrete data as input, so discretising data before searches is an essential process. It was attempted to split data into 3, 4 and 5 quantiles and learn DBNs from this. As expected, the fewer levels the data was split into, the more edges were present in the top networks [18]. Splitting data into 5 quantiles (call this q5) resulted in top networks that were too sparse. Both q3 and q4 resulted in networks with a reasonable number of edges in their top networks; however, q3 also sacrifices more information by having fewer levels, so it was decided to proceed with the q4 discretisation policy (splitting data into 4 quantiles). This is also a common and successful discretisation level in prior literature relating to BNs in neuroscience [20].

It was decided to run searches over 1 million networks in 3 threads for each BN. This was chosen as a balance between accuracy and the limited resources available. The consensus network, a type of 'average graph' across the highest scoring networks found, is used in results; however, these are almost identical to the top network itself. The number of top networks stored was varied between 10 and 500 with little to no differences found, so 10 top networks were stored in searches for the remainder of the project.

It was also attempted to apply a zero phase FIR filter of a different width to 0.2–40 Hz, however, this did not have any immediate stand-out results, so it was decided to not devote more time to varying such an extra parameter. Another parameter modified was the Markov lag, though once again this was ultimately held at 1 to reduce the number of varying parameters due to limited time. At the 200 Hz sampling rate, this means that edges in the BNs indicate that a variable is conditionally dependent on another 5 milliseconds in the past; in other words, it conveys neural information flow that needs 5 milliseconds to travel between the two regions.

2.2 Networks

There are many technical considerations one must take into account before searching for networks from the EEG data. Edges between spatially adjacent electrodes may not be due to neural flow but rather to these electrodes picking up on the same signal due to volume currents. Hence, longer edges

could be considered to convey meaningful connectivity patterns than shorter ones. It is not enough to simply create visualisations across subjects by setting the width or transparency of an edge to be determined by how many times that edge occurs in each network because this would ignore the fact that networks may have edges that are not exactly the same but convey almost identical neural flow. Subjects will have slightly differently shaped skulls, so the exact positions of each electrode will vary between subjects.

One way of handling this is to apply what we will call a soft overlay, which allows edges between similarly spatially located nodes to be counted as the same. To do this we can count the number of occurrences of an edge from node A to node B across subjects for a specific task as the number of subjects whose network for that task contains an edge that travels from node A, or an adjacent node, to node B, or an adjacent node. Then the opacity of each edge can be scaled by this value. However, the number of edges shown to go from node A to node B is determined by the number of subjects whose DBN for that task contains that exact edge. These soft overlays allow us to visualise common structures in the networks across subjects, within each task. This can be useful for visually comparing networks between tasks.

Figures 4–6 show these soft overlays across the 27 subjects used for each task. The nodes, representing the EEG channels, are drawn corresponding to their spatial location; frontal electrodes are drawn at the top and occipital electrodes at the bottom. The longer an edge is, the more of a red hue it has; this is done to add emphasis to edges between nodes that are spatially further apart. The shortest edges, those between adjacent nodes, have been explicitly hidden to improve the readability of the networks. Moreover, the soft counting method is not effective with shorter edges, as it is likely that a similar edge will be found in all networks across subjects. The opacity of each edge is calculated as $(\frac{C}{N})^2$ where C is the soft count of that edge across subjects, and N is the total number of subjects, 27.

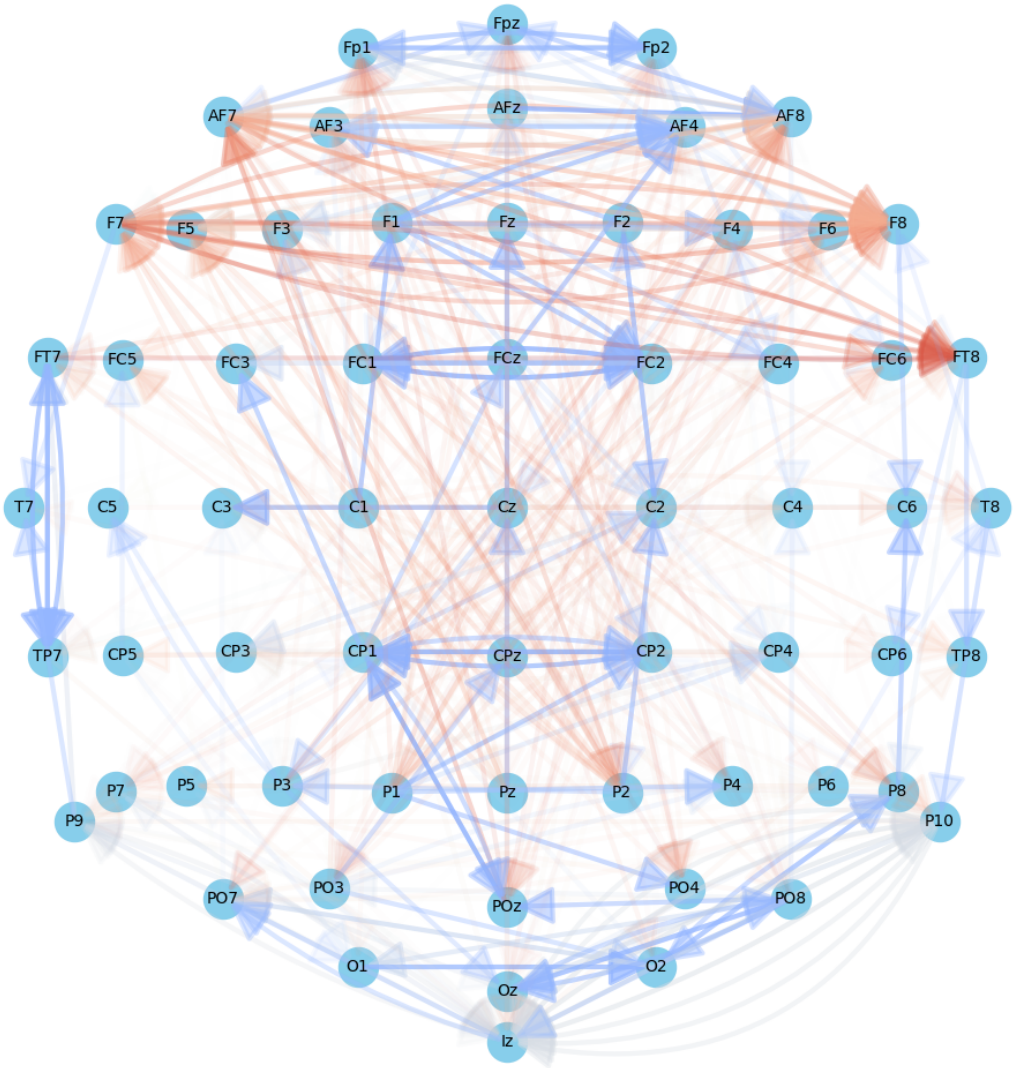

Figure 4: No task soft overlay across 27 subjects. Frontal electrodes are drawn at the top and occipital electrodes at the bottom. Longer edges are drawn to have a more red hue. The opacity of an edge is determined by how common it or similar edges are in subjects' no task networks.

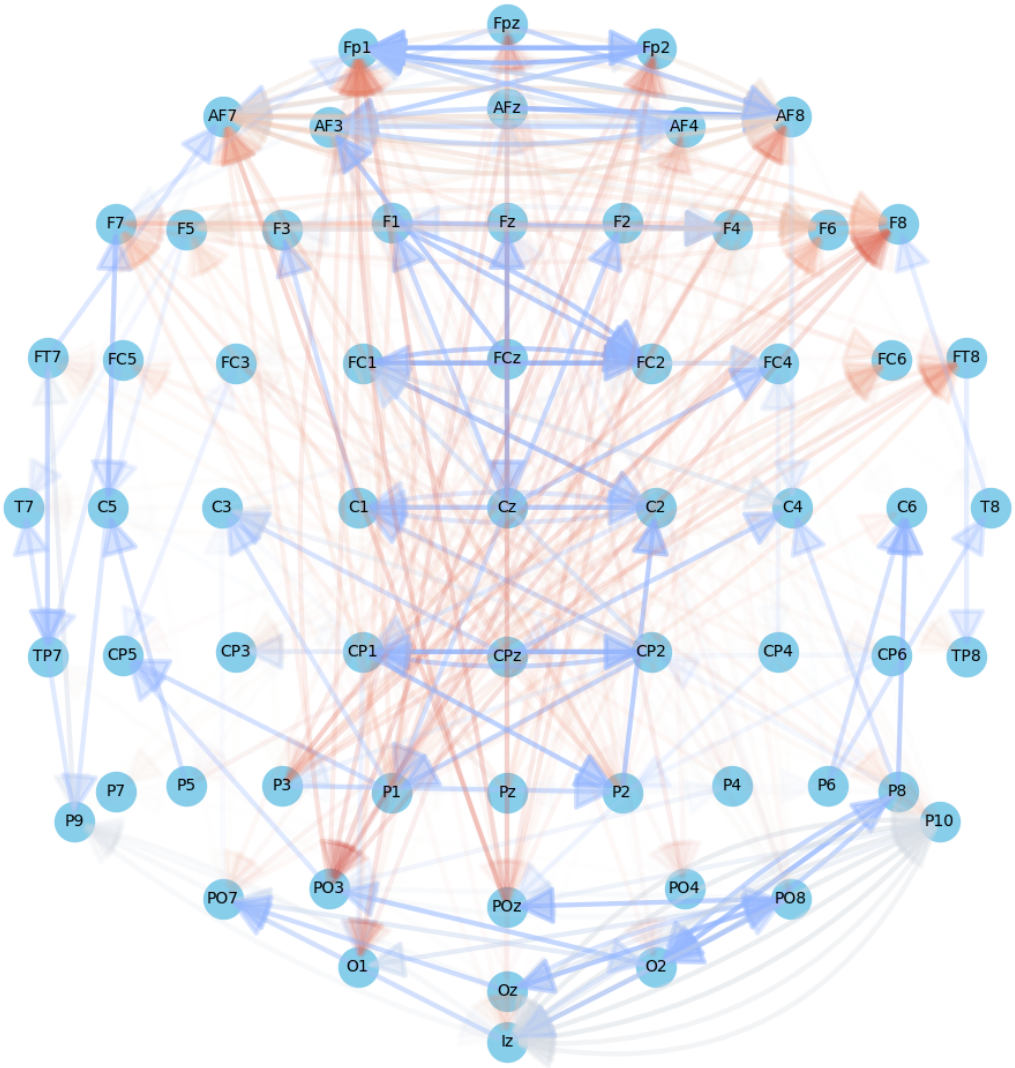

Figure 5: Semantic task soft overlay across 27 subjects. Frontal electrodes are drawn at the top and occipital electrodes at the bottom. Longer edges are drawn to have a more red hue. The opacity of an edge is determined by how common it or similar edges are in subjects' semantic task networks.

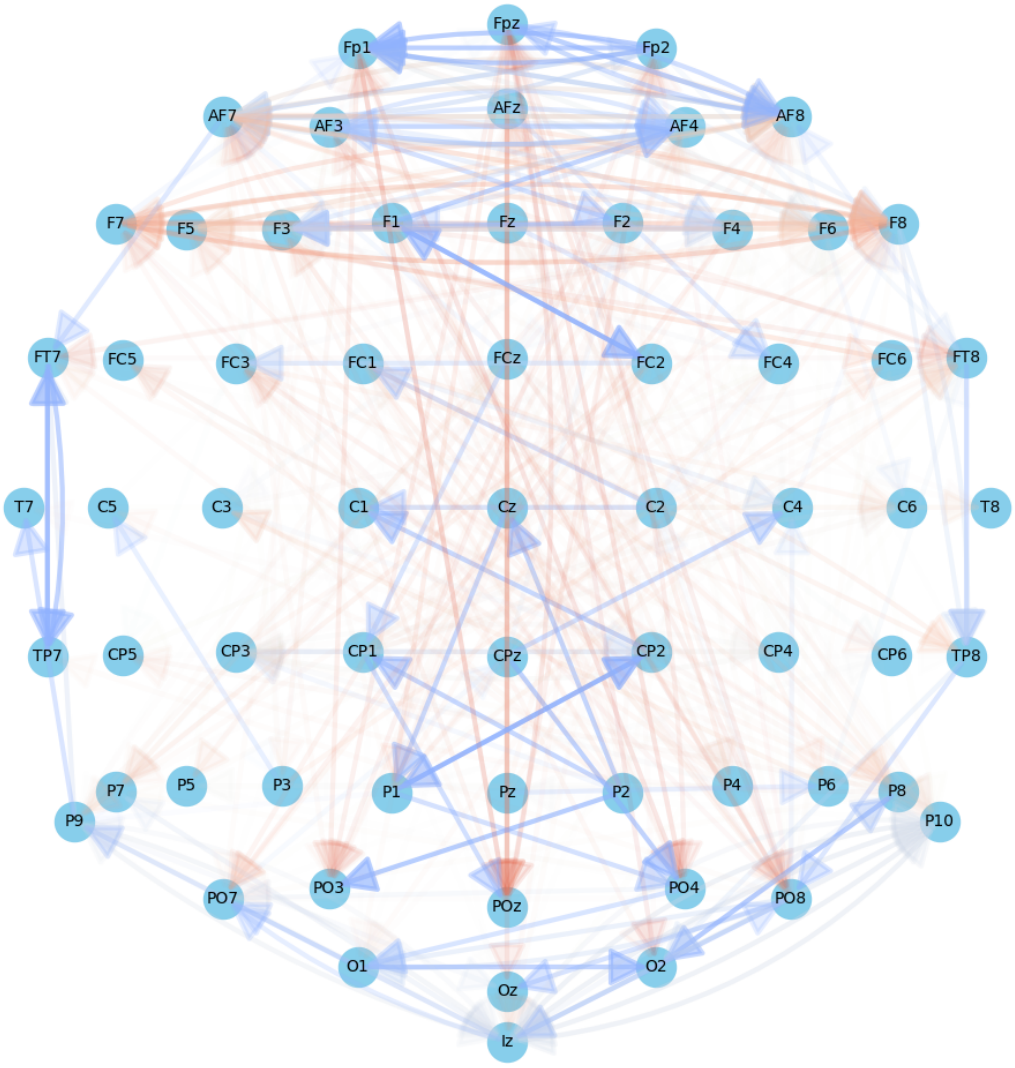

Figure 6: Auditory task soft overlay across 27 subjects. Frontal electrodes are drawn at the top and occipital electrodes at the bottom. Longer edges are drawn to have a more red hue. The opacity of an edge is determined by how common it or similar edges are in subjects' auditory task networks.

Initial observations can be made from these. The no task soft overlay (Figure 4) shows strong bilateral frontal activity, which is expected by related literature [6]. The semantic task soft overlay (Figure 5) predominantly has strong cross-hemispheric activity, which is also a promising result, as cross-hemispheric activity is known to increase with task difficulty [5]. The auditory task soft overlay (Figure 6) appears to contain the fewest opaque edges.

2.3 Classifier

One of the main goals of this work is to be able to differentiate between tasks from BNs. Instead of leaving this to the human eye, we can train a classifier model that takes as input a set of features, in our case usually a list of 1s and 0s representing whether an edge exists or not, and outputs what task it believes the subject was doing which resulted in that BN. This is repeated for each subject, meaning that 26 subjects (78 networks) are used as training data, and 1 subject (3 networks) is used as testing data. The average classification accuracy across all 27 repeats is then calculated. It is important to not mix networks from the same subject in both the training and testing data, as these are not independent of each other. A high-accuracy classifier could also importantly allow us to find which structures and edges are indicative of certain tasks. In the early stages of this work, an unrefined random forest classifier [2] was used to make some basic comparisons between different methods of describing BNs. As will be described later, this was refined once a method had been chosen. Initially, the classifier was used on the 64 node networks, though, as would be expected, the features are far too noisy at this level and classification accuracies were near one-third, meaning random.

2.4 Node Degrees

In an attempt to further analyse which brain regions were receiving and emitting neural information, and whether different tasks exhibit differences in this, the heat maps shown in Figures 7 and 8 were created. The red heat maps in Figure 7 show which parts of the brain were receiving neural information the most, by shading those regions in a darker colour. This was calculated using the average in-degree of each node and its adjacent nodes, ignoring the contribution of the short edges between adjacent nodes. The blue heat maps in Figure 8 function similarly, but instead indicate which parts of the brain are emitting the most neural information using out-degrees.

Each heat map also contains some clusters of highlighted nodes. These represent groups of adjacent nodes that are shaded in an especially darker colour. The edges that go into and leave these highlighted nodes for Figures 7 and 8, are also drawn, respectively.

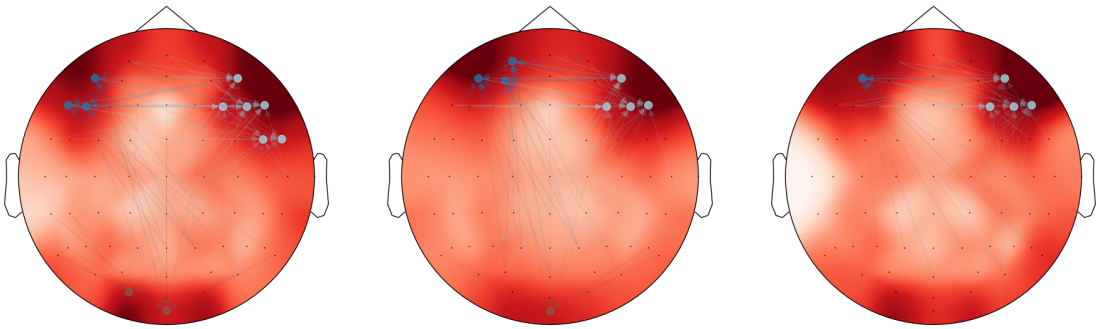

Figure 7: Heat maps for receiving neural information for each task. From left to right: no task, semantic task, auditory task. Darker regions indicate higher smoothed in degree of networks, and lighter regions indicate lower.

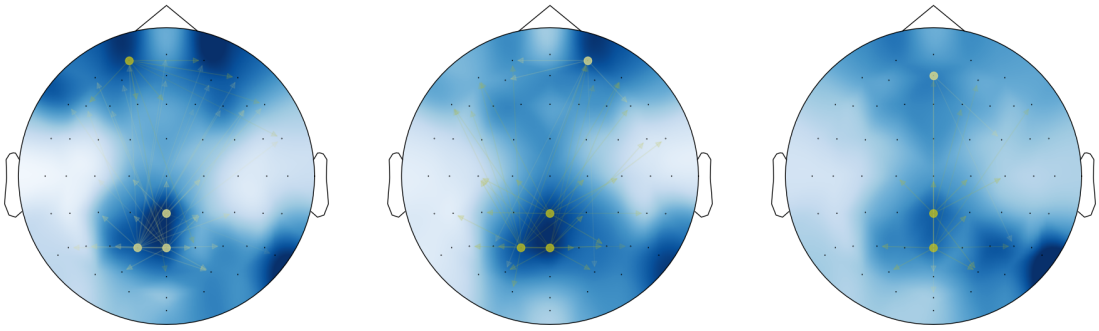

Figure 8: Heat maps for emitting neural information for each task. From left to right: no task, semantic task, auditory task. Darker regions indicate higher smoothed out degree of networks, and lighter regions indicate lower.

Both sets of heat maps have dark frontal regions. This would be expected by comparing to the soft overlays in Figures 4 – 6, which show a lot of edges entering and leaving the nodes in the frontal regions. There are many slight modifications that can be made to this process to alter the heat maps, including simply taking the degree rather than smoothing it by taking the average between spatial neighbours, though it is unclear which is most suitable. Although the heat maps presented here provide some level of information on directionality, which could be very useful, they do not seem to help to differentiate between the three tasks.

It was briefly attempted to train a classifier with node degrees (and smoothed node degrees) as features rather than edge existence; however, this did not prove to be successful. Doing this only removes information, since node degree information is provided in edge existence features, and clearly the information removed is not exclusively noise.

2.5 Clustering

Instead of smoothing, as is done in the soft overlays and slightly differently in the heat maps, we could instead spatially cluster nodes together to obtain networks with fewer nodes overall. The clustering can be performed on the channels before the searching process, or on the 64 node networks after the searching process. The prior was attempted both by combining channels by taking the mean, and also by using PCA. The channels were split into groups of varying sizes, including 5, 8 and 10, according to their spatial location. The advantage of searching for such smaller networks is that they require far less computing power, so deeper analyses can be performed. However, this strategy repeatedly offered little consistency and accuracy and so was ultimately set aside.

Instead, the more successful method was the latter option, clustering the nodes together from the 64-node network. This can be done by separating nodes into clusters, representing each cluster as a node in a new network, and drawing an edge from cluster A to cluster B if there is at least one edge going from a node within cluster A to a node within cluster B. It was initially attempted to cluster nodes together based off regions in which we would expect channels to behave similarly (e.g. left frontal, right temporal, etc.), though this can be quite arbitrary.

To find a more methodical approach to clustering nodes together, it was necessary first to estab-

lish a way of measuring how similar nodes are to each other. Inspired by the soft overlays, spatially adjacent nodes A and B were defined to be *similar* in a network X if there exists a directed edge from/to node A to/from some node C in some network Y, and there is a directed edge, respectively, from/to node B to/from some node D which not spatially adjacent to node B in network X, where node D is either the same as node C itself or is spatially adjacent to C. We add the restriction that node B cannot itself be spatially adjacent to node D, because short edges can be misleading. We then count the number of times each pair of nodes is similar, across all the different networks, to obtain values for how similar those nodes, effectively creating a meta-graph.

With 27 subjects used and 3 tasks per subject, the maximum weight of an edge in the meta-graph is 81. Equipped with these edge weights, which indicate a strength of how similar nodes are to each other, we can apply a clustering algorithm to split the 64 nodes into groups. The clustering algorithm that was used for this is known as the Louvain clustering method and is described in [1]. This method has a resolution parameter, which affects the size of clusters and therefore how many clusters the 64 nodes are split into. To find the optimal resolution, and hence the number of clusters, the clustering method was used on many different resolution values (repeated multiple times to find global peak), and then new networks were created for these clusters. Then these networks were provided to a basic random forest classifier as edge features, of which the accuracy was recorded. The highest accuracy came from the clusters generated from the Louvain clustering method having a resolution of 9.0, which created 23 clusters (see Figure 9). However, there was not a single peak during this search and, although the final choice also had a low variance among the 5 classification runs, other resolutions could have easily been chosen too (see Figure 10). Ultimately, this process simply gives loose justification for the clusters, which is what is desired.

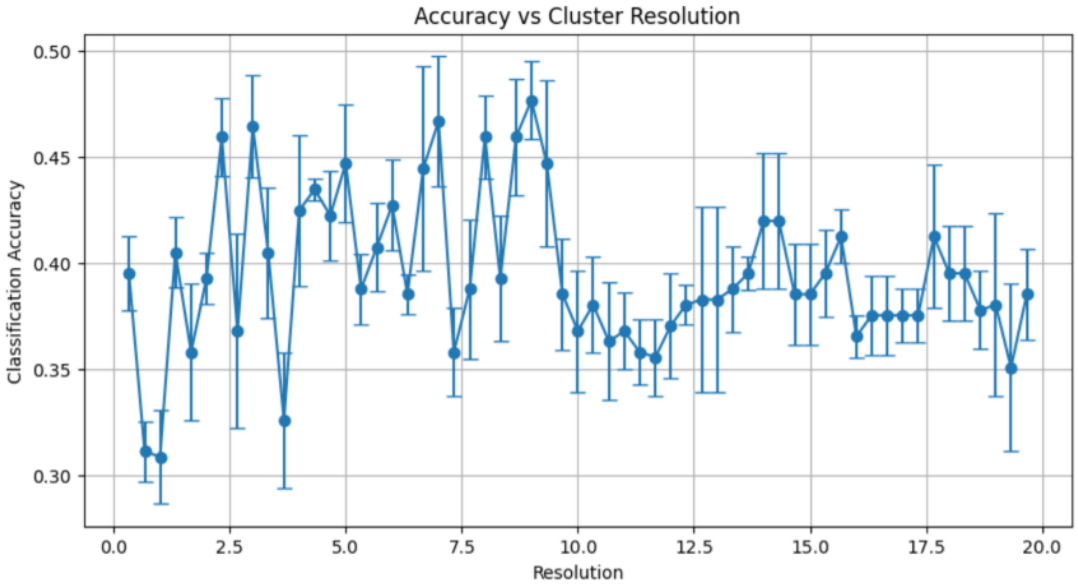

Figure 9: Plot of accuracy of random forest classifier for graphs generated from the clusters against the resolution the clusters were generated from. Peak at resolution 9.0. Error bars indicate variance over 5 runs of the classifier, done due to the random element of the specific classifier model used here. The dots indicate mean accuracy.

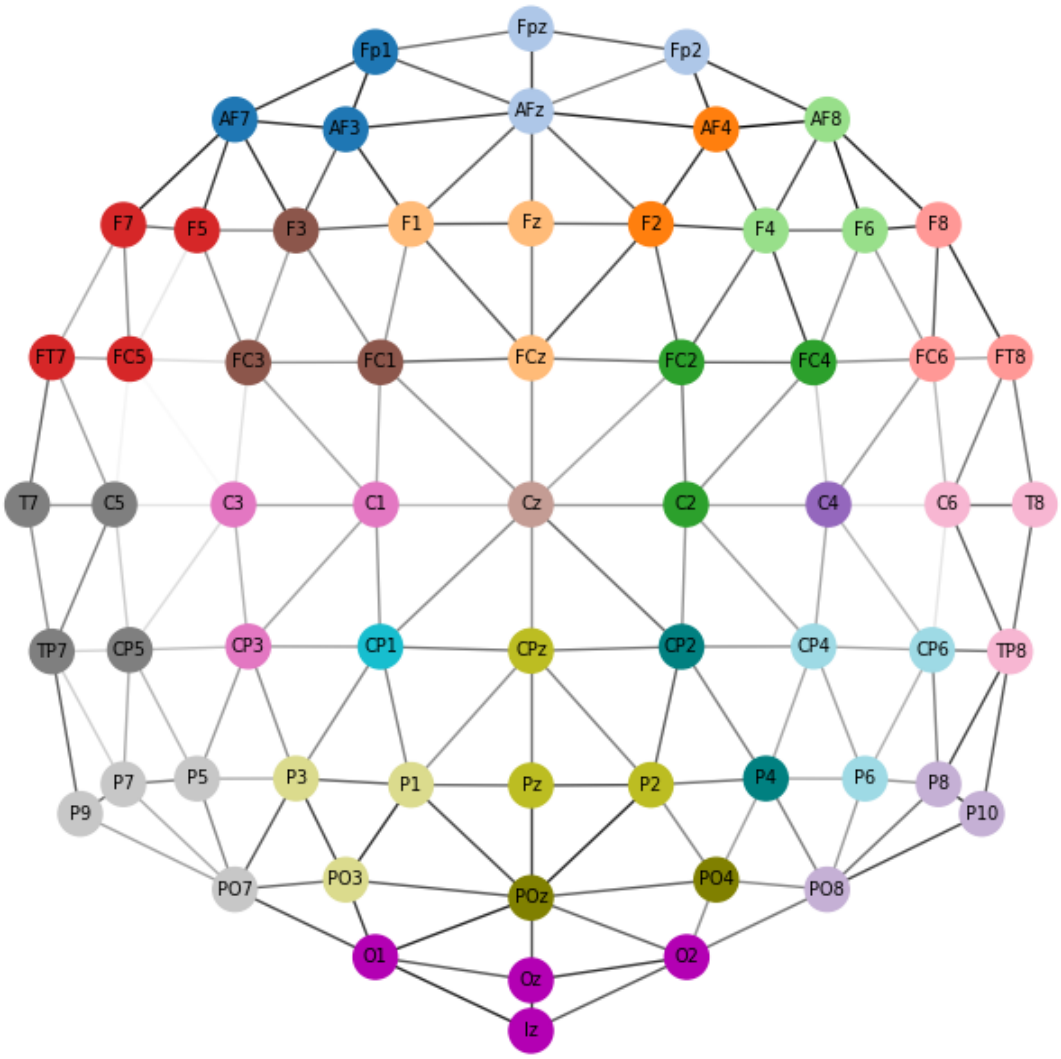

Figure 10: Meta network constructed from networks from all 27 subjects and 3 tasks per subject. The edge opacity is calculated as $(\frac{C}{N})^4$, where C is the number of networks in which the two nodes connected by an edge are treated as similar, and N is the total number of networks, 81. C is also the edge weight used for the Louvain clustering method, here with a resolution of 9. This has split nodes into 23 clusters; each cluster is shown by highlighting nodes in the same cluster with the same colour.

These clusters can then be used to create new 23 node networks as described (see Figure 11).

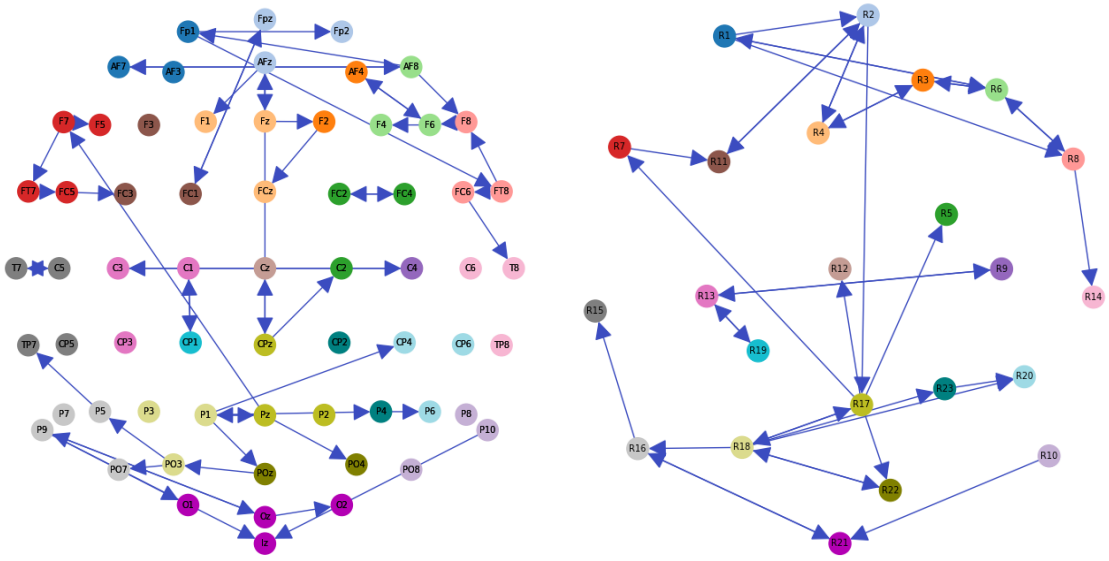

Figure 11: On the left, no task 64-node network for a subject. The nodes are highlighted according to which cluster they will go into. On the right, the clustered network with 23 regions, labelled R1–R23. Region coordinates are set as the average coordinate of its constituent nodes. An edge exists from Ri to Rj if there is an edge in the 64-node network going from a node within Ri to a node within Rj.

2.6 Refinement

To further improve the accuracy of the classifier, it was decided to reduce the number of edge features, from the current $23 \times 22 = 506$. Many features provide more noise than information. The initial method used to achieve this worked as follows. For each subject, find difference graphs (a graph exclusively containing edges found in graph A but not in graph B) between each pair of networks from the three tasks. This results in six difference graphs (two per pair). Then create a set consisting of all the edges in these six difference graphs; these are the edges that differentiate the networks for the different tasks for this subject. This is repeated for each subject. Finally, only the edges found in at least a certain fraction of these sets will be kept as features. The optimal fraction was found to be $\frac{3}{27}$, leaving us with 82 features (see Figure 12).

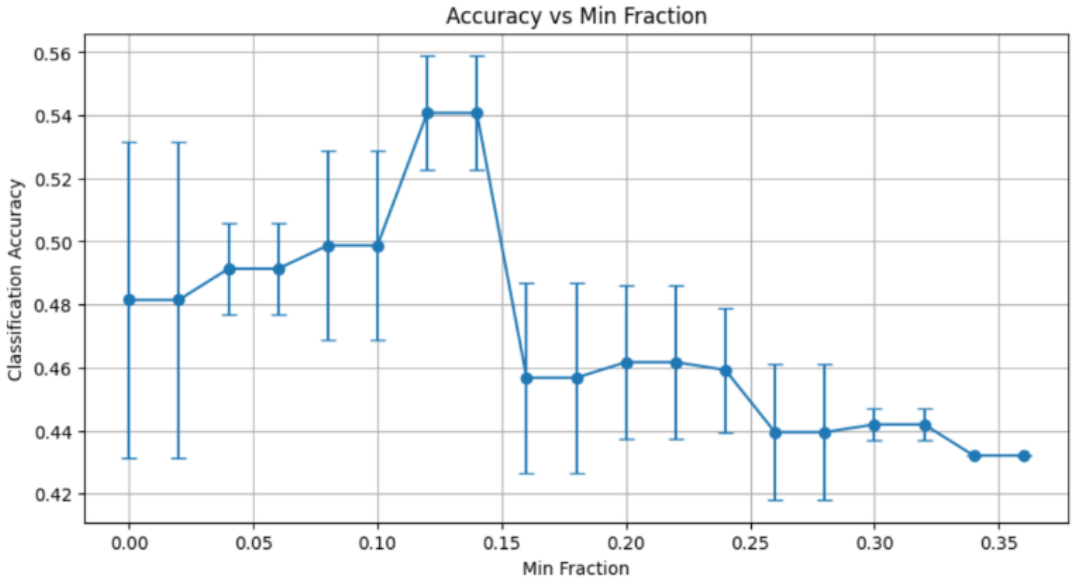

Figure 12: Plot of accuracy of random forest classifier against the minimum fraction used to obtain the features it was trained on. Error bars indicate variance over 5 runs of the classifier. Dots indicate mean accuracy. Peak at fraction corresponding to at least 3 out of 27 subjects.

To further filter out noisy features, the permutation importance of each feature was calculated. This is a measure of how much accuracy decreases on average when a certain feature is randomly permuted while the others are left the same. 29 features were found to have a negative permutation importance, which means that permuting them randomly improved accuracy on average. Hence, these features were excluded, leaving 53 edge features.

Alternative classifier models were then tested, including neural network classifiers. Ultimately, it was decided to use a logistic regression classifier (described in [23]), due to its high accuracy, non-randomness, and high speed. This was refined by altering the regularisation strength to 2.

A further advantage of using a logistic regression classifier is that the coefficients of the model, when trained on all data, can easily be obtained to provide insight into which features (edges) are suggestive of a specific task and to what extent. It was noticed that some features had near-zero weights in all 3 tasks, suggesting that these may also provide more noise than information. To find what effect removing these features would have, the features were ordered by the sum of the absolute value of their weights across the 3 tasks. The lowest n features were then removed, for different values of n, and the accuracy of the classifier trained on the resulting smaller set of features was recorded. From this it was found that the highest accuracy was obtained when the bottom 27 features were removed, leaving 26 features (see Figure 13).

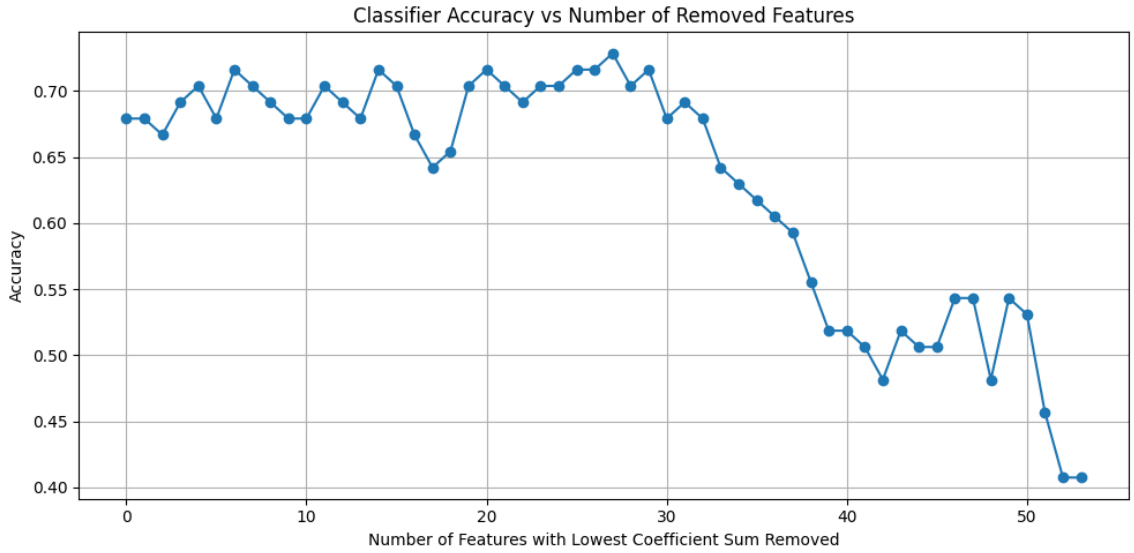

Figure 13: Plot of accuracy of logistic regression classifier against the number of features with the lowest sum of absolute value of coefficients across the 3 tasks being removed. No clear pattern is visible up until the peak at 27, which is followed by a sharp drop in accuracy. This minimises the number of features while maximising accuracy, which aids the simplicity of further analysis.

2.7 Analysing the Classifier

Figures 14–17 contain various plots detailing the performance of the logistic regression classifier when trained on the remaining 26 features. The final accuracy obtained was 0.728.

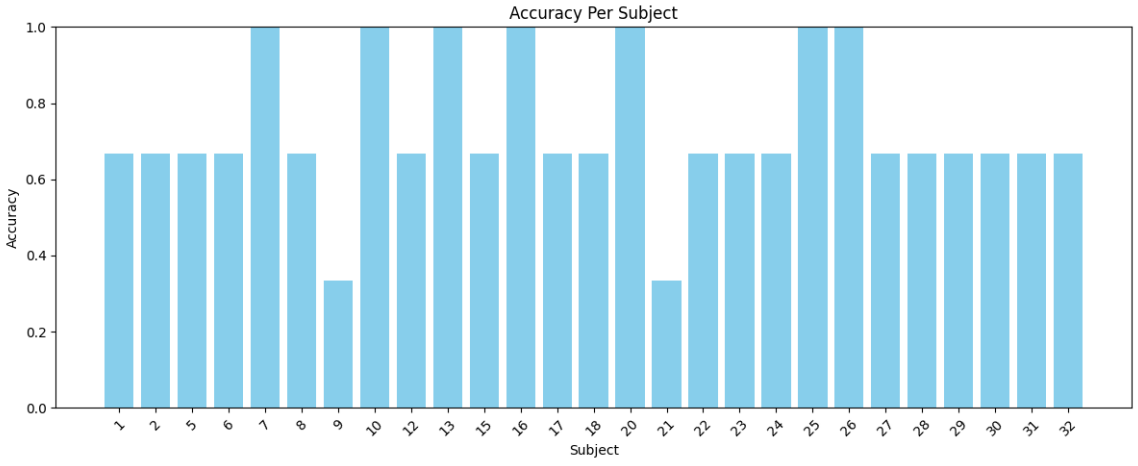

Figure 14: Plot of the accuracy of the classification for each of the 27 subjects. For each subject, the model is trained on all other subjects before classifying the subject's networks for each of the 3 tasks. Subject number labels skip some values due to these subjects not being used.

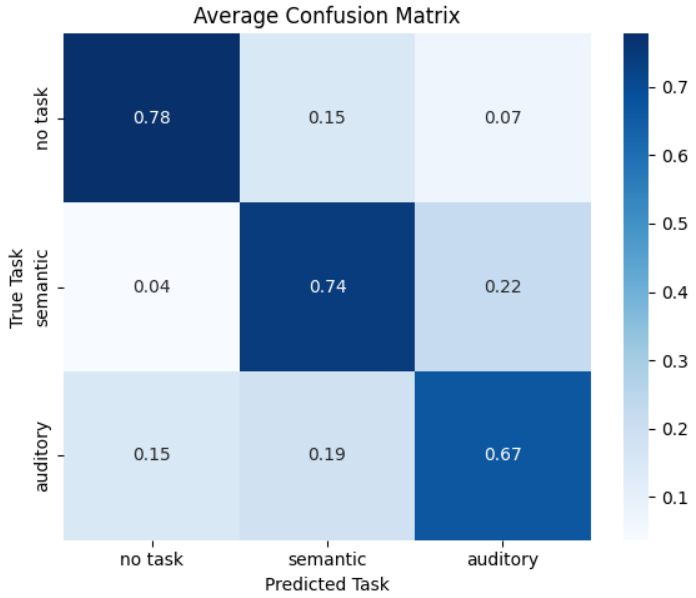

Figure 15: Confusion matrix indicating proportions of each classification for each true class.

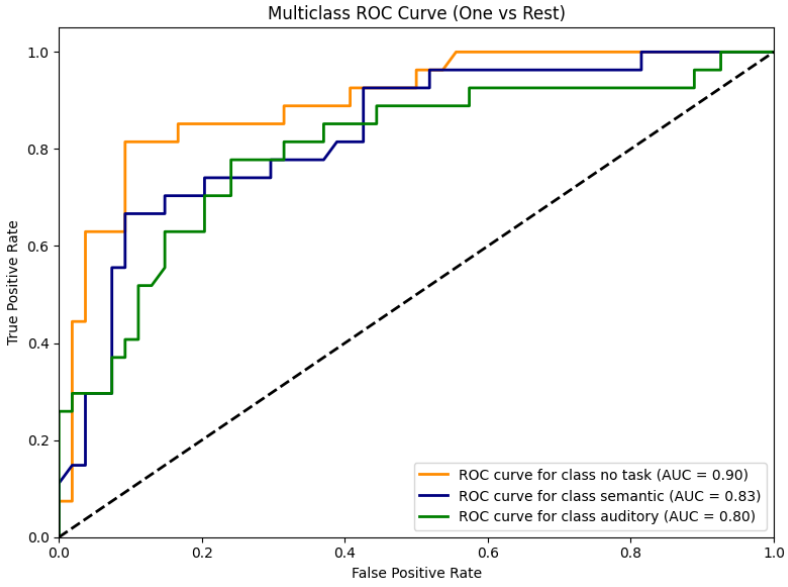

Figure 16: Receiver operating characteristic (ROC) curve indicating the trade-off between true positive rate and false positive rate, calculated for each class as itself or either of the others. The area under the curve for each curve is indicated in the legend to the lower right of the graph.

Class	Precision	Recall	F1-score
no task	0.685	0.778	0.716
semantic	0.586	0.741	0.636
auditory	0.537	0.667	0.580

Figure 17: Table indicating the precision for each class (the proportion of how many classifications were correct when that class was the guess) and recall (the proportion of how many classifications were correct when that class was being guessed). The F1 score is the harmonic mean of these.

We can use the remaining features of the model, and their coefficients, to analyse what structures help to differentiate the most between tasks and in what way. Figures 18 – 20 show for each task the sign and magnitude of the coefficient of each feature (edge).

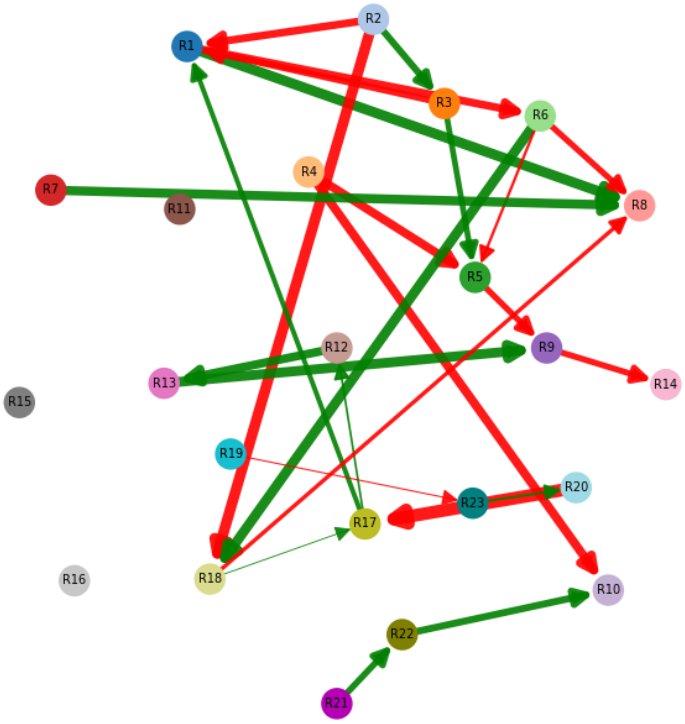

Figure 18: No task classification coefficients. Green edges indicate positive coefficients (edge associated with no task). Red edges indicate negative coefficients (edge associated with not being no task). Width of each edge is proportional to the coefficient magnitude (strength of the association).

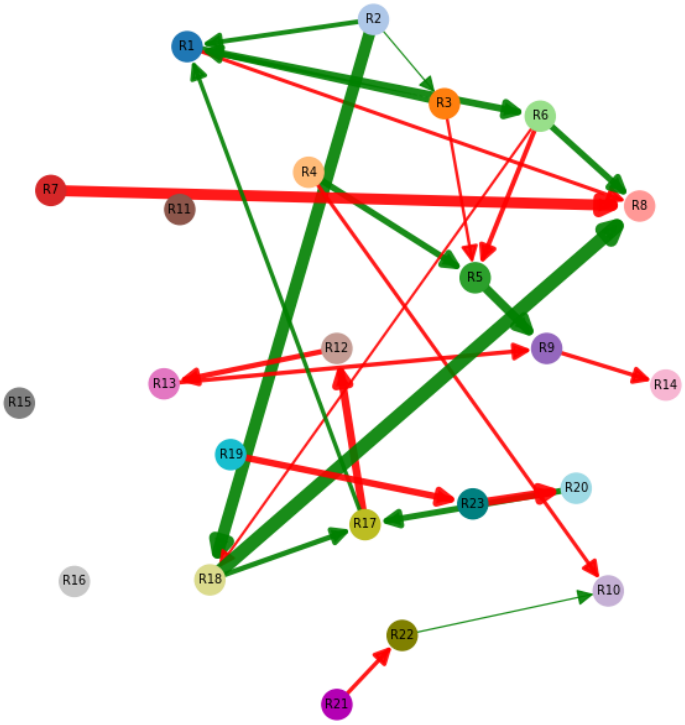

Figure 19: Semantic task classification coefficients. Green edges indicate positive coefficients (edge associated with semantic task). Red edges indicate negative coefficients (edge associated with not being semantic task). Width of each edge is proportional to the coefficient magnitude (strength of the association).

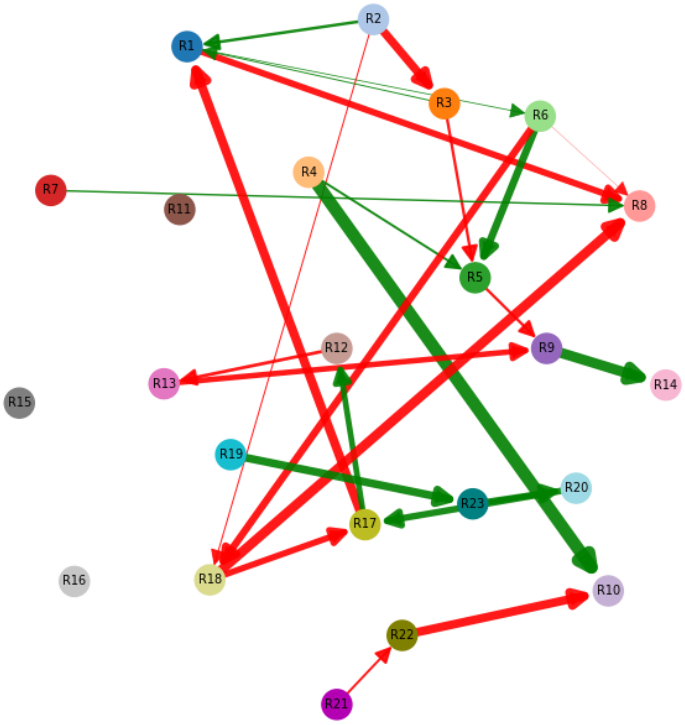

Figure 20: Auditory task classification coefficients. Green edges indicate positive coefficients (edge associated with auditory task). Red edges indicate negative coefficients (edge associated with not being auditory task). Width of each edge is proportional to the coefficient magnitude (strength of the association).

Immediately apparent from these figures is the lack of edges in the left temporal and parietal regions, meaning these edges were filtered out as unhelpful. This suggests that edges connecting to regions in the left hemisphere do not help to differentiate between tasks. In fact, this is the area of the brain responsible for basic speech processing [11], which we would expect all 3 tasks to perform.

Figure 18 shows thick green lines in the frontal region between R7 and R8, as well as R1 and R8 for no task. This corresponds to the heavy frontal activity seen in the no task soft overlay (Figure 4). Figure 19 shows stronger green edges between the temporal and frontal regions, which is expected when focusing more in the semantic task [25].

Figure 20 shows the auditory task to have fewer thick green edges than the other two tasks. This corresponds to the auditory soft overlay in Figure 6, which showed that the auditory task has generally more transparent edges, suggesting that the networks contain fewer edges in general. The thickest green edge for the auditory task is from R4 to R10, indicating that the edges from the central frontal region to the right parietal region are indicative of the auditory task. In fact, this is what we would expect, because the right hemisphere plays a crucial role in processing fine-grain

patterns in speech, as would be done in the auditory task [12].

3 Discussion

It is clear that there is a lot of noise captured by BNs from the EEG data; however, we have shown that it is possible to use certain methods to filter some of this out while retaining important information. The greatest similarities between networks are found when comparing the networks for different tasks for a single subject; however, task-specific consistencies in the networks also exist, evidenced by the high-accuracy classifier we have obtained. When we find what these task-specific structures are, they show some results that are supported by fundamental knowledge in the field of neuroscience [11][12][25]. This clearly suggests that Bayesian networks are, to some extent, a useful tool that can be applied to EEG data.

This work experimented with many different methods for creating and analysing the BNs obtained from the EEG data. A lot of time was spent trying to extract useful information from node in- and out-degrees, as this summarises which regions of the brain are receiving and emitting information. This ultimately did not prove to be useful. It was evident that some form of combining of nodes was required to reduce the noise in the system; the soft overlays were a start at this; however, they do not provide a rigorous way to assess differences between tasks. The clustering of nodes into regions helped enormously, more so when the clustering was performed computationally in a principled way, rather than manually based on assumptions. The classifier provided both a way to measure the success of each strategy and also had uses as a tool itself to uncover identifying structures.

It was also briefly attempted to calculate the p-values for the edge existence within a task across subjects using a Mann-Whitney U-test. This was never very successful, with any low p-values not surviving the FDR correction applied afterwards. Only 27 subjects and 81 EEG recordings were used for this work. With a larger number of subjects, it is likely that higher classifier accuracies, lower p-values, and more precise task-specific structures can be obtained.

4 Future Research

It is planned to submit a paper version of this work to a journal to be published. We have shown that there is promising potential for BNs to be applied to EEG data. This work mainly focused on discovering this potential and laying the groundwork. This opens opportunities in future to obtain more novel discoveries in the field of neuroscience through more innovative and refined applications of BNs to EEG data.

A specific area that was left out of this work is the variation of the Markov lag. The ability to change what time frame of neural flow is being analysed could be very powerful and could help to gain insight into differences between short- and long-term neural flow. It is also very possible that the BNs learned from certain time frames are more useful for inference than others.

This work briefly tried to isolate specific band frequencies by applying band-pass filters of varying ranges to the EEG data in pre-processing. It was clear that this has massive effects on the networks generated, though this meant that there would not be enough time to analyse each of these, so the

full range was used for the rest of the work. More work could be done to assess the differences in the selected frequency ranges.

5 Acknowledgments

I would like to thank the Laidlaw Foundation for their generous support in the Laidlaw Scholars Leadership and Research Programme, without which this work would not be possible. I would also like to thank my supervisor, Dr V. Anne Smith (School of Biology), whose knowledge and guidance has been essential. Finally, I would like to extend my gratitude to the team that gathered and pre-processed the data used in this work at the University of Aberdeen; this was led by Dr Anastasia Klimovich-Gray, who additionally provided very valuable advice and expertise.

References

[1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics*, 2008(10):P10008–, 2008.

[2] Leo Breiman. Random forests. *Machine learning*, 45(1):5–, 2001.

[3] Giovanni Chiarion, Laura Sparacino, Yuri Antonacci, Luca Faes, and Luca Mesin. Connectivity analysis in eeg data: A tutorial review of the state of the art and emerging trends. *Bioengineering*, 10(3), 2023.

[4] Kathryn A. Davis, Seth P. Devries, Abba Krieger, Temenuzhka Mihaylova, Daniela Minecan, Brian Litt, Joost B. Wagenaar, and William C. Stacey. The effect of increased intracranial eeg sampling rates in clinical practice. *Clinical neurophysiology*, 129(2):360–367, 2018.

[5] Simon W. Davis and Roberto Cabeza. Cross-hemispheric collaboration and segregation associated with task difficulty as revealed by structural and functional connectivity. *The Journal of neuroscience*, 35(21):8191–8200, 2015.

[6] Gaelle E. Doucet, Xiaosong He, Michael R. Sperling, Ashwini Sharan, and Joseph I. Tracy. From "rest" to language task: Task activation selects and prunes from broader resting-state network. *Human brain mapping*, 38(5):2540–2552, 2017.

[7] Hana El-Samad. Biological feedback control—respect the loops. *Cell systems*, 12(6):477–487, 2021.

[8] Alexandre Gramfort. Meg and eeg data analysis with mne-python. *Frontiers in neuroscience*, 7:267–, 2013.

[9] James Hammond and V. Anne Smith. Bayesian networks for network inference in biology. *Journal of the Royal Society Interface*, 22(226):20240893–, 2025.

[10] David Heckerman, Dan Geiger, and David M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

[11] Gregory Hickok and David Poeppel. The cortical organization of speech processing. *Nature reviews. Neuroscience*, 8(5):393–402, 2007.

[12] Annukka K. Lindell. In your right mind: Right hemisphere contributions to language processing and production. *Neuropsychology review*, 16(3):131–148, 2006.

[13] Franziska Matthäus, V. Anne Smith, Peter J Gebicke-Haerter, Georg Winterer, Felix Tretter, Eduardo R Mendoza, and Peter J Gebicke-Haerter. Some useful mathematical tools to transform microarray data into interactive molecular networks. In *Systems Biology in Psychiatric Research*, pages 277–300. Wiley-VCH Verlag GmbH Co. KGaA, Weinheim, Germany, 2010.

[14] Isobel Milns, Colin M Beale, and V. Anne Smith. Revealing ecological networks using bayesian network inference algorithms. *Ecology*, 91(7):1892–1899, 2010.

[15] A.Y. Mutlu and S. Aviyente. Inferring effective connectivity in the brain from eeg time series using dynamic bayesian networks. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2009, pages 4739–4742, United States, 2009. IEEE.

[16] Aaron J. Newman. Averaging and re-referencing erps. In *Neural Data Science in Python*, 2025.

[17] Jagath C. Rajapakse and Juan Zhou. Learning effective brain connectivity with dynamic bayesian networks. *NeuroImage (Orlando, Fla.)*, 37(3):749–760, 2007.

[18] Jürgen Sladeczek, Alexander J. Hartemink, and Joshua Robinson. Banjo user guide, 2005.

[19] V. Anne Smith. Revealing structure of complex biological systems using bayesian networks. *Network science*, pages 185–204, 2010.

[20] V. Anne Smith, Erich D Jarvis, and Alexander J Hartemink. Influence of network topology and data collection on network inference. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 164–, 2003.

[21] V. Anne Smith, Jing Yu, Tom V. Smulders, Alexander J. Hartemink, and Erich D. Jarvis. Computational inference of neural information flow networks. *PLoS computational biology*, 2(11):e161–1449, 2006.

[22] Bas van Steensel, Ulrich Braunschweig, Guillaume J. Filion, Menzies Chen, Joke G. van Bemmel, and Trey Ideker. Bayesian network analysis of targeting interactions in chromatin. *Genome Research*, 20(2):190–200, 2010.

[23] Hsiang-Fu Yu, Fang-Lan Huang, and Chih-Jen Lin. Dual coordinate descent methods for logistic regression and maximum entropy models: Special issue on model selection and optimization in machine learning. *Machine learning*, 85(1-2):41–75, 2011.

[24] Jing Yu, V. Anne Smith, Paul P. Wang, Alexander J. Hartemink, and Erich D. Jarvis. Advances to bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20(18):3594–3603, 2004.

[25] Qiuhai Yue, Linjun Zhang, Guoqing Xu, Hua Shu, and Ping Li. Task-modulated activation and functional connectivity of the temporal and frontal areas during speech comprehension. *Neuroscience*, 237:87–95, 2013.

[26] Lei Zhang, Dimitris Samaras, Nelly Alia-Klein, Nora Volkow, and Rita Goldstein. Modeling neuronal interactivity using dynamic bayesian networks. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'05, page 1593–1600, Cambridge, MA, USA, 2005. MIT Press.