# Extended Abstract

**Motivation** Noninvasive brain–computer interfaces (BCIs) that decode natural speech from neural recordings hold promise for restoring communication in motor-impaired individuals and for probing human language processing. Magnetoencephalography (MEG) offers millisecond-scale temporal resolution across the whole head, but its lower spatial precision and the complexity of spoken language representations limit decoding accuracy. Prior MEG studies have largely confined themselves to small vocabularies or phoneme classification, achieving only coarse reconstructions of imagined or heard speech. To move beyond these constraints, we investigate whether combining MEG-based neural decoders with the rich contextual priors of large language models (LLMs) can yield more coherent and accurate reconstructions of heard speech.

**Method** We propose a hybrid two-stage framework: first, we train deep recurrent neural network (RNN) decoders to map MEG time-windows (–0.5 s to 2 s around each word onset) onto probability distributions over a closed vocabulary of 6,975 tokens. Second, we feed the top-k candidate distributions for each of 15 sequential tokens into a Llama 3.2 3B model. We fine-tune this LLM via supervised instruction tuning (SFT) with a low-rank adaptor (LoRA) to learn proper parsing and tagging of decoder outputs, and then apply Online Direct Preference Optimization (DPO) to align its generation preferences with decoder accuracy.

**Implementation** Our dataset comprises over 10 hours of MEG recordings from three native English speakers listening to Sherlock Holmes audiobooks. Neural decoders were implemented in PyTorch, with architectures ranging from a two-layer MLP to a six-layer deep RNN and a bidirectional LSTM with convolutional front end. For LLM tuning, we generated 5,000 synthetic sequences by sampling from a unigram distribution augmented with simulated decoder noise, and trained the LoRA adaptor for up to 50 epochs with early stopping at 97

**Results** The deep RNN decoder achieved 5.92% word-level accuracy—over five hundred times chance—on held-out MEG data. Instruction tuning elevated the LLM's tagging consistency from 0% to 90% and boosted task accuracy on synthetic distributions to 63%. Online DPO further refined preferences, yielding 62% accuracy. Qualitative analysis showed the hybrid model's ability to resolve ambiguous top-k candidates in semantically coherent ways, while also revealing occasional "runaway" reasoning and bias toward high-frequency words.

**Discussion** These findings demonstrate that even modest neural decoding signals from noninvasive MEG can be substantially amplified by LLM priors, achieving fluency far beyond raw classifier accuracy. The performance gap between synthetic and real MEG-driven distributions suggests opportunities for tighter integration—such as adaptive vocabulary pruning or hierarchical classification—to better align LLM training with actual decoder noise characteristics. Addressing failure modes (e.g., grammatical overcorrections, reasoning drift) will require richer preference labelling and more robust fine-tuning objectives.

**Conclusion** We present a novel framework that merges MEG-based neural decoding with RL-tuned LLMs, achieving word-sequence coherence orders of magnitude above chance. While practical noninvasive speech BCIs remain aspirational, our hybrid approach establishes a viable pathway: improving decoder quality, refining LLM integration, and iteratively aligning language priors with neural signals promise continued gains toward real-time heard-speech reconstruction.

# Neural Decoding of Heard Speech Using RL-tuned LLMs

**Ricky Rojas**
Department of Data Science, Statistics
Stanford University
rickyro@stanford.edu

**Ramani Tyagi**
Department of Computer Science
Stanford University
ramanit@stanford.edu

## Abstract

Brain-computer interfaces for natural speech decoding face significant challenges in understanding heard speech from neural recordings. We present a novel hybrid approach that combines deep learning-based neural decoders with fine-tuned large language models to enhance heard speech decoding from magnetoencephalography (MEG) data. Using over 10 hours of MEG recordings from 3 subjects listening to naturalistic speech in the form of Sherlock Holmes audiobooks, we trained neural decoders to classify words from a vocabulary of 6,975 unique tokens. Our best-performing deep RNN achieved 5.92% accuracy, exceeding random chance (0.01%) but insufficient for practical applications. To address this limitation, we fine-tuned a Llama 3.2 3B large language model using simulated decoder outputs, incorporating instruction tuning and Online Direct Preference Optimization (DPO). The base language model initially achieved 0% task accuracy, but instruction tuning and DPO improved performance to 63% and 62% respectively.

## 1 Introduction

Decoding spoken language directly from brain activity is a long-standing goal of brain–computer interface (BCI) research, with the potential to restore communication for individuals with severe motor impairments and to deepen our understanding of human language processing. While early work has demonstrated proof-of-concept reconstructions of imagined or overt speech using invasive recordings Dash et al. (2020), noninvasive modalities such as magnetoencephalography (MEG) remain underexploited despite offering millisecond-level temporal resolution and whole-head coverage. MEG's noninvasiveness makes it an attractive platform for scalable BCIs, but its lower spatial resolution and the complex, distributed nature of speech representations pose significant decoding challenges.

Most prior MEG decoding studies have focused on coarse classification tasks—such as binary phoneme discrimination or classifying among a small vocabulary of tens of words—which do not capture the richness of natural language comprehension Yang et al. (2024). Furthermore, these approaches typically treat decoding as a stand-alone pattern recognition problem, without leveraging the powerful priors available from large language models (LLMs). Meanwhile, recent advances in deep language models have shown that instruction tuning and reinforcement-learning-based fine-tuning methods can dramatically improve generation quality and controllability for open-ended text tasks Gang et al. (2025); Guo et al. (2024). However, the application of such methods to neural decoding remains largely unexplored.

In this work, we introduce a hybrid framework that synergistically combines deep neural decoding of MEG signals with fine-tuned LLM reasoning to reconstruct heard speech from continuous, naturalistic stimuli. First, we train recurrent neural network (RNN) decoders to map MEG data segments, time-locked to word onsets, onto probability distributions over a closed vocabulary of 6,975 tokens.

Although our best deep RNN achieves an accuracy of only 5.92%—substantially above chance but insufficient for direct use—the resulting token probabilities carry valuable information about likely word identity. To exploit this signal, we present the top-k candidate distributions to a Llama 3.2 3B model, which we refine via two complementary fine-tuning stages: supervised instruction tuning (SFT) with a low-rank adaptor (LoRA) to learn proper tagging and output formatting, and Online Direct Preference Optimization (DPO) to align generation preferences with decoding accuracy.

Our contributions are threefold:

- We demonstrate that deep RNN-based MEG decoders can extract above-chance word probabilities from naturalistic speech listening, achieving 5.92% accuracy on a 6,975-token closed-set task.

- We develop a LoRA-augmented instruction fine-tuning pipeline that teaches the LLM to interpret neural decoder outputs and produce correctly formatted word guesses, raising tagging consistency from 0% to 90%.

- We apply Online DPO to further align the LLM's preferences with decoder accuracy, achieving task accuracy improvements from 0% (base) to 90% (SFT) and 62% (DPO) in simulated experiments.

By integrating neural decoding with LLM reasoning, our hybrid approach points toward a new direction for noninvasive speech BCIs that leverages state-of-the-art language modeling. Although practical deployment remains challenging given current decoding accuracies, this framework provides a proof of concept for augmenting noisy neural signals with powerful priors from LLMs—and lays the groundwork for future improvements as both decoding models and language models continue to advance.

## 2 Related Work

Auto-prediction— the phenomenon in which the human brain continuously anticipates upcoming words during spoken language comprehension— has been extensively documented in both behavioral and neural studies. This predictive mechanism is thought to facilitate rapid and efficient speech comprehension by enabling the brain to pre-activate likely lexical candidates, thereby reducing the computational burden of bottom-up auditory processing Goldstein et al. (2022); Caucheteux et al. (2023).

Parallel to these cognitive insights, significant progress has been made in neural decoding, particularly in the domain of imagined and covert speech. Recent work by Dash et al. (2020) demonstrated the feasibility of reconstructing spoken and imagined phrases using non-invasive magnetoencephalography (MEG), a relatively low-spatial-resolution technique. While MEG lacks the precision of intracranial recordings, its temporal resolution makes it particularly suitable for capturing dynamic, time-locked processes such as predictive language modeling.

However, a major limitation in current decoding paradigms is the focus on realized speech—words that are overtly spoken or imagined with explicit intent. The neural representation of predicted but unrealized speech remains relatively unexplored. This represents a critical gap, especially in light of growing evidence that the brain internally simulates upcoming words during passive listening without the need for explicit speech generation.

In this work, we seek to bridge these two domains by leveraging the logical reasoning skills and inherent linguistic understand to resolved predicted words from MEG recordings of subjects exposed to extended naturalistic speech. Offline RL has shown promise in learning complex, temporally extended representations from static datasets Levine et al. (2020), and we propose that its application to prediction-driven neural activity could offer new insights into the intersection of cognition, computation, and language. Our dataset consists of over 10 hours of MEG recordings from three participants, providing a rich resource to explore the temporal structure of linguistic prediction in the brain.

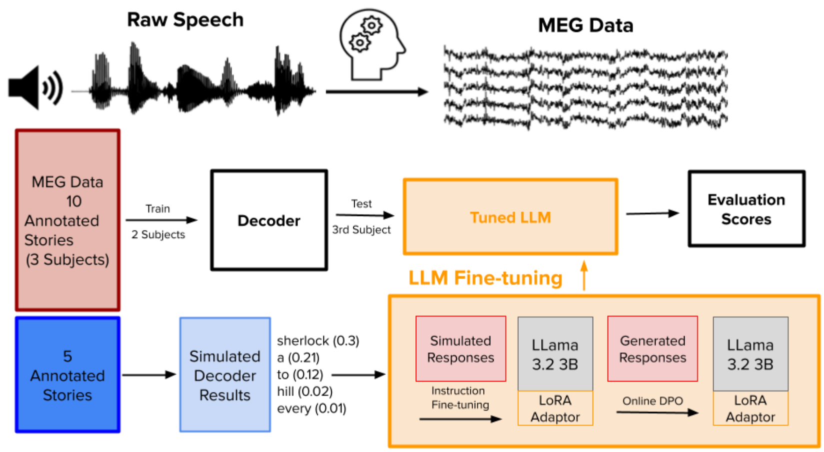


Figure 1: Method Overview.

# 3 Methods

### Overview

Our project works to enhance the accuracy of deep-learning MEG speech decoders by leveraging the lingustic and reasoning capabilities of large language models. As seen in Figure 1, we use magnetoencephalography (MEG) data of 2 subjects listening to Sherlock Holmes audio books to train a heard speech decoder. We then use a fine-tuned LLM to enhance our predictions by choosing words both off their probability score as generated by the decoder and its semantic fit into the sentence.

### Data

We used the Armeni et al. (2022) dataset. It consists of 3 right-handed, native English speakers (1 female) listening to ten stories of "The Adventures of Sherlock Holmes" by Arthur Conan Doyle while in a MEG-compatible head cast to minimize motion artifacts from head movement.

To prepare the word labels for our dataset, we tokenized the text from the 10 stories using spaCy's tokenizer. These tokens were used to create the vocabulary of all possible word labels. Contractions and other words that were not consistent between the vocabulary and the events were skipped to ensure a closed-set classification space.

Word onset timings were extracted from the provided `events.tsv` files, filtering out any entries labeled as silence ("sp") or non-word events. Each context window was extracted from $-0.5$ to $2.0$ seconds relative to each word onset, resulting in a fixed shape tensor of $275 \times 500$ for each word that was heard.

### Decoders

The goal of the decoder model is to predict the word that was heard given a context window containing the MEG data corresponding to that word. The class probabilities outputted from the decoder model are used by the LLM.

We experimented with common deep learning architectures used for MEG decoding tasks such as RNNs and LSTMs, and used an MLP as a baseline. RNNs and LSTMs were our primary model choice for the decoder model because of the high temporal resolution of the MEG data. RNN and LSTM based architectures are highly effective in handling sequential data and have been widely used for similar neural decoding tasks.

### LLM Fine-Tuning

The goal of integrating the LLM is to pass through a context window of sequential probability distributions generated by the decoder model. Then, using the options presented to the LLM, it will choose words that fit best within the sequence. For the purposes of this project, we chose a sequence length of 15 and only presented the top 5 candidates from the decoder to the LLM in order to balance token length and reasoning complexity necessary to complete the task. We chose to work with the Llama 3.2 3B INSTRUCT model for this project Meta (2024). For the LLM fine-tuning portion of our project we generated synthetic data using 5 other Sherlock Holmes stories that weren't contained in the original dataset in order to prevent overfitting.

**Prompt Tuning:**

We first tweaked a prompt to get close to the results we wanted (Appendix A.1) on the base model.

**Low-Rank Approximation Adaptor:**

In order to tweak the LLM efficiently with the time and resources available to us we used a LoRA adaptor Gang et al. (2025). Instead of preforming gradient descent upon the base model we freeze the original weights $W_0$ and preform gradient descent with respect to $W^* = W_0 + \alpha AB$, where $A$ and $B$ are cold-started matrices with rank 8. This reduces the number of tunable of parameters from 3.21B to 2.7M, greatly reducing train time.

**Instruction Fine-Tuning:**

Our base model struggled to properly format its final answer in a consistent manner that could be parsed out. We generated a set of 5000 simulated responses that demonstrated the chain of reasoning expected by the model and ended in a properly tagged format. We preformed supervised learning on our LoRA adaptor until the model outputed an answer in the correct format over $97\%$ of the time. We tried to minimize the amount of SFT to not overfit our model to the simulated responses and preserve diverse reasoning skills Dong et al. (2023).

**Online DPO:**

Finally, we performed Online Direct Preference Optimization on the LoRA adaptor. We chose Online DPO over other methods like RLHF or Offline DPO because it has been shown to more efficiently align LLMs with less resources Guo et al. (2024) and it was easy to assign preference labels based off of accuracy scores of the final predictions.

## 4 Experimental Setup

### 4.1 Decoder

To decode heard speech from the MEG data, we trained deep learning models to predict the most likely word given the context window for the word. We framed decoding as a classification problem, where the model attempts to classify the correct word for the context window out of the total 6975 words in the vocabulary. Each context window consisted of MEG data $-0.5$ to 2 seconds relative to word onset and resampled to 200 Hz, resulting in input tensors of shape $275 \times 500$.

We implemented four decoder models with increasing complexity:

1. A baseline MLP with two hidden layers of sizes (128, 64).
2. A shallow RNN with four hidden layers of 512 units.
3. A deep RNN with six hidden layers of 768 units and dropout of $p = 0.3$.
4. A bidirectional LSTM with a convolutional layer (filter size $32 \times 64$), followed by two hidden LSTM layers with 256 units in each direction.

We used Cross Entropy Loss as the objective to optimize, given that this was framed as a classification problem. All models in the initial stage were trained for sessions 1-9 for subject 1, and the 10th session was used as a validation set to assist with final model selection and hyperparameter tuning. After this stage, the best performing decoder model, which was the deep RNN as shown in results, was trained fully on 2 subjects (for all 10 sessions) and the first 5 sessions of subject 3. This trained decoder was integrated with the tuned LLM.

### 4.2 LLM

We generated our synthetic data to be from a decoder with 50% accuracy. Though this is significantly higher than our actual decoder we wanted to simulate an environment with sufficient signal that an LLM could feasibly improve the result. We first constructed a unigram distribution from the corpus of the 5 held out stories. Then, for each position in the sequence, we sample 5 words including the true word and assign them probabilities. To do this we sample 5 normal random variables where the mean is 0.3 for the true word and 0.1 for the others and standard deviation is 0.1, add the resulting values to the unigram distribution probability, and clip the values to be valid probabilities. By a quick statistical calculation, this gives the true word a 50% chance of being the highest chosen word by the "model." We generated 5000 synthetic sequences and split them into a 90/10 train-validation split.

For the SFT we first looked at the reasoning results of the base model and replicated them with some slight tweaking. The "ideal responses" go through each position and select a word with a 50% chance of including a randomly selected reason. At between 3 and 10 of the positions the "model" will choose the wrong word. After all the words are chosen the model will go back through and correct the incorrect words, giving reasoning. Finally it will format the final answer with tags for easy parsing, see Appendix A.2 for more details. We performed SFT for 50 epochs with a batch size of 8 and learning rate of $3e-4$. At each 10 epochs we evaluated the performance of our LLM for both correct tagging and overall accuracy on 30 synthetic sequences. If the LLM correctly tagged its answer at a rate greater than 96% we stopped SFT early to prevent early fitting.

For the Online-DPO we generated sets of on-policy response pairs and then labeled them with the one with the highest accuracy being "preferred" (if they had the same accuracy, the shorter response was preferred). We trained with a batch size of 8. After 10 epochs we evaluated the model with the same metrics as SFT and then generated a new set of reposnse pairs using the new policy. We ran Online-DPO for 50 epochs.

### 4.3 Integration

Finally we integrated the two methods together by generating 100 context windows using the true probabilities chosen by the best performing decoder model and evaluated the accuracy.

## 5 Results

Our evaluation consisted of two complementary analyses: a quantitative assessment to measure the raw decoding accuracy and tagging consistency of both the neural decoders and the fine-tuned language models, and a qualitative examination of representative reconstructions to uncover systematic patterns in successes and failures. In the quantitative analysis, we first compare four deep-learning architectures on the closed-vocabulary classification task, highlighting the deep RNN's superior performance of 5.92% accuracy- nearly six orders of magnitude above random chance. We then present how instruction fine-tuning (SFT) and Online DPO reshape the Llama 3.2 3B model's outputs, boosting tagging consistency from 0% to as high as 90% and raising task accuracy to over 60% on simulated decoder distributions. Following the numerical results, our qualitative analysis provides an overview of how the different tuning methods influence the output of the LLM- for better and for worse.

### 5.1 Quantitative Evaluation

#### 5.1.1 Decoders

From the results of training the decoder model, we found that the best performing decoder was the deep RNN. This is expected due the sequential nature of the MEG data. One of the challenges was the imbalanced nature of the dataset, since there were lots of occurrences of common words such as 'and', 'I', 'the', etc. compared to other words that occurred only once across all 10 stories. To address

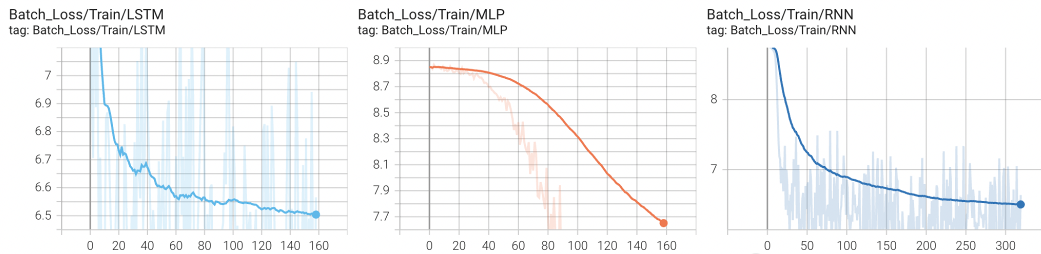


Figure 2: Loss decreasing while training decoder models.

this, we tried various regularization methods. Dropout helped slightly but we did not see a drastic improvement in performance. Other methods that could have helped with this challenge are custom or weighted loss functions, or reformulating the problem from next-word prediction to an easier task. After the initial training results, we trained the deep RNN model on a larger set of the data

Table 1: Performance Comparison for Decoder Models

| Model | Accuracy |
|---|---|
| Random chance | 0.01% |
| MLP (Baseline) | 1.96% |
| RNN (shallow) | 5.54% |
| RNN (deep) | 5.59% |
| Bidirectional LSTM with Convolutional Layer | 5.38% |

which included all 10 sessions for subjects 1  2, and the first 5 sessions for subject 3. The additional data helped improve the performance of the decoder model. Using the remaining held out data, we achieved an accuracy of 5.92%.

#### 5.1.2 LLM

| Tuning Technique | Epochs Trained | Tagging Rate | Accuracy |
|---|---|---|---|
| Base | – | 0 | 46* |
| SFT | 30 | 63 | 56 |
| SFT | 50 | 90 | 55 |
| Online DPO† | 50 | 73 | 68 |
| RNN (Deep) + LLM | – | 74 | 4.31 |

Table 2: Tagging and accuracy results by tuning technique. * because the base model was unable to tag it's answer in a consistent manner we manually calculated the answer. † Online DPO was preformed on the LoRA Adaptor after being SFT'd for 30 epochs due to time.

The results, as seen in Table 2, show that SFT is successful at teaching the model how to properly tag its solution, with a small boost to the accuracy above the base model quickly. However, while Online DPO proved to be much more effective at improving the accuracy of the model's predictions, it seemed to struggle to encourage the model to tag it's responses correctly. This support our use of both SFT and Online DPO in tuning the LLM.

On the otherhand, combining the real results of the RNN and LLM slightly decreased the model's accuracy. This is to be expected for a few reasons. First of all, with an accuracy of below 10% the LLM doesn't have much signal to form cohesive sentences. Secondly, the LLM was tuned from a simulated decoder that has a 50% base accuracy so in all the training data it has seen the highest probability word is correct 50% of the time. This encourages the model to chose the highest probability instead of trying other words. By tuning the simulated decoder to be more closely aligned with the RNN's accuracy and performing Online DPO on the tuned-LLM using some of the held out RNN data would likely help realign the LLM and improve the accuracy slightly.

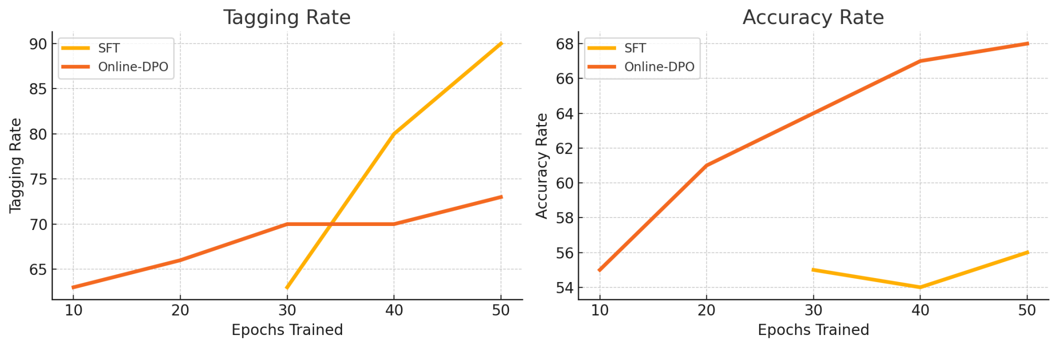


Figure 3: LLM tuning results. First 2 epochs of SFT weren't able to be validated due to max_tokens being set too low.

### 5.2 Qualitative Analysis

Taking a qualitative look at the LLM's outputs we see a few things. First of all the base model has a hard time stopping. In addition to failing to tag its answer, it will continue trying to predict new words and is prone to generating completely new distributions - unrelated to the inputed data - so that it can make more predictions.

As for SFT, though the model slowly became more aligned over time, it also began to become less creative as a result of overfitting. Looking through the validation results as the number of epochs increased the variety of language decreased and the model was less likely to use reasons to change words that weren't present in the simulated responses. This is why we were careful to only tune using SFT as much as required.

Finally, we saw a few different behaviors as a result of DPO. Though it also increased the likelihood that responses were tagged correctly (because an accuracy of 0 always got dis-preferred) it didn't have the same effect of decreasing reasoning variety as SFT. Additionally as training continued under DPO, the model became more willing to exchage words during the revision step, increase accuracy. However, Online DPO also saw an increase in erratic behaviors in comparison to SFT. Sometimes the model would continue reasoning after providing the final answer, going off on tangents until it reached the token limit. This suggests that we should use a more complex labeller than just simple accuracy to reign in DPO responses.

## 6 Discussion

Our results demonstrate that while deep learning architectures can partially decode linguistic information from MEG recordings during natural speech listening, the task of decoding heard speech in naturalistic settings remains challenging with current approaches. The deep RNN achieved the highest accuracy of 5.92% after being trained on a larger portion of the dataset, representing a substantial improvement over random chance (0.01%) but still falling short of practical decoding thresholds.

Our vocabulary size of 6,975 unique words represents a substantially more challenging classification problem than typical top-k closed-set problem formulations with 10-100 words. The class imbalance with words like "the" and "and" occurring many times more frequently than other less common words makes the model prone to overfitting the more common words. While we attempted to address this through dropout regularization, more sophisticated approaches such as focal loss or hierarchical classification with part-of-speech classification first might yield better results. Additionally, exploring shared embedding or alignment methods like CLIP Radford et al. (2021) which significantly improved performance for MEG-to-text speech decoding in Yang et al. (2024), could also have resulted in improved performance. For the purpose of this class, we chose to focus on the RL aspect of the problem formulation.

Since our decoder didn't perform as well as we expected, we pivoted to using the simulated decoder when evaluating instruction tuning methods to allow for enough positive predictions to be input into the LLM.

With the base LLM, the task accuracy was 0% and manual evaluation by parsing the prediction from the LLM output was required. Instruction tuning and DPO both improved the task accuracy to 63% and 62% respectively. This suggests that it's challenging for small language models to follow instructions and output answers in a desired format without fine-tuning. Addressing the remaining failure modes (e.g., grammatical overcorrections, reasoning drift) require richer preference labelling and more robust fine-tuning objectives.

Despite the limitations in decoding models, our hybrid approach combining neural decoders with LLMs points toward a promising direction for future research. As LLMs continue to improve and as we develop better techniques for fine-tuning them, this framework could help overcome the accuracy limitations of only deep learning based neural decoding approaches.

## 7 Conclusion

This work demonstrates both the promise and current limitations of integrating large language models with neural decoding for heard speech reconstruction from MEG recordings. While our deep RNN decoder achieved modest but meaningful performance (5.92% accuracy) in extracting linguistic information from naturalistic speech listening, this performance remains insufficient for direct LLM enhancement. However, our simulated experiments with 50% decoder accuracy reveal the potential of hybrid approaches, where fine-tuned LLMs can leverage their linguistic knowledge to improve word sequence coherence. The substantial improvements achieved through instruction tuning and DPO (from 0% to 63% and 62% task accuracy) highlight both the challenges of working with smaller language models and the necessity of comprehensive fine-tuning pipelines for specialized tasks. Although practical brain-computer interfaces for natural speech decoding remain distant, our work establishes a framework for combining neural decoding with RL-tuned LLMs.

## 8 Team Contributions

- **Ricky Rojas** Cleaned and preprocessed MEG, generated prompts and synthetic data for LLM tuning, implemented SFT and Online-DPO and evaluated them.
- **Ramani Tyagi** Implemented the deep learning decoder for word prediction from MEG data context windows.

**Changes from Proposal** We changed our project from setting up an RL agent to fine-tuning an LLM. We decided to use simulated decoder outputs for evaluating the performance of instruction tuning and DPO since there weren't enough positive signals from our trained deep learning decoder with poor accuracy.

## References


K. Armeni, U. Güçlü, M. van Gerven, et al. 2022. A 10-hour within-participant magnetoencephalography narrative dataset to test models of language comprehension. *Scientific Data* 9 (2022), 278. `https://doi.org/10.1038/s41597-022-01382-7`

C. Caucheteux, A. Gramfort, and J. R. King. 2023. Evidence of a predictive coding hierarchy in the human brain listening to speech. *Nature Human Behaviour* 7 (2023), 430–441. `https://doi.org/10.1038/s41562-022-01516-2`

Debadatta Dash, Paul Ferrari, and Jun Wang. 2020. Decoding Imagined and Spoken Phrases From Non-invasive Neural (MEG) Signals. *Frontiers in Neuroscience* 14 (2020), 290. `https://doi.org/10.3389/fnins.2020.00290` Original Research; Section: Neural Technology.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. How Abilities in Large Language Models are Affected by Supervised Fine-tuning Data Composition. `https://doi.org/10.48550/arXiv.2310.05492` arXiv:arXiv:2310.05492 [cs.CL] Accepted to ACL 2024 Main Conference; last revised June 7, 2024.

Yue Gang, Jianhong Shun, and Mu Qing. 2025. Smarter fine-tuning: How LoRA enhances large language models. *arXiv preprint arXiv:2407.19299* (2025).

A. Goldstein, Z. Zada, E. Buchnik, et al. 2022. Shared computational principles for language processing in humans and deep language models. *Nature Neuroscience* 25 (2022), 369–380. https://doi.org/10.1038/s41593-022-01026-4

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, Johan Ferret, and Mathieu Blondel. 2024. Direct Language Model Alignment from Online AI Feedback. *arXiv preprint arXiv:2402.04792* (2024). https://doi.org/10.48550/arXiv.2402.04792 arXiv:arXiv:2402.04792 [cs.AI] Version 2, last revised February 29, 2024; accessed June 8, 2025.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv preprint arXiv:2005.01643* (2020). https://doi.org/10.48550/arXiv.2005.01643 Last revised 1 Nov 2020, version 3.

Meta. 2024. Llama 3.2 3B Instruct. Hugging Face Model Card, https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct. Release date: September 25, 2024; accessed June 8, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. *CoRR* abs/2103.00020 (2021). arXiv:2103.00020 https://arxiv.org/abs/2103.00020

Yiqian Yang, Hyejeong Jo, Yiqun Duan, Qiang Zhang, Jinni Zhou, Won Hee Lee, Renjing Xu, and Hui Xiong. 2024. MAD: Multi-Alignment MEG-to-Text Decoding. arXiv:2406.01512 [cs.CL] https://arxiv.org/abs/2406.01512

# A Implementation Details

## A.1 LLM Input

The prompt given to the LLM was structured in two parts:

**1. Instructions**

```
<Instructions>

You are given a sequence of 15 words, where each word is represented by the
**top 5 most likely candidates** predicted by a language model, along with
their softmax distributions. These predictions are not perfect - the model
has **some knowledge of the true words**, but may be wrong.

Your task is to:
1. **Infer the most likely true sequence** of the last 15 words based on the
   provided word distributions. Use probabilities to guide your choices, but
   prioritize sentence coherence, grammar, and semantic plausibility.
2. **Choose the best next word** to follow the inferred sequence. This can be
   one of the 5 candidates, or a different word that fits better.
3. **Use natural language to reason through your selections.**

Guidelines:
- For each of the 15 positions, run a ‘<choosing>‘ step: weigh probability and
  semantic/grammatical fit, selecting the best word (you may pick a
  lower-probability word if it improves coherence).
- Keep a running list of chosen words to guide later decisions.
- Once all 15 words are chosen, perform a ‘<revision>‘ pass to swap out any
  poorly fitting words via another ‘<choosing>‘ step.
- End your prompt exactly with the required format to receive credit:
  ‘<answer>word_1 word_2 ... word_15</answer>‘

</Instructions>
```

### 2. Data Input

```
{Data Input}
Sequence Distributions:

Position 1:
- what (0.37)
- see (0.27)
- success (0.18)
- i (0.15)
- was (0.03)

Position 2:
- salary (0.31)
- you (0.20)
- a (0.17)
- out (0.14)
- carbuncle (0.12)

% ... etc. for all 15 positions ...
```

### A.2 SFT Sample Response

Below is an example response used for supervised fine-tuning:

```
I will construct the sequence by going through each of the 15 positions.

<choosing>
Position 1: He She It
The best choice is ‘It’
Position 2: was had is
The best choice is ‘is’
Position 3: not no a
The best choice is ‘a’
Position 4: major gross capital
The best choice is ‘capital’
Position 5: error flaw mistake
The best choice is ‘flaw’ because it is the most semantically coherent
Position 6: for toward to
The best choice is ‘to’
Position 7: speculate guess theorize
The best choice is ‘theorize’
Position 8: after during before
The best choice is ‘before’
Position 9: any someone one
The best choice is ‘someone’ because it fits the sentence structure
Position 10: owns holds has
The best choice is ‘has’
Position 11: facts numbers data
The best choice is ‘data’
Position 12: for in to
The best choice is ‘in’ because it matches the grammatical tense
Position 13: verify validate confirm
The best choice is ‘confirm’
Position 14: a this the
The best choice is ‘the’
Position 15: hypothesis. assumption. theory.
The best choice is ‘theory.’
</choosing>
```

The predicted sequence is:
"It is a capital flaw to theorize before someone has data in confirm the theory."

<revision>
Position 1: ‘It’ fits the sequence.
Position 2: ‘is’ fits the sequence.
Position 3: ‘a’ fits the sequence.
Position 4: ‘capital’ fits the sequence.
Position 5: replaced ‘flaw’ with ‘mistake’ because it is high probability.
Position 6: ‘to’ fits the sequence.
Position 7: ‘theorize’ fits the sequence.
Position 8: ‘before’ fits the sequence.
Position 9: replaced ‘someone’ with ‘one’ because it fits contextually.
Position 10: ‘has’ fits the sequence.
Position 11: ‘data’ fits the sequence.
Position 12: replaced ‘in’ with ‘to’ because it fits the sentence structure.
Position 13: ‘confirm’ fits the sequence.
Position 14: ‘the’ fits the sequence.
Position 15: ‘theory.’ fits the sequence.
</revision>

<answer>It is a capital mistake to theorize before one has data to confirm the theory.</answer>
</Reasoning>