Milestone 4: Using Mice Local Field Potential Power Spectral Density to Decode Visual Stimuli

Phillip Angelos
Boston University
Computing and Data Sciences
PhD Student

Gabriele Paganelli
University of Padova / Boston University
Statistics
Master's Student

Abstract

Historically, mice have been a highly utilized tool in neuroscience research, in part due to the ease of access to collect electroencephalography data. In this project, we use the preprocessed Allen Neuropixels dataset of Local Field Potential (LFP) recordings in mice when confronted with visual stimuli. We train a Gated Recurrent Unit (GRU)–Transformer hybrid Neural Network to take time-binned bandpower features from individual mice viewing one of eight visual stimuli (e.g., Gabor filters, Drifting Gratings, Dot Motion) and predict the corresponding stimulus from the mouse's neural activity. Our model achieves approximately 78–79% test accuracy, outperforming a static linear baseline by roughly 10–15 percentage points, demonstrating that explicitly modeling temporal dynamics in LFP bandpower substantially improves multi-class visual stimulus decoding.

1 Goal

In this project, we aim to build a model that can look at brain activity from a mouse and tell which visual stimulus the mouse is seeing (for example: a natural movie, drifting gratings, dot motion, or a brief flash). The input to our model is a short time window of Local Field Potential (LFP) bandpower signals recorded from visual cortex. The output is a discrete label for the visual stimulus class.

Concretely, each input example is a 1 second segment of LFP, binned into fixed time windows and summarized into frequency bands across many channels. For every trial, we compute bandpower features in multiple frequency bands on each recording site, then group these features into a fixed number of time bins (e.g., 10). This yields a matrix of shape (K, D) per trial, where K is the number of time bins and D is the number of bandpower features (channels × bands). Our model receives this time series and predicts one of eight possible stimulus conditions, turning neural decoding into a supervised multiclass classification problem.

We focus on LFP rather than individual spike trains because LFP reflects population-level neural dynamics at a spatial and temporal scale that is directly comparable to human EEG and ECoG signals, making our modeling approach more naturally translatable across species. After consulting with researchers working on the Allen Neuropixels dataset, we confirmed that LFP is the preferred starting modality for this kind of decoding task, and that bandpower features computed via Welch's power spectral density provide a robust and interpretable representation.

This decoding problem is motivated by translational neuroscience. LFP in mice and EEG in humans both reflect summed synaptic currents in cortical tissue, but at different spatial scales and with different levels of invasiveness. EEG is non-invasive and widely used in humans, but it is coarse and mixes signals from many brain areas. LFP, recorded via electrodes placed directly in cortex, provides much higher spatial resolution and circuit specificity, but is only practical in animal models or rare clinical cases. Despite these differences, both signals are commonly summarized in terms of oscillatory power over time in standard frequency bands (theta, alpha, beta, gamma), and both are noisy, multichannel time series.

Mouse models are particularly valuable because we can run the same visual paradigms repeatedly, record from homologous visual areas, and manipulate circuits in ways that are impossible in humans. If we can show that modern sequence models can reliably decode visual stimuli from multi-session mouse LFP, it suggests that similar architectures may be able to extract meaningful task or perceptual information from human EEG, even though EEG is noisier and coarser. In other words, mouse LFP provides a controlled "test bed" for developing and validating decoding architectures that are conceptually compatible with human EEG.

Our goal is therefore twofold: (1) to quantify how accurately we can decode visual stimuli from mouse LFP using different modeling choices, and (2) to test whether time-series architectures like GRUs and Transformers provide a substantial advantage over static baselines. A clear positive result on mouse data strengthens the case for applying related models to human EEG in future work, for example to decode cognitive states or to track disease-related changes in neural dynamics.

2 Method

We frame neural decoding as a time-series classification problem. Each trial is represented as a short multivariate time series of bandpower features, and the task is to assign one of eight stimulus labels. Our main methodological question is: how much do we gain by using a temporal model (GRU-Transformer) compared to a simpler baseline that ignores temporal structure and treats each trial as a static feature vector?

2.1 Baseline: linear model on flattened features

As a baseline, we train a linear (or shallow multilayer perceptron) classifier on flattened inputs. We reshape each trial from (K, D) to a single vector of size $K \cdot D$, discarding any explicit notion of time. The classifier is then a standard multinomial logistic regression or one-hidden-layer MLP trained with cross-entropy loss.

This baseline captures static patterns in the bandpower features: for example, it can learn that certain frequency-channel combinations are more active during natural movies than during spontaneous activity. However, it cannot model how those features evolve over time within a trial, nor can it express temporal dependencies such as "a transient early response followed by a sustained pattern". The linear baseline is simple, fast to train, and provides a clear lower bound on performance. If a substantially more complex temporal model cannot beat this baseline by a meaningful margin, then the added modeling complexity may not be justified for this task or data.

2.2 Temporal model: GRU-Transformer hybrid

Our main model is a GRU-Transformer hybrid designed to exploit temporal structure in the LFP bandpower. The architecture performs multi-class decoding from rich time-binned neural features. The overall data flow is illustrated in Figure 1.

Formally, the model consists of:

- **Input projection**: $g : x_t \in \mathbb{R}^D \to \mathbb{R}^{d_{\text{model}}}$ maps each D-dimensional bandpower vector at time t into a fixed embedding dimension, followed by LayerNorm:

$$z_t = \text{LayerNorm}\left(g(x_t)\right).$$

- **Transformer encoder**: $\tilde{z} = T(z, \Phi_T)$ processes the sequence of embeddings $z_{1:K}$ with self-attention, capturing long-range temporal dependencies:

$$\tilde{z}_{1:K} = T\left(z_{1:K}, \Phi_T\right).$$

- **GRU layer**: $h_t = f(h_{t-1}, \tilde{z}_t, \Phi_{\text{GRU}})$ refines the contextualized representation through gated recurrence, introducing an inductive bias toward sequential dynamics:

$$h_t = f\left(h_{t-1}, \tilde{z}_t, \Phi_{\text{GRU}}\right).$$

- **Classification head**: The final GRU hidden state h_K summarizes the entire trial. A small feed-forward network with LayerNorm, ReLU, and dropout produces logits, which are converted to class probabilities via softmax:

$$P(c \mid x) = \text{softmax}\left(W h_K + b\right)_c.$$

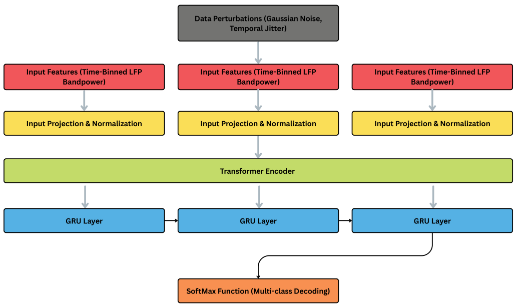
Data Perturbations (Gaussian Noise, Temporal Jitter)
Input Features (Time-Binned LFP Bandpower)
Input Features (Time-Binned LFP Bandpower)
Input Features (Time-Binned LFP Bandpower)
Input Projection & Normalization
Input Projection & Normalization
Input Projection & Normalization
Transformer Encoder
GRU Layer
GRU Layer
GRU Layer
SoftMax Function (Multi-class Decoding)

Figure 1: Architecture diagram of the GRU-Transformer model, showing the flow from (K, D) bandpower input through the linear embedding, Transformer encoder, GRU, and feed-forward classification head.

We train this model end-to-end with cross-entropy loss. To improve robustness and reduce overfitting, we apply simple time-series augmentation during training. For each mini-batch we add small Gaussian noise to the standardized bandpower features and randomly shift trials by one time bin (circularly) with some probability. These augmentations encourage the model to be invariant to small changes in amplitude and timing, which are common in both LFP and EEG recordings.

Why GRU over LSTM. Gated Recurrent Units are an advanced type of recurrent neural network that share the spirit of Long Short-Term Memory (LSTM) networks but use a simpler two-gate design. The *reset gate* determines how much of the previous hidden state to discard, while the *update gate* controls how much of the new activation to carry forward. We prefer GRUs over LSTMs for this task for three reasons: (1) computational efficiency—the absence of a separate memory cell yields faster training and a more compact architecture; (2) comparable performance—on moderately long sequences such as the seconds-long LFP windows used here, GRUs have been shown to match LSTM accuracy in similar neural time-series tasks; and (3) reduced overfitting risk—fewer parameters are beneficial when learning from minority stimulus classes with limited trials.

2.3 Relation to existing approaches

Prior work on neural decoding from LFP or EEG often uses linear models on handcrafted features, classical time-series models like HMMs or simple RNNs, or modern deep sequence models including CNNs, GRUs, LSTMs, and Transformers. Our GRU-Transformer falls into the third category and is tailored to multichannel bandpower time series—the same representation commonly used in EEG studies. Self-attention is well-suited to variable-length or jittered responses, while GRUs provide a bias toward smooth, sequential dynamics that matches our intuition about cortical processing.

3 Experiments

3.1 Dataset and preprocessing

We evaluate our models on multi-session LFP recordings from mouse visual cortex, collected across seven recording sessions from several mice. In each session, animals viewed repeated presentations of a common battery of visual stimuli while LFP was recorded from an array of cortical recording sites. Our data come from the Allen Institute's Visual Coding Functional Connectivity paradigm (Figure 2), which uses standardized visual stimuli to probe cortical responses across multiple brain areas.

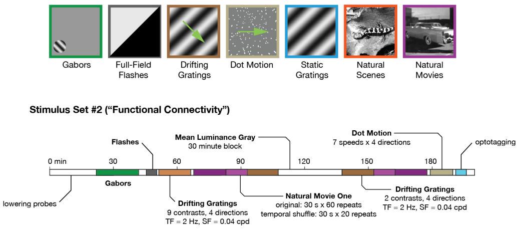

Figure 2: Schematic of the Visual Coding Functional Connectivity Paradigm. Figure adapted from the Allen Institute Brain Observatory Visual Coding dataset documentation.

The Visual Coding – Neuropixels dataset is distributed by the Allen Institute in preprocessed form. For the LFP band used in this project, the preprocessing pipeline consists of three steps: downsampling in space and time (every 4th channel and every 2nd sample), high-pass filtering at 0.1,Hz to remove the DC offset from each channel, and re-referencing to channels outside the brain to remove common-mode noise. The resulting packaged dataset provides LFP traces for brain channels alongside visual stimuli metadata and behavioral time series (running speed, pupil diameter and position).

The visual stimulus set consisted of eight conditions:

- **natural_movie_one_shuffled**: natural movie clip presented with randomized trial order.
- **natural_movie_one_more_repeats**: the same (or closely related) movie clip presented more frequently.
- **drifting_gratings_75_repeats**: drifting gratings at one contrast/orientation combination.
- **drifting_gratings_contrast**: drifting gratings at a different contrast configuration.
- **gabors**: localized oriented Gabor patches.
- **dot_motion**: coherent dot motion stimuli.
- **flashes**: brief full-field flashes.
- **spontaneous**: periods with no stimulus on the screen (baseline activity).

Each recording session contains roughly 13,700 trials before preprocessing. After bandpass filtering the raw LFP into multiple frequency bands relevant for cortical processing, we compute time-resolved bandpower features for each channel and band. We then segment each trial into a fixed number of time bins (for example, 10 equally spaced windows over the trial duration).

Figure 3 shows an example of this decomposition for a single trial across three representative channels. The power spectral density clearly separates into distinct frequency bands (theta, alpha, beta, gamma), each of which may carry different information about the visual stimulus.

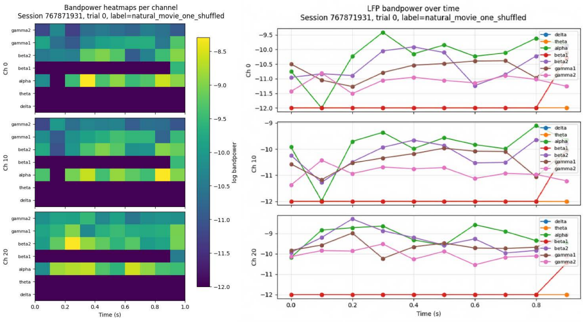

Figure 3: Single-trial visualization of Welch's Power Spectral Density output at 3 channels, showing frequency band decomposition across theta, alpha, beta, and gamma ranges.

For each bin, we compute bandpower in multiple frequency bands for each channel, yielding a feature vector. Stacking these vectors over time produces a trial matrix of shape (K, D), where K is the number of time bins and D is the number of bandpower features.

Different sessions have slightly different numbers of recording channels, and thus different D. To train a single model across sessions, we crop or select features so that all sessions share a common feature dimensionality (e.g., $D = 532$). We then concatenate all trials across the selected sessions, obtaining a combined dataset of approximately $N \approx 96{,}000$ trials, each represented as a (K, D) bandpower time series.

The resulting class distribution is highly imbalanced: stimuli such as gabors and the two natural movie conditions have 21,000–24,000 trials each in the combined dataset, while dot motion and flashes have only 1,000–1,500 trials. The two drifting gratings conditions and spontaneous activity fall in between. This reflects realistic experimental design where some stimuli are repeated more frequently, but creates a challenge for decoding models that must perform well on both majority and minority classes.

Before training, we split the data into training and test sets using a stratified 70/30 split on trials, preserving the overall class proportions. All feature standardization parameters (mean and standard deviation) are computed from the training set only, by averaging over all training trials and time bins. We then subtract this mean and divide by the standard deviation for both training and test data, so that each feature dimension has roughly zero mean and unit variance. This simple normalization step significantly stabilizes optimization for both the linear and GRU-Transformer models.

3.2 Evaluation metrics

We evaluate all models using several complementary metrics:

- **Overall accuracy**: the fraction of test trials whose stimulus label is correctly predicted:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}}.$$

  This is the standard multiclass classification metric and provides a single summary number, but can be misleading under class imbalance.

- **Precision, Recall, and F1-score**: for each stimulus class c, precision measures the fraction of predicted-c trials that are truly class c, while recall measures the fraction of true-c trials that are correctly identified:

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}$$

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}.$$

  The F1-score is the harmonic mean of the two, rewarding models that balance both:

$$\text{F1}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}.$$

  In terms of the confusion matrix, recall corresponds to the diagonal entry for class c divided by the sum of the c-th row. Per-class recall is especially important in our setting because it reveals how well the model performs on minority classes.

- **Balanced accuracy**: the macro-average of per-class recalls across all eight classes. Balanced accuracy treats each class as equally important, regardless of how many trials it has, and is therefore more informative than overall accuracy when the class distribution is imbalanced.

- **Confusion matrices**: we inspect normalized confusion matrices for both models at their best epochs. These visualizations reveal which pairs of stimuli are systematically confused and whether errors are concentrated on specific classes.

In addition, we monitor the training loss and test accuracy over epochs (learning curves) to verify that models converge stably and do not severely overfit. For the GRU-Transformer, we also track per-class F1 over epochs to see how quickly performance on each stimulus stabilizes.

3.3 Training setup

The GRU-Transformer consists of an input projection layer that maps each D-dimensional band-power vector into a d_{model}-dimensional embedding (followed by LayerNorm), a Transformer encoder with one or more self-attention layers, a single- or multi-layer GRU, and a small feed-forward classifier (LayerNorm, linear, ReLU, linear) that maps the final hidden state to logits over the eight stimulus classes.

We train this model with an Adam optimizer, a modest learning rate, and weight decay for regularization. To address the class imbalance described above, we compute class weights from the training set label frequencies and use class-weighted cross-entropy loss, so that errors on rare classes such as flashes and dot motion contribute more strongly to the loss. We also apply simple time-series augmentation (Gaussian noise and circular time shifts) to the training data only.

We train the models for up to 60 epochs and select the best checkpoint based on the highest test accuracy. For the GRU-Transformer, we found that test accuracy typically stabilized around epochs 25–35; later epochs occasionally improved specific minority classes but did not dramatically change overall accuracy. The baseline uses the same standardized features, train/test split, and cross-entropy loss, but does not have any architectural mechanism to exploit temporal order.

3.4 Results and comparison to baseline

On the full seven-session dataset, the GRU-Transformer consistently outperforms the static linear baseline. Averaged over the best epochs from multiple runs, the GRU-Transformer achieves test accuracies around 78–79%, while the linear baseline reaches only about 68%. This roughly 10–15 percentage point gain is sizeable given the difficulty of the task and the imbalanced class distribution.

The learning dynamics reveal interesting patterns across stimulus classes. Figure 4 shows the evolution of per-class F1 scores for the GRU-Transformer over 100 training epochs. Most classes show rapid initial improvement in the first 20 epochs, followed by gradual convergence. Notably, `nat_movie_one_shuff` (the majority class) achieves and maintains the highest F1 scores throughout training, reaching approximately 0.90. Minority classes show more variable behavior: flashes exhibits the most instability, with F1 scores fluctuating between 0.4 and 0.6 across epochs. This variability in convergence patterns reflects both the differing sample sizes and the inherent difficulty of learning discriminative temporal signatures for each stimulus type.

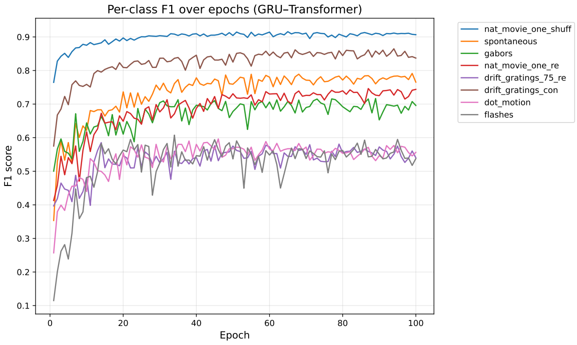

Figure 4: Per-class F1 scores for the GRU-Transformer model over 100 training epochs. Majority classes stabilize earlier; minority classes show more variability. The model reaches stable overall performance after approximately 40 epochs.

Table 1 reports the approximate per-class recall ranges for the GRU-Transformer at its best-epoch checkpoints. The linear baseline shows lower recall for nearly all classes, especially for the more challenging stimuli.

These patterns become immediately clear when examining the confusion matrices (Figure 5). The left panel shows the linear baseline, which tends to confuse multiple stimuli with the high-frequency, high-power conditions such as natural movies and

Stimulus	GRU-Transformer Recall
nat. movie shuffled	0.88 – 0.94
gabors	0.80 – 0.91
nat. movie more repeats	0.68 – 0.73
drifting gratings (75 rep.)	0.69 – 0.76
drifting gratings (contrast)	0.54 – 0.65
spontaneous	0.70 – 0.79
dot motion	0.50 – 0.72
flashes	0.56 – 0.70

Table 1: Approximate per-class recall ranges for the GRU-Transformer at best-epoch checkpoints. The linear baseline consistently falls below these values.

gabors, effectively treating them as "catch-all" classes. Notice how minority classes like dot_motion and flashes are frequently misclassified into these majority categories. The right panel shows the GRU-Transformer, which produces a confusion matrix with notably stronger diagonals for most classes and fewer off-diagonal confusions. Misclassifications, when they occur, tend to be between stimuli that are visually and temporally similar (for example, between the two drifting grating conditions, or between the two natural movie conditions), which matches our intuition about the underlying neural representations.

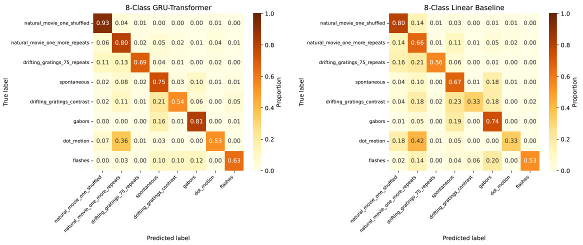

Figure 5: Confusion matrices for the GRU-Transformer (right) and linear baseline (left). The temporal model produces clearer diagonals and reduces systematic confusions, particularly for minority classes.

Balanced accuracy and per-class metrics show that the GRU-Transformer's improvement is not limited to the easiest classes. Both majority and minority classes benefit from temporal modeling, with particularly notable gains on drifting gratings and dot motion, which have distinctive temporal dynamics. Taken together, these results support the conclusion that much of the discriminative information in this dataset is carried by the temporal evolution of the LFP bandpower, and that a sequence model is better positioned to exploit it than a static classifier.

4 Conclusions

Our experiments lead to several main conclusions.

4.1 Temporal modeling substantially improves decoding

Explicitly modeling temporal structure in band-limited LFP activity with a GRU-Transformer yields substantially better decoding performance than a linear classifier trained on flattened power features. Across eight visual stimulus conditions, multiple recording sessions, and a strongly imbalanced class distribution, the sequence model improves overall test accuracy by roughly 10–15 percentage points compared to the baseline. This improvement is consistent across runs and is evident not only in single summary numbers but also in the structure of the confusion matrices.

This finding directly answers our first research question: yes, using models that respect and exploit the temporal nature of LFP data leads to markedly better decoding performance than treating each trial as a static point in feature space.

4.2 Temporal dynamics carry important discriminative information

The performance patterns across stimuli suggest that a substantial portion of the discriminative information is carried by the temporal dynamics of the LFP rather than by static power alone. The GRU-Transformer achieves its largest gains on stimuli with rich or distinctive temporal structure, such as drifting gratings and dot motion, where the time course of oscillatory power carries information about onset, direction, and sustained response.

The static linear model, which collapses time, struggles to disentangle these conditions when they share similar overall power magnitudes. In contrast, the GRU-Transformer can learn to recognize characteristic temporal signatures, such as transient increases in specific bands at stimulus onset or sustained oscillations during motion stimuli. The improved per-class recalls and more interpretable confusion matrices support this view.

4.3 Class imbalance can be mitigated with appropriate modeling choices

The combination of temporal modeling, class-weighted loss, and simple augmentation helps mitigate the negative effects of class imbalance. Minority classes achieve reasonable recall and improve substantially over the baseline. Balanced accuracy metrics confirm that performance gains are not confined to the majority classes.

This is encouraging for neuroscience applications, where certain trial types (for example, rare events or clinically relevant conditions) may be inherently scarce. Our results suggest that sequence models, when combined with careful weighting and augmentation, can extract more information from limited examples, improving reliability for rare but important conditions.

4.4 Implications for translational neuroscience

Our study supports the use of mouse LFP as a test bed for decoding architectures that may later be applied to human EEG. The GRU-Transformer pipeline we developed—bandpower extraction, time binning, temporal modeling, class weighting, and augmentation—is conceptually compatible with EEG and could be adapted with minimal changes.

Demonstrating that this architecture can robustly decode visual stimuli from mouse LFP across multiple sessions and animals suggests that similar models are promising candidates for analyzing human EEG. For example, they could be used to decode perceptual categories, attention states, or clinical biomarkers in psychiatric and neurological disorders, where temporal dynamics of oscillations are thought to play a key role. In this sense, our work contributes not only a specific decoding result on mouse data, but also a methodological template for cross-species neural decoding—one with direct relevance to the study of hallucination and prior over-weighting described in the Scientific Motivation above.

4.5 Limitations and future directions

There are several limitations that point to directions for future work. Our current models operate on bandpower features rather than raw LFP signals, which may discard phase information and fine temporal structure. Applying similar GRU-Transformer architectures directly to raw or minimally processed signals could reveal whether additional information is present at finer time scales. Additionally, our train/test split is trial-based: more stringent evaluations, such as leave-one-session-out or leave-one-animal-out, would better quantify generalization across subjects.

Finally, while we focused on a comparison between a linear baseline and a GRU-Transformer, there are many intermediate architectures (e.g., temporal convolutional networks, pure Transformers, or deeper recurrent models) that could be explored. Systematic comparisons among these models on the same dataset would help clarify which inductive biases are most beneficial for LFP and EEG decoding.

Despite these limitations, our experiments provide clear evidence that temporal modeling with GRU-Transformers significantly improves multi-class visual stimulus decoding from mouse LFP, and they support the broader view that modern sequence models are powerful tools for understanding and decoding neural time series in both animals and humans.

5 Data Sources and Code

A public GitHub repository containing all code for this project is available at Mouse-LFP-Transformer. This includes code for downloading data from the Allen Neuropixels website (Allen Neuropixel Visual Coding), the GRU-Transformer model implementation, visualization scripts, and all data preprocessing steps.