A-T-L-A-S Atlas!

Nisarg Suthar, Microsoft, India.

Tasks Completed

1. Dataset Creation ✓
2. Graph Analysis ✓
3. Community Detection ✓
4. Bonus
   (a) Link Prediction using Node2Vec ✓
   (b) Link Prediction using GNN ✓
5. Paper Reading Task ✓

I. INTRODUCTION

ATLAS is a classic word game where players name geographical locations. Each name must start with the last letter of the previous one (e.g., Madagascar → Russia → Armenia). The game ends when a player cannot provide a valid, previously unused location. This project analyzes the game using complex network theory. It models countries and cities as nodes in a directed graph, where an edge exists if one location's name ends with the letter that the next location's name begins with. Additionally, we dive deep into the Node2Vec algorithm for graph representations.

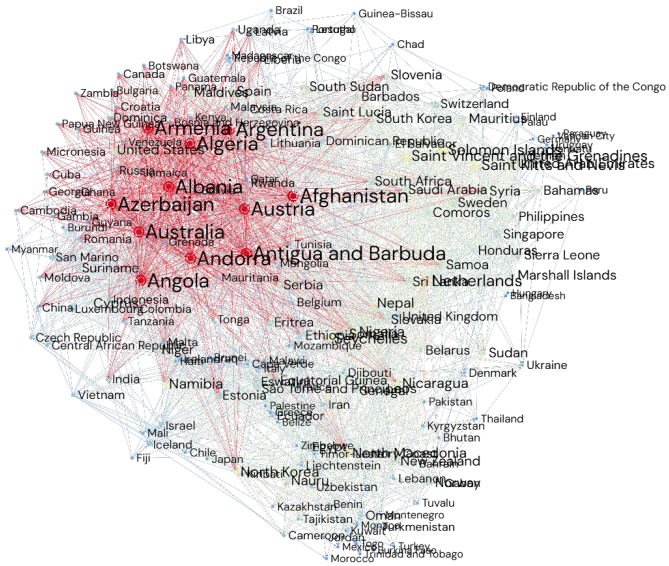

FIG. 1: The ATLAS game graph for countries. [Made using Gephi [1]]

II. DATASET CREATION

The following 3 datasets were created using web scraping with the latest information available as of date on the respective sources.

- Countries Dataset (Names of all 195 UN recognised countries)
- Cities Dataset (Names of the top 500 cities by population globally as of Dec 10, 2025)
- Combined Dataset (The union of the above 2 lists)

The information regarding the country names was obtained from Wikipedia [11] with no changes made to the official names recorded for the countries in the UN list of recognised countries.
The list of cities was obtained from the Geonames dataset on Huwise DataHub. [2]
Further filtering and data formatting can be referred from the code on my GitHub repository [6] under the data folder.

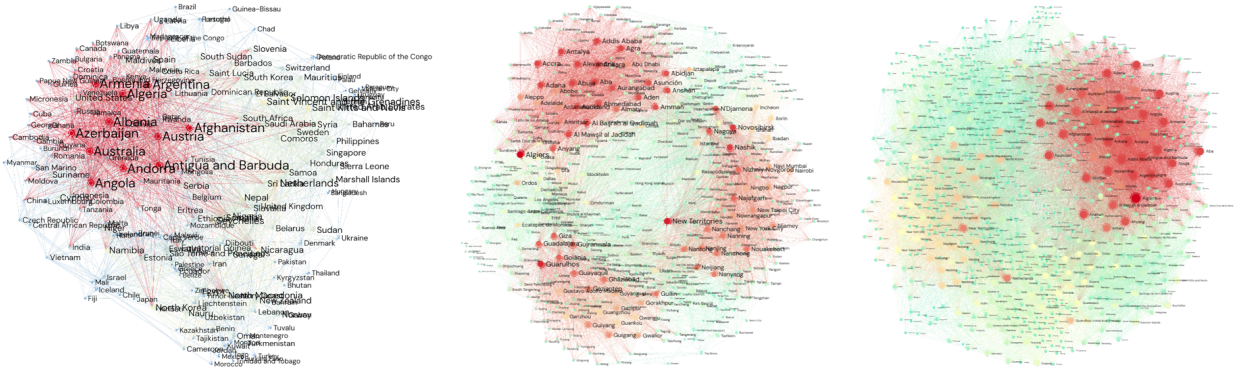

FIG. 2: Graph visualization for the three datasets. From left to right: Countries, Cities, Combined. [Made using Gephi [1]]

III. TASK: GRAPH ANALYSIS

Number of nodes in Country Network = 195
Number of nodes in City Network = 496
Number of nodes in Combined Network = 691

Number of edges in Country Network = 2035
Number of edges in Country Network = 8895
Number of edges in Country Network = 19666

A. Sure Shot Wins

Anyone who says Norway after Yemen wins!

1. Based on Cut Nodes

Cut vertices disconnect the graph. If there is a cycle from the cut vertex to itself, then the cycle can be used as a sure shot win.

For example in the country graph, Yemen and Oman are cut vertices. I have prepared a list of 3-length, 4-length, 5-length cycles that bring you back to Oman or Yemen.

Yemen → Nauru → Uruguay
Oman → Nepal → Lesotho
and so on,

Refer the `code/analysis/winning_paths.txt` for more paths.

B. Network Level Properties

1. Degree Distribution

I was looking for scale free distribution to see if the network is resilient to node deletions. However, i could not see any significant scale-free behaviour proving that the node deletions are significant for the graph structure. This makes sense as the game prohibits repeating a country which translates to node deletion in the graph. The plots can be seen in under the `code/analysis/network_properties` folder in the repo.

2. Mean Degree

Mean Degree for Country Network: 10.435
Mean Degree for City Network: 17.933
Mean Degree for Combined Network: 28.501

Mean degree is higher for the combined graph reflecting availablility of more options to go from a place in the combined network. Also, cities mean degree is higher than that of country meaning there are more options available at each node. This means the game becomes easier if you have the knowledge of all the places. Mean In-degree and Out-degree values are similar.

3. Density

Density of Country Network: 0.0537
Density of City Network: 0.0362
Density of Combined Network: 0.0413

The density of country network is highest, meaning the nodes are well connected than expected number of links randomly. The lower number density in the other graphs signify there are lower number of node pairs not being an edge due to the ATLAS game constraints. So game in the country is less constraint relative to the number of nodes.

4. Assortativity

Degree assortativity of Country Network: -0.28670908757653174
Degree assortativity of City Network: -0.1339223966568021
Degree assortativity of Combined Network: 0.010356628944605649

Assortativity is relatively more in the combined graph, meaning higher degree nodes prefer to attach to other higher degree nodes. The other 2 networks have negative assortativity meaning that higher degree nodes lead to lower degree nodes and vice versa. This is important as we want to be on a higher degree node and force the opponent to have lesser options in order to win.

5. Transitivity

Transitivity of Country Network: 0.31078401464307503
Transitivity of City Network: 0.10759340829179707
Transitivity of Combined Network: 0.1688104185192622
Higher transitivity means that if A can lead to B, and B can lead to C, then A can often lead directly to C. The game has redundant paths. It can help to strategise and create odd length cycles which are a sure shot win. There is higher chance to do this in the country graph.

6. Avg Clustering Coefficient

Avg Clustering Coefficient of Country Network: 0.18014462954148772
Avg Clustering Coefficient of City Network: 0.06181904785254553
Avg Clustering Coefficient of Combined Network: 0.08414218341202077
Measures how tightly a node's neighbors connect with each other. High clustering around a country means the countries that are reachable from it tend to be inter-reachable. There is a safe zone where nodes can lead to one another instead of going down into a single trajectory. This means that the country graph is safer to play in the beginning.

C. Edge Level Properties

1. Betweenness

Higher edge betwenness means there is high chance that the game play is going to visit either of the 2 nodes. Travelling via a high betweenness edge means going thorugh a structural chokepoint which can force the opponent to a dead end sooner.

Examples of high betweenness edges for country graph:
('Nauru', 'United Kingdom') 0.08051174865607857
('New Zealand', 'Dominican Republic') 0.05613551015612871
('Afghanistan', 'Netherlands') 0.04540561672004976

The highest edge betweenness for the edeges is more in the country graph, meaning that the country graph provides more impact on steering an opponent to a death trap.

D. Node Level Properties

I have analysed various node level properties and centrality measures like Out degree advantage (Out Degree - In degree), Eigen Vector centrality, PageRank, HITS, Trophic Levels, and betweenness centrality. The intent is to send the opponent to a node which has lesser number of options which means lower out degree, higher eigen centrality, higher page rank, higher HITS authority, low HITS hub value, lower trophic level and higher betweenness centrality.

1. Custom Parity Property

I have come up with a custom node level property called parity which is a randomized algorithm to find the difference between the number of odd length and even length terminating paths starting at a node. An odd length path is a sure sign of win for anyone who starts that trajectory. More the number of odd length terminating paths starting at a node, higher the chance of winning.

The code generates a sample of terminating paths that start from a node and calculates the difference in an efficient way. This approach tackles the problem of huge computation times required for listing all the paths from a node.

Listing 1: Randomized Node Parity

```
1
2  def sample_maximal_paths(G, source, num_samples=1000):
3      sampled_paths = []
4
5      for _ in range(num_samples):
6          path = [source]
7          visited = {source}
8          current = source
9
10         while True:
11             neighbors = [n for n in G.successors(current) if n not in
                   visited]
12             if not neighbors:
13                 sampled_paths.append(path.copy())
14                 break
15
16             # Randomly choose next node
17             next_node = random.choice(neighbors)
18             path.append(next_node)
19             visited.add(next_node)
20             current = next_node
21
22     return sampled_paths
23
24 def source_parity_adv(G, source, **kwargs):
25     sample_paths = sample_maximal_paths(G, source, **kwargs)
26     sample_path_lengths = []
27     with ThreadPoolExecutor() as executor:
```

```
28          sample_path_lengths = list(executor.map(lambda path: 1 if len(
                path)%2 == 1 else -1, sample_paths))
29      return sum(sample_path_lengths)
```

E. Playing the GAME OF ATLAS

I have implemented the ATLAS game to be played with the computer along with helpful tournaments among strategies based on the above observations. I have plotted a heatmap to show the head-to-head strategy winnings. Also since there is some assymetry between the strategy that starts the game, I have kept all the $n \times n$ values where the i^{th} row and j^{th} columns means that the i^{th} strategy played the first move against the j^{th} strategy.

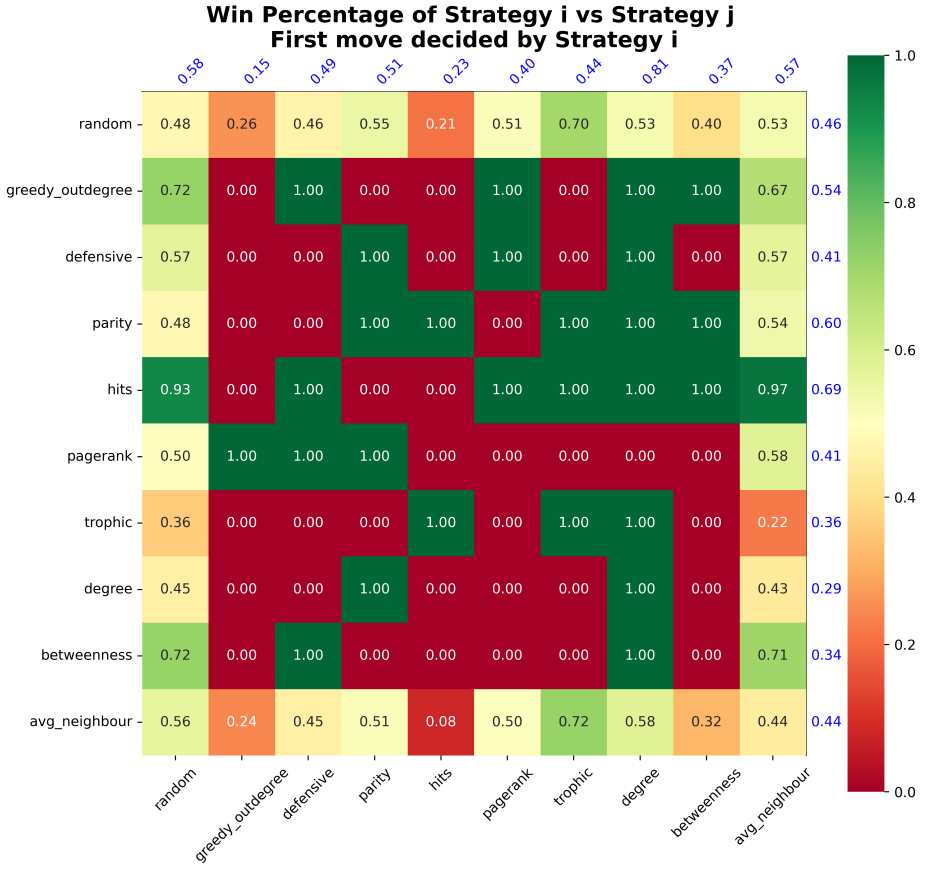

FIG. 3: The ATLAS game heatmap countries.

I observe that parity is a good approach on the cities and combined graph with lower clustering values signifying the existance of death trajectories opponents cannot escape. Whereas HITS advantage and greedy outdegree optimization also work well.

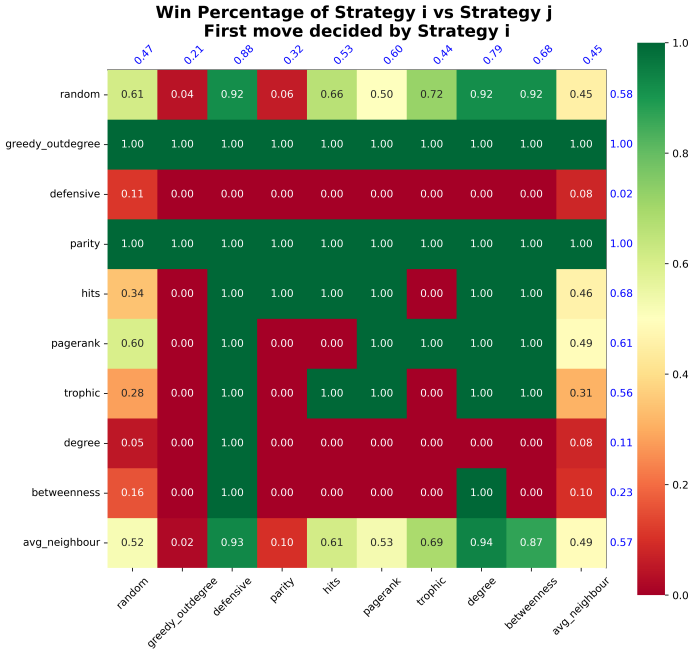

FIG. 4: The ATLAS game heatmap cities.

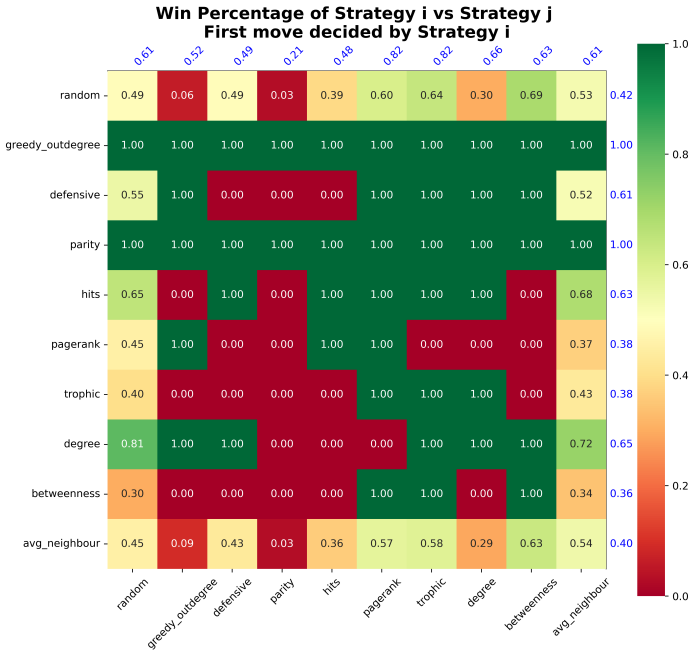

FIG. 5: The ATLAS game heatmap cities.

IV. TASK: COMMUNITY DETECTION

I performed community detection using 4 different methods:

1. Greedy Modularity Maximization
2. Lovain Community Finding Algo
3. Girvan-Newman Algo
4. K-Means clustering on node2vec vectors with p=1, q=0.1

The highest modularity was achieved with Louvain's algo.

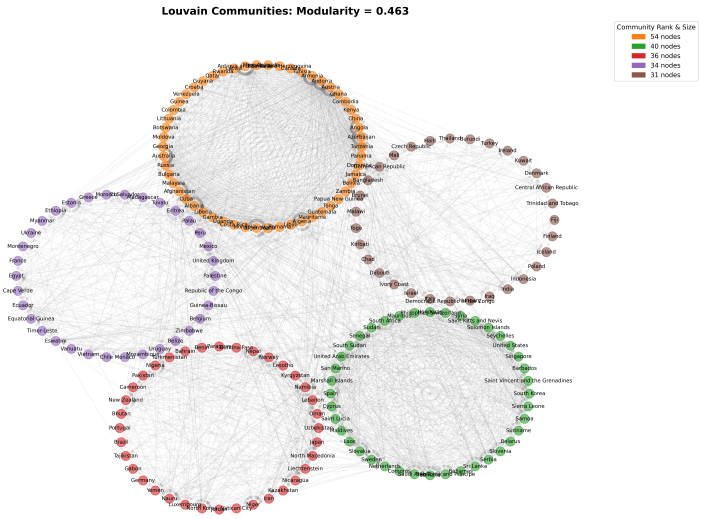

FIG. 6: Louvain Communities for Country Graph.

The communities obtained using greedy modularity and louvain's algo are not very interpretible. However, on a high level the countries that end with a common letter are in the same community.

The Girvan-Newman algo is an iterative algo removing the edge with highest betweenness centrality at every point. I have created an animation for the same. Please refer to the `code/community/animation.gif`. The communities here mostly have the countries that end with a common letter like 'a' or 's' with some other countries being a singleton node. I did 50 iterations and the highest modularity obtained was 0.423.

V. BONUS: LINK PREDICTION

A. Using Node2Vec

B. Using Graph Neural Networks (GNN)

VI. PAPER READING

The node2vec algorithm learns feature representations of nodes from graphs.

- Structural Connectivity: the actual anatomical links between neurons through synapses and fibre pathways. This requires invasive methods, which are not always feasible.
- Functional Connectivity: statistical or causal relationships between neurons measured as cross-correlations, coherence, or information flow.

We focus on studying the functional connectivity of the brain using network science and multifractal methods. The functional connectivity of the brain is usually studied by gathering EEG/MRI/f-MRI data that provides a non-invasive way to study neuron activation potentials. Also, it offers high temporal resolution to study the dynamical properties of brain activity.

Throughout the report, we present the results of our exploration based on a publicly available dataset of resting-state EEG recordings by Sockeel and colleagues. [Source]. It consists of the EEG recordings of twelve young, healthy participants with 62 channels sampled at a rate of 5 kHz.

VII. CORRELATION

Understanding the behaviour of a system usually involves understanding the system and its interactions with its surroundings. Thus, we try to find the similarity between the various interacting components. In terms of EEG data, we try to find out how one channel of EEG is related to the other channels. Mathematically, similarity can be quantified using correlation measures. Hence, we compute the pair-wise correlation between the 62 EEG channels. A few well-known correlation measures exist, like the Pearson correlation and Spearman's rank correlation. We will examine Pearson correlation and a non-linear correlation measure called Synchronization Likelihood.

A. Pearson Correlation

It is a well-known linear correlation measure used widely in statistical studies. Given two samples of data $\{x_i\}$ and $\{y_i\}$, the Pearson correlation between them is defined as:

$$\rho = \frac{\Sigma\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sqrt{\Sigma\left(x_i - \bar{x}\right)^2}\sqrt{\Sigma\left(y_i - \bar{y}\right)^2}}$$

$$\begin{aligned}
\rho &= correlation\ coefficient \\
x_i &= values\ of\ the\ x\ variable\ in\ a\ sample \\
\bar{x} &= mean\ of\ the\ values\ of\ the\ x\ variable \\
y_i &= values\ of\ the\ y\ variable\ in\ a\ sample \\
\bar{y} &= mean\ of\ the\ values\ of\ the\ y\ variable
\end{aligned}$$

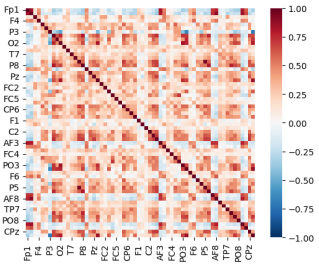

FIG. 7: Heatmap of pair-wise correlation values of the 62 channel resting state EEG data.

The Pearson correlation appears in the formula for linear regression where we assume the dependent variable y to be related to x as $y_i = \beta_0 + \beta_1 x_i + \epsilon$. Here, β_0 and β_1 are regression coefficients, and ϵ is the error term. The coefficient β_1 is computed as:

$$\frac{cov(x,y)}{var(x)} = \frac{\rho(x,y) \times std(x) \times std(y)}{var(x)}$$
$$= \frac{\rho(x,y) \times std(y)}{std(x)}$$

B. Synchronization Likelihood

Since the complex brain activity is expected to be nonlinear, we need a correlation measure that can capture the nonlinear dynamics of the process. Hence, we define Synchronization Likelihood, a phase space reconstruction-based method for detecting nonlinear correlations.

- Suppose we have M time series recorded simultaneously denoted by x_i^k where k denotes the channel number and i is the time stamp of observation.
- For each channel k, create embedded vectors at each time stamp i denoted by X_i^k defined by:

  $$X_i^k = (x_i^k, x_{i+l}^k, x_{i+2l}^k, ..., x_{i+(m-1)l}^k)$$

  where l is the time lag and m is the dimension of the embedded vector.
- Define a window by parameters w_1 and w_2 where $w_2 > w_1$.
- For each time series, we define a probability for a vector at timestamp i to be closer to other vectors in the window centred around i than some specified distance.
- We find a specific distance such that this probability for each time series at each timestamp equals some constant reference probability.
- Then, for each timestamp pair (i, j) satisfying $w_1 < |i - j| < w_2$, we calculate the number of time series where Xi is closer to X_j than the specific distance for timestamp i for that time series.
- We define the synchronization likelihood for each time series at each timestamp pair to be the count calculated above normalized by M - 1 if the distance between the embedded vectors of the time stamp pair are closer than the specific distance for the time series. Otherwise the synchronization for the timestamp pair for that time series is zero.
- Now, we can find the synchronization of a time series to other time series at each timestamp i by averaging across the j dimension.
- Similarly we can average over the i or k dimension to get a modified synchronization value.

VIII. MULTIFRACTAL ANALYSIS OF EEG

A. DFA

B. MF-DFA

C. Fish Tails

IX. CONCLUSIONS AND CONTINUING WORK

---

[1] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 3, pages 361–362. AAAI Press, 2009.

[2] GeoNames. Geonames - all cities with a population ¿ 1000. https://hub.huwise.com/explore/assets/geonames-all-cities-with-a-population-1000/, 2025. [Online; accessed 17-December-2025].

[3] Espen AF Ihlen. Introduction to multifractal detrended fluctuation analysis in matlab. *Frontiers in physiology*, 3:141, 2012.

[4] Jan W. Kantelhardt, Stephan A. Zschiegner, Eva Koscielny-Bunde, Shlomo Havlin, Armin Bunde, and H.Eugene Stanley. Multifractal detrended fluctuation analysis of nonstationary time series. *Physica A: Statistical Mechanics and its Applications*, 316(1–4):87–114, December 2002.

[5] T. Montez, K. Linkenkaer-Hansen, B.W. van Dijk, and C.J. Stam. Synchronization likelihood with explicit time-frequency priors. *NeuroImage*, 33(4):1117–1125, 2006.

[6] Nisarg Suthar. Atlas.
urlhttps://github.com/itsNisarg/Atlas, 2025. Accessed: 16-December-2025.

[7] Frigyes Samuel Racz, Orestis Stylianou, Peter Mukli, and Andras Eke. Multifractal dynamic functional connectivity in the resting-state brain. *Frontiers in Physiology*, 9:1704, 2018.

[8] Frigyes Samuel Racz, Orestis Stylianou, Peter Mukli, and Andras Eke. Multifractal and entropy analysis of resting-state electroencephalography reveals spatial organization in local dynamic functional connectivity. *Scientific Reports*, 9(1):13474, 2019.

[9] Steven J. Schiff, Paul So, Taeun Chang, Robert E. Burke, and Tim Sauer. Detecting dynamical interdependence and generalized synchrony through mutual prediction in a neural ensemble. *Phys. Rev. E*, 54:6708–6724, Dec 1996.

[10] C.J. Stam and B.W. van Dijk. Synchronization likelihood: an unbiased measure of generalized synchronization in multivariate data sets. *Physica D: Nonlinear Phenomena*, 163(3):236–251, 2002.

[11] Wikipedia contributors. List of countries and dependencies by population. — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population, 2025. [Online; accessed 17-December-2025].