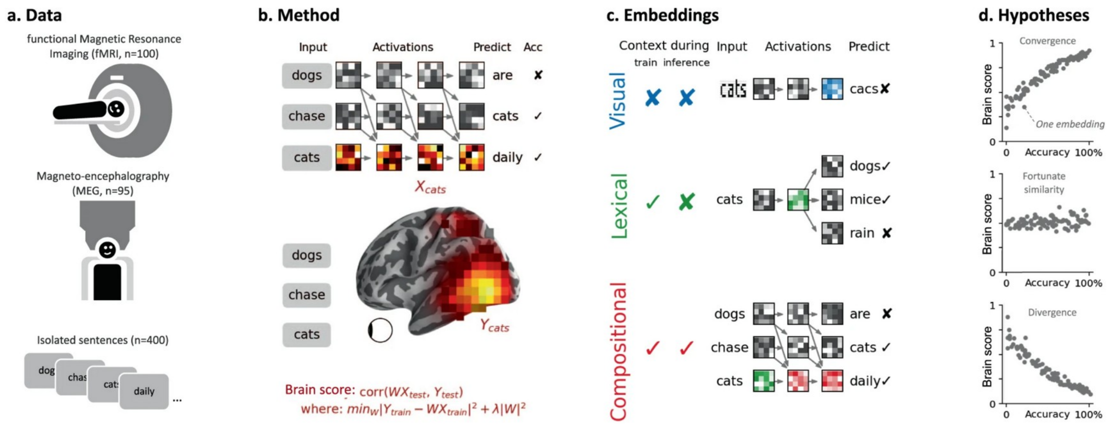
🔹 Project 1: Aphasia language model. Complex project (3-5 people)

Investigate whether there is a common space (with autoencoders or) between corrupted LLMs and brain signal of stroke subjects.



https://www.nature.com/articles/s42003-022-03036-1

Goal: use AphasiaBank recordings (as **stimuli** and/or retrieval database) and align neural responses from stroke patients (fMRI and/or EEG) to internal representations of an LLM (e.g., GPT-2/GPT-Neo/GPT-style transformer), following the *Caucheteux & King* approach (encode brain↔LLM representations with linear models & layer-wise comparisons). The pipeline has two linked parts:

A. **Language side** — feed the *same stimuli text/audio* into a language model and extract time-locked activations (layer-by-layer hidden states).
B. **Neuro side** — preprocess fMRI / EEG from stroke patients during the *same stimuli* presentation, then fit encoding/decoding models between neural signals and LLM activations.

Data: pre-post recovery fMRI: https://openneuro.org/datasets/ds003999/versions/1.0.2

Word transcript of stroke subjects: https://talkbank.org/aphasia/

Practically,

1. choose an available LLMs but not the API-end point for chat, find checkpoints for embeddings. This is the embeddings of healthy wordings (see the reference paper).

2. Download broken text patterns from the AphasiaBank, and create further embeddings (word-to-vec, langchain embeddings for LLMs… etc).

3. Create also embeddings of healthy people fMRI (or stroke subjects with recovered aphasia post-therapy) and people with aphasia (non-recovered or pre-therapy) and see if the embeddings are relatable.

C. Evaluate explained variance, layer-wise alignment, temporal profiles (for EEG), and control baselines (non-contextual embeddings, shuffled stimuli). This mirrors Caucheteux & King. Nature

The Caucheteux & King recipe

1. **Encoding model**: linear ridge regression that predicts neural responses from LLM activations (per layer). For fMRI: predict voxel/ROI time series (after HRF) from LLM features. For EEG: fit TRF-style linear model across time lags.
2. **Decoding model**: reverse mapping; predict LLM embedding from neural data.
3. **Layer-wise analysis**: for each model layer, compute prediction performance (cross-validated correlation or explained variance). This reveals which LLM layer best matches which brain region / latency. Nature
4. **Significance & controls**:
    - Permutation tests (shuffle stimulus–brain pairing), compare with static embeddings (word2vec), phonetic baselines, and with temporally shifted regressors.
    - Correct for multiple comparisons over voxels/ROIs.

Suggested Evaluation metrics (what to report)

- Cross-validated **Pearson correlation** between predicted and observed neural signals (or variance explained, R^2).
- Compare best-matching LLM layer per ROI/voxel.
- Statistical significance (permutation).
- For EEG, present latency-resolved TRFs and time–frequency analyses if needed.
- For stroke, stratify by lesion location / aphasia severity and show whether alignment changes with impairment.

🔹 **Project 2: Autoencoder-based dynamic functional connectivity (3-4 people)**

- **Goal:** Learn latent representations of **time-varying functional connectivity** from fMRI or EEG.
- **Task for students:**
    - Compute sliding-window connectivity matrices (EEG coherence, PLV, or fMRI correlations).
    - Use a **graph autoencoder** to compress these matrices.
    - Examine whether latent trajectories reflect cognitive states or clinical conditions.
- **Focus:** The novelty would be in showing how autoencoders reveal **low-dimensional dynamics of network reconfigurations** during tasks or pathologies.

🔹 **Project 3: Spatiotemporal trajectory forecasting with reservoir computing (2-4 people)**

- **Goal:** Predict short-term future evolution of brain signals (like MEG or EEG) from past states.
- **Task for students:**
  - Apply ESNs to forecast neural oscillations or traveling wave dynamics.
  - Quantify forecasting error across frequency bands.
  - Compare with LSTM/GRU performance.
- **Focus:** Demonstrating that reservoirs can capture **nonlinear wave-like brain activity** efficiently is of interest to computational neuroscience.
- Use **open datasets** (EEG: CHB-MIT, TUH, BCI Competition; fMRI: HCP, OpenNeuro).

🔹 **Project 4: Cross-subject latent alignment with autoencoders (2-4 people)**

- **Goal:** Learn a **shared latent space** of brain dynamics across individuals.
- **Task for students:**
  - Train autoencoders on multiple subjects' EEG or fMRI time series.
  - Use a shared encoder but subject-specific decoders (or adversarial alignment).
  - Test whether the latent space improves cross-subject classification of tasks or pathology.
- **Focus:** This links to the current trend of **representation alignment** in neuroscience, similar to Dabaglia et al.'s work on aligning latent representations of neural activity,
- Use **open datasets** (EEG: CHB-MIT, TUH, BCI Competition; fMRI: HCP, OpenNeuro).

Projects for either Machine Learning for neuroscience, or Bio-inspired AI

🔹 **Project 5 : Simulating Neural Dynamics with Recurrent Neural Networks (RNNs) (2-4 people)**
Train an RNN on a cognitive task (e.g., a decision-making task like the "Random Dot Motion" task). Analyze the activity of the artificial neurons in the RNN and compare it to

the activity of real neurons recorded from the prefrontal cortex of animals performing the same task. Skills: RNNs (LSTMs/GRUs), dynamical systems, data analysis.
Goal: Gain insights into how biological networks might implement cognitive functions by studying artificial analogues.

References

- Mante, Sussillo, Shenoy & Newsome (2013) - "Context-dependent computation by recurrent dynamics in prefrontal cortex."

- Sussillo, Churchland & Pandarinath (2015) - "A neural network that finds a naturalistic solution for the production of muscle activity."

- Yang & Wang (2020) - "Emergence of stable and structured receptive fields in a random neural network."

Successful Project Could Involve:

1.Downloading a public neural dataset from a decision-making task: https://crcns.org/data-sets/vc
2.Training an LSTM or GRU (or a more biologically plausible RNN) to perform the same task from the sensory inputs.
3.Using PCA to visualize and compare the neural trajectories of the biological data and the RNN.
4.Drawing conclusions about whether the artificial network found a similar solution to the biological one.

Further suggestions:
Idea: Explain Neural Variability. Instead of just comparing averages, analyze trial-to-trial variability. Can the RNN model explain why neural activity is sometimes different even for identical stimuli? Does the RNN's internal noise model match the brain's?
Idea: Test a Specific Hypothesis. Perhaps there's a debate in the literature about how a particular brain region solves a task. Your RNN could generate a testable prediction. For example: "Our model predicts that lesioning neurons with property X will specifically impair decision accuracy under high uncertainty, but not under low uncertainty." Then, you could see if this holds in the biological data.
Idea: Compare Architectures. Don't just use a standard LSTM. Train different RNN types (e.g., LSTMs vs. GRUs vs. more biologically plausible networks with Dale's Law). Systematically compare which architecture's dynamics most closely match the brain's and hypothesize why. This is a novel contribution.
Idea: Develop a New Analysis Technique. The standard is PCA. Could you develop or apply a more sophisticated dynamical systems analysis tool (e.g., based on Riemannian geometry or topological data analysis) that reveals something about the neural trajectories that PCA misses? The novel contribution is the method itself.

🔹 Project 6: **Reservoir Computing for Brain-Computer Interface (BCI) Decoding (2-4 people)**

Use Reservoir Computing (RC) as a powerful, low-cost method to decode intended movements from neural signals (EEG, ECoG, or spike trains).
RC is inherently good at processing time-series data, which is exactly what neural signals are. Moreover, it has l**ow Training Cost:** Only the readout layer needs to be trained, making it very fast compared to training a full RNN with backpropagation. This is crucial for adaptive BCIs.

References

"Brain-Computer Interface using Echo State Networks" by Buteneers et al. (2008, 2009).

Further suggestions:
Idea: Most BCIs are synchronous (the system tells you when to imagine a movement). An asynchronous BCI (where the user can initiate a movement at any time) is more practical but much harder, as it requires distinguishing movement intention from idle states.
Idea: Can we leverage the temporal memory of Reservoir Computing for robust asynchronous BCI control?

Successful Project Could Involve:

Downloading a public neural dataset from a decision-making task:

- BNCI Horizon 2020 Dataset 004 (Motor Imagery, Self-Paced), Way EEG HQ Motor Imagery Dataset, Berlin Brain-Computer Interface (BBCI) Datasets

Use the data to train an RC classifier

https://www.nature.com/articles/s41586-025-09544-4