LITcoder: A General-Purpose Library for Building and Comparing Encoding Models

Taha Binhuraib
Georgia Tech
tbinhuraib3@gatech.edu

Ruimin Gao
Georgia Tech
ruimin.gao@gatech.edu

Anna A. Ivanova
Georgia Tech
a.ivanova@gatech.edu

Abstract

We introduce LITcoder, an open-source library for building and benchmarking neural encoding models. Designed as a flexible backend, LITcoder provides standardized tools for aligning continuous stimuli (e.g., text and speech) with brain data, transforming stimuli into representational features, mapping those features onto brain data, and evaluating the predictive performance of the resulting model on held-out data. The library implements a modular pipeline covering a wide array of methodological design choices, so researchers can easily compose, compare, and extend encoding models without reinventing core infrastructure. Such choices include brain datasets, brain regions, stimulus feature (both neural-net-based and control, such as word rate), downsampling approaches, and many others. In addition, the library provides built-in logging, plotting, and seamless integration with experiment tracking platforms such as Weights & Biases (W&B). We demonstrate the scalability and versatility of our framework by fitting a range of encoding models to three story listening datasets: LeBel et al. (2023), Narratives, and Little Prince. We also explore the methodological choices critical for building encoding models for continuous fMRI data, illustrating the importance of accounting for all tokens in a TR scan (as opposed to just taking the last one, even when contextualized), incorporating hemodynamic lag effects, using train-test splits that minimize information leakage, and accounting for head motion effects on encoding model predictivity. Overall, LITcoder lowers technical barriers to encoding model implementation, facilitates systematic comparisons across models and datasets, fosters methodological rigor, and accelerates the development of high-quality high-performance predictive models of brain activity.

Project page: https://litcoder-brain.github.io

1 Introduction

Encoding models are a powerful tool for predicting brain responses to external stimuli. Traditionally, encoding models were used to model responses to static images [1–5]; however, in the last decade their use has been expanded beyond vision to other modalities, such as language [6], and from static to continuous stimuli, such as stories [7–9]. These advances make it possible to study brain responses to diverse stimuli in naturalistic settings, track hierarchical processing across cortical networks, and evaluate the extent to which modern AI models align with brain activity [10, 11]. However, the rapid growth of encoding-model-based research has resulted in substantial heterogeneity of encoding model design choices made by different research groups, complicating replication and cross-paper comparisons. We provide a methodological framework aimed at resolving these inconsistencies.

A central challenge for the field is the lack of standardized infrastructure for building and evaluating encoding models. Most works that build encoding models of continuous stimuli only report results
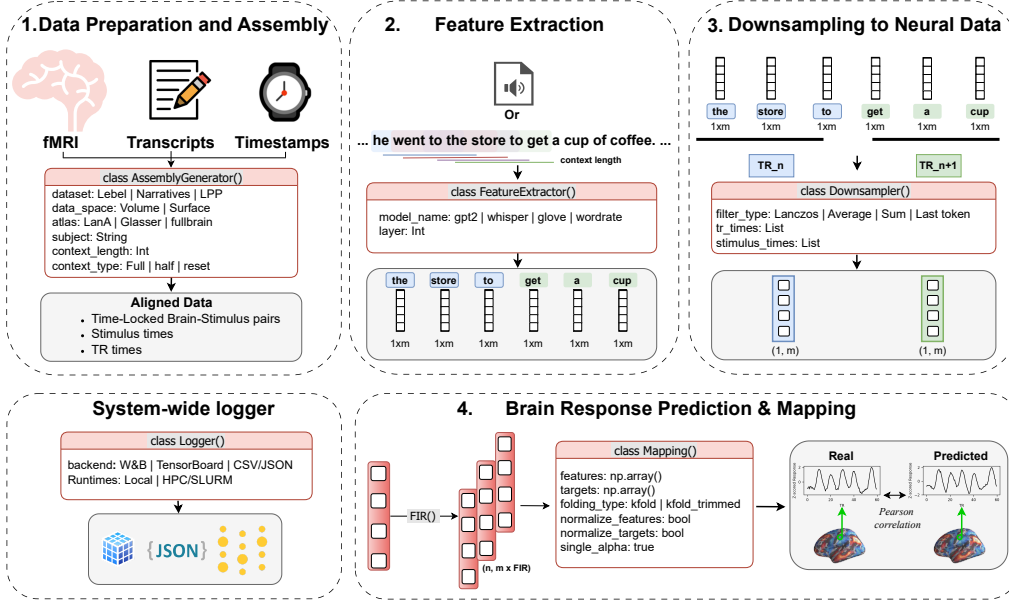
Figure 1: **Overview of the LITcoder library architecture.** The library implements a modular pipeline for constructing and evaluating neural encoding models. **(1)** Functional MRI data, aligned transcripts, and timestamps are processed through the `AssemblyGenerator` to produce time-locked brain–stimuli pairs in either volumetric or surface brain spaces. TR=repetition time, a term denoting fMRI data acquisition timepoints. **(2)** Stimuli (text or audio) are transformed into representational features using the `FeatureExtractor`, which supports a variety of models (e.g., GPT-2, Whisper, GloVe) and non-model-based features (such as word rate). **(3)** Extracted features are passed to the `Downsampler`, which implements multiple pooling and filtering strategies (Lanczos, average, sum, last token) to produce TR-matched features. **(4)** Aligned features are expanded using finite impulse response (FIR) modeling and passed to the `Mapping` module, which supports flexible cross-validation schemes (e.g., k-fold), feature normalization, and voxel-/vertex-wise prediction using ridge regression. All stages are monitored using the `Logger`, which interfaces with multiple backends (e.g., Weights & Biases, TensorBoard, CSV/JSON) and runtime environments (local, HPC, SLURM).

from a single dataset [12]; one reason for this gap is the heterogeneous nature of these datasets' formats, making it difficult to apply the same design choices to all. Furthermore, these works rely on ad hoc pipelines that vary in feature extraction, temporal alignment, cross-validation, and evaluation metrics. Such heterogeneity makes results difficult to compare across papers and limits reproducibility [13]. In some cases, methodological choices can introduce systematic biases that inflate predictivity if not carefully controlled. As an example, Hadidi et al. [14] demonstrate how temporal autocorrelation can produce spurious model performance, highlighting the need for principled approaches that minimize such confounds. Together, these issues underscore the importance of consistent, transparent pipelines that allow reliable comparison across datasets, models, and experimental paradigms.

To address these limitations, we introduce **LITcoder**, an open-source library for the systematic evaluation of neural encoding models (Figure 1). LITcoder provides a modular backend that standardizes core components of the pipeline: feature extraction, temporal alignment and downsampling, hemodynamic lag modeling, voxelwise/vertexwise mapping, and evaluation. The current implementation supports fMRI as the neural data source and language (text and audio) as the input stimulus, but it can be extended to other neural data modalities (e.g., MEG and iEEG) and stimulus types (e.g., videos and audiovisual inputs). The library enforces reproducible, transparent practices while remaining extensible—allowing researchers to plug in new datasets, models, or alignment strategies with minimal overhead. By unifying infrastructure while decoupling it from front-end interfaces, LITcoder makes it vastly easier to compare encoding model performance across datasets, feature spaces, and methodological choices, fostering reproducibility and methodological rigor.

2 Related Work

One well-known example of standardized encoding model pipelines is **Brain-Score**, a powerful framework for benchmarking models against the brain. Brain-Score aggregates multiple datasets and metrics to provide a common interface for ranking candidate neural networks by their ability to predict neural and behavioral responses for vision [15] and language [6]. A key difference between Brain-Score and LITcoder is that the former aims for standardization, whereas we aim for flexibility. When a new neural benchmark and/or model is integrated into Brain-Score, all encoding model design choices are fixed (from stimulus context length to the neural network layer that should be used to map onto neural data). In contrast, LITcoder provides the flexibility to easily experiment with dozens of encoding model choices. We therefore view the Brain-Score and the LITcoder frameworks as complementary.

To select encoding model design choices that LITcoder should offer, we take inspiration from existing encoding model frameworks, primarily in the domain of language. We choose four representative papers from different research groups to sample the diversity of design strategies: Schrimpf et al. [6], Antonello et al. [8], Oota et al. [16], and Caucheteux and King [9]. We have identified core design choice decisions made by each of these papers, compiled them together (Table 1), and implemented these options in LITcoder for systematic comparison.

Table 1: Encoding model design choices across four reference studies.

Study	Downsampling	FIR Delays	Context Updating	Dataset(s)	Projection
Schrimpf et al. [6]	Last-token	No FIR modeling	Full context	Blank2014[17], Pereira2018[18]	Volume
Antonello et al. [8]	Lanczos	4 delays	Half context	LeBel	Volume
Caucheteux et al. [19]	Sum/Average	6 delays	Full context	Narratives	Surface
Oota et al. [16]	Lanczos	8 delays	Full context	Narratives	Surface

3 The Framework

LITcoder is designed as a modular backend that provides flexible, interchangeable components for building encoding models while abstracting away implementation details. Each module inherits from a small set of base classes, ensuring that new datasets, stimulus feature extractors, or downsampling procedures can be added with minimal boilerplate. This modularity enables researchers to prototype new ideas rapidly while maintaining a consistent scientific pipeline.

LITcoder proceeds in four steps (Fig. 1): **(1) AssemblyGenerator** aligns BIDS fMRI with transcripts in volume or surface space; **(2) FeatureExtractor** turns the aligned stimuli into features (from baselines to GPT-2/Whisper); **(3) Downsampler** aggregates those features to TR resolution; **(4) FIRExpander** adds lags and passes the design matrix to **Mapping**, which fits ridge models with flexible CV and returns voxel/vertex predictivity. All runs are automatically logged via a `Logger` that interfaces with Weights & Biases, TensorBoard, or lightweight CSV/JSON backends across local, HPC, and SLURM environments.

3.1 Data Preparation and Assembly

This first module takes in functional MRI data (in BIDS format), stimulus transcripts, and timestamps. These inputs are integrated by the `AssemblyGenerator`, which outputs time-locked brain–stimuli pairs, as well as additional files with stimulus onset times and TR times. The module provides flexibility in supporting both volume and surface brain representations, whole-brain and region-based anatomical masks (e.g., from the Glasser atlas [20]), and a range of stimulus context lengths when generating stimulus–brain pairs.

Regardless of dataset origin, after assembly all downstream modules receive identically structured inputs. This ensures that observed differences in model performance can be attributed to encoding model design choices rather than idiosyncrasies of data handling.

Contextual Representation Strategies. LITcoder implements three policies for managing context windows. **Full context** uses a fixed-size sliding window and extracts the hidden state of the final token of each word, enabling long-range dependency modeling [6]. **Half context** advances the window

start index by half the window size, reducing computational load while retaining partial overlap [8]. **Reset context** resets the window entirely when the lookback length exceeds the maximum, producing local non-overlapping representations. These strategies provide opportunities to balance comprehensiveness and computational efficiency.

3.2 Feature Extraction

A key design goal of LITcoder is to make the extraction of stimulus features both flexible and extensible. Through the `FeatureExtractor` base class, researchers can draw from a wide range of representational sources: pretrained language models (e.g., GPT-2, Pythia), speech models (e.g., Whisper, HuBERT), static word embeddings (e.g., GloVe, word2vec), and low-level baselines such as word rate or autocorrelation. Implementing a new extractor requires inheriting from the base class and overriding a small number of functions, ensuring that custom models or representations can be seamlessly integrated into the same pipeline. This design separates library infrastructure from scientific experimentation, enabling reproducible extensions without additional effort.

3.3 Downsampling to Neural Data

A challenge in naturalistic modeling is the temporal mismatch between stimuli and neural data. Language models produce token-level activations ($\sim$3–4 tokens/s), while fMRI samples hemodynamic responses much slower (1.5–2 s TRs). The `Downsampler` bins token-level activations into TR-sized windows using sum pooling [9], average pooling [21], last-token selection, or Lanczos filtering [22].

3.4 Brain Response Prediction and Evaluation

The `Mapping` module implements voxel- or vertex-wise ridge regression with flexible evaluation protocols. Different folding schemes include standard k-fold cross-validation and k-fold with boundary trimming to reduce autocorrelation effects (see Section 4.3). Both feature and target normalization are available as options, and models can be run with a single global regularization parameter or voxel-specific parameters. LITcoder ensures comparability across datasets and methods. For visualization of voxel-wise and surface-based predictivity maps, we use `nilearn` [23].

Hemodynamic Response Modeling with Finite Impulse Responses (FIR). fMRI BOLD signal follows neural activation with a delay and temporal spread. To model this, feature vectors aligned to TRs are expanded into a finite impulse response (FIR) design matrix. For each TR-aligned feature vector $\mathbf{h}_t$, the design matrix includes the current vector and k lagged copies:

$$\mathbf{H}_t = [\mathbf{h}_t, \mathbf{h}_{t-1}, \ldots, \mathbf{h}_{t-k}],$$

where t indexes fMRI timepoints in units of TRs and k specifies the number of delays. This procedure allows the regression model to estimate voxel-specific response functions directly from the data rather than assuming a fixed hemodynamic response function. The FIR expansion therefore captures both the delayed onset and the extended temporal profile of the BOLD response, enabling accurate mapping between stimulus features and measured neural activity. In **LITcoder**, this functionality is encapsulated in a dedicated `FIRExpander` class, which constructs concatenated, delayed copies of stimulus features with configurable delays.

4 Experiments

We use LITcoder to run a set of experiments that test the library across feature families, alignment choices, and temporal modeling. Unless noted, **Predictivity Scores** are reported as *average voxel-wise Pearson correlations within the language network* as defined by the LanA atlas [25], computed on held-out data following the dataset-specific protocols in Appendix C. These experiments showcase the scientific utility of the pipeline and its extensibility across models and evaluation settings.

4.1 Datasets

We evaluated encoding models using three naturalistic fMRI datasets: Narratives [26], Little Prince [27], and LeBel [28]. All three involve participants listening to extended spoken narratives, providing ecologically valid contexts for studying language comprehension.
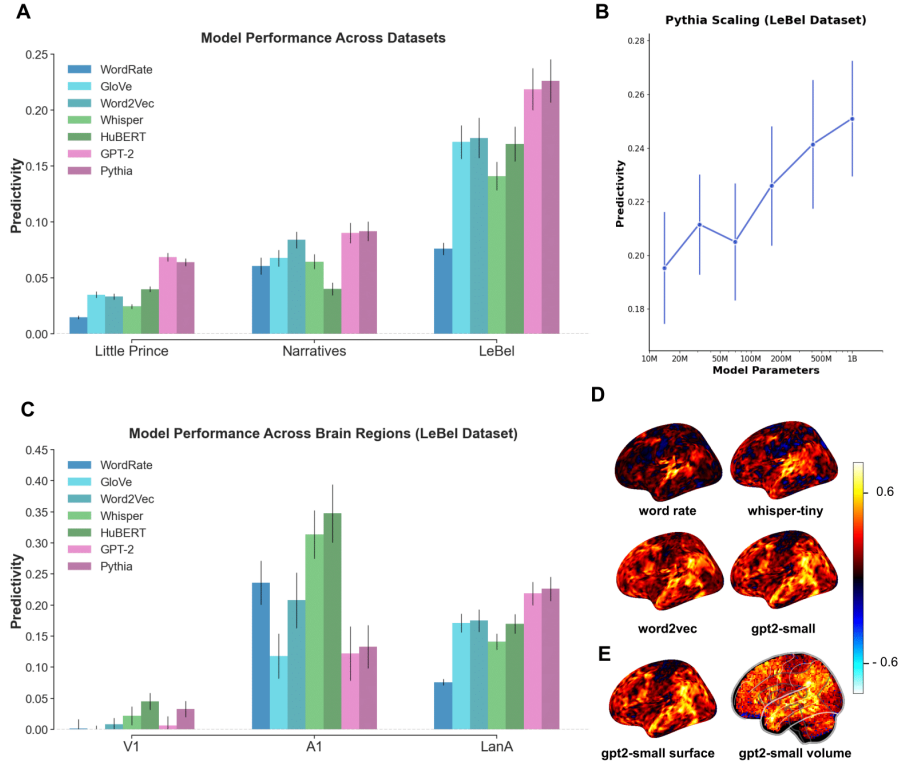
Figure 2: **Encoding model performance across feature families, datasets, and brain regions.** All scores in (A) and (B) are reported as *average voxel-wise correlations within the language network atlas*. **(A)** Predictivity across three datasets (Little Prince, Narratives, LeBel) for four feature families: simple baseline (word rate), static embeddings (GloVe, word2vec), speech models (Whisper-tiny, HuBERT-base), and language models (GPT-2 small, Pythia-160M). **(B)** Scaling of the Pythia language model family on the LeBel dataset, showing increased predictivity with model size. **(C)** Model performance across three cortical regions of interest: V1 (primary visual cortex), A1 (primary auditory cortex), and the language network. For language models, we selected layers that maximized predictivity within each ROI (layer 0 in A1 for both GPT-2 and Pythia; layer 6 in GPT-2 and layer 5 in Pythia for the language network; the same respectively for V1). For speech models, we used the final encoder layer, following Samara et al. [24]. **(D)** Whole-brain voxel-wise predictivity across the entire cortical surface for representative models (word rate, Whisper-tiny, word2vec, GPT2-small) for a single subject from the LeBel dataset. **(E)** GPT2-small-based encoding model's voxel-wise predictivity correlations in surface (fsaverage5) and volume (MNI) spaces; same subject as (D). Here and elsewhere, error bars denote standard error across participants.

The **Narratives** [26] dataset is one of the largest publicly available collections of auditory story-listening fMRI data, aggregating scans acquired over multiple years and labs. Stimuli comprised 28 naturalistic spoken stories ranging in duration from 3 to 56 minutes, totaling about 5 hours of unique audio material. Following [16], we focused on the story "21st year" (56 min), which has been widely adopted in prior work and allows comparability across studies. 18 participants listened to that story and were therefore included in our analysis.

The **Little Prince** [27] dataset includes fMRI data from participants listening to Antoine de Saint-Exupéry's book *Little Prince* in English, Chinese, and French, with audiobooks translated and read by professional narrators in each language. We restricted the analyses to the data from 49 English speakers, to maintain comparability with the other datasets.

The **LeBel** [28] dataset contains naturalistic story listening fMRI data from 8 participants. The stimulus set consisted of 26 autobiographical spoken stories (10-15 minutes each) from The Moth
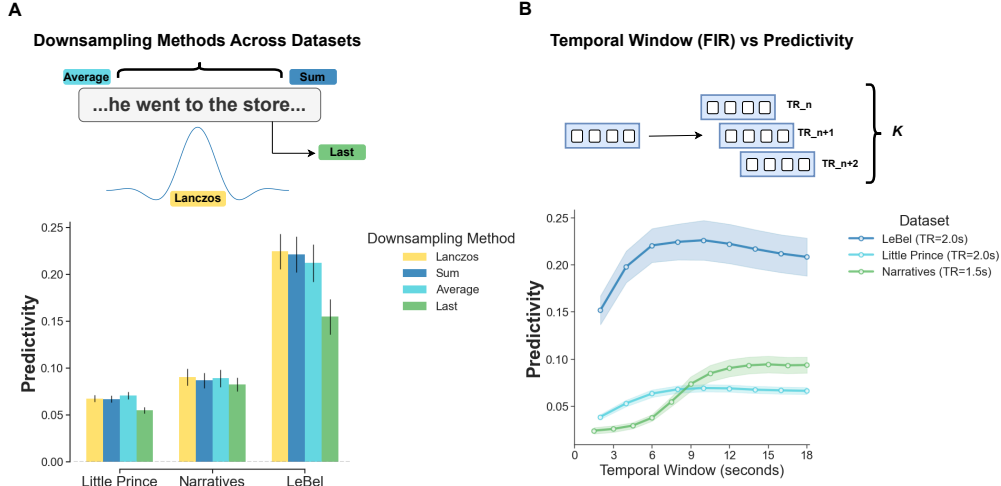
Figure 3: **Evaluating downsampling and temporal modeling choices. (A)** Comparison of token-to-TR downsampling methods. The schematic (top) illustrates four strategies implemented in the `Downsampler`: average pooling, sum pooling, final-token selection, and Lanczos filtering. The bar plot (bottom) shows predictive performance across datasets (LeBel, Narratives, Little Prince) using GPT-2 small, evaluated at the most predictive layer for each dataset. **(B)** Temporal modeling with finite impulse response (FIR). The schematic (top) illustrates FIR expansion, in which each TR-aligned feature vector is concatenated with k lagged versions. The line plot (bottom) shows the effect of varying the FIR window length k on predictivity for the best-performing GPT-2 small layers in each dataset.

Radio Hour, totaling about 320 minutes, plus one additional 10-minute test story presented in each session (total about 370 minutes per subject).

To ensure comparability, we preprocessed all datasets using `fMRIPrep` [29] with slice-time correction disabled and other parameters at default values (see Appendices D and C for details).

4.2 Experiment 1: LITCoder makes it easy to benchmark feature families across datasets

LITcoder enables comparisons across datasets, AI models, brain regions, and cortical surfaces. We evaluated four families of stimulus features: simple baselines (word rate), static embeddings (GloVe, word2vec), speech models (Whisper-tiny, HuBERT-base), and language models (GPT-2 small, Pythia) (see Appendix B for model details).

First, we show model performance across all feature sets and datasets on the language network voxels, defined with the LanA atlas (Figure 2A). As expected, features from language models (GPT-2 small and Pythia-160m) predict neural responses better than other feature sets. Then, we replicate the findings (e.g., [8]) that larger models typically yield better predictivity, using the Pythia models as our feature set and the language-network-constrained neural data from the LeBel dataset (Figure 2B).

Figure 2C extends the analysis to multiple cortical regions of interest—V1 (primary visual cortex), A1 (primary auditory cortex), and Language network—showing that predictivity is highest in regions associated with auditory and language processing. Figure 2D showcases model performance projected onto the entire cortical surface (fsaverage5, ~22k vertices) for subject UTS03 from the **LeBel** [28] dataset, demonstrating a way to quickly visualize voxelwise predictivities. Finally, Figure 2E shows surface- and volume-based analyses for GPT-2 small, indicating that LITcoder supports both vertexwise and voxelwise encoding models and that these models yield similar results if all other parameters are held constant.

Together, these analyses illustrate how LITcoder enables systematic comparison of model families and datasets while remaining flexible in spatial scope, from targeted region-of-interest evaluation to full-cortex mapping.

4.3 Experiment 2: Downsampling and temporal modeling choices affect predictivity

To demonstrate how LITcoder can be used to iteratively refine methodological design choices, we provide a side-by-side comparison of feature downsampling strategies and varying FIR lengths.

Downsampling. Figure 3A compares four aggregation strategies with GPT-2 small at the most predictive layer for each dataset. Average and sum pooling perform similarly, with Lanczos filtering matching or slightly exceeding them. Last-token selection produces consistently lower scores, suggesting that ignoring within-TR tokens loses useful information.

Temporal modeling. Figure 3B shows the effect of FIR window length. Adding temporal lags improves predictivity until performance stabilizes—around 9–12 s for LeBel and Little Prince (TR=2.0s), and somewhat shorter for Narratives (TR=1.5s). Extending the window further yields diminishing or negative returns, consistent with overfitting and noise.

Overall, downsampling methods that incorporate all tokens within a TR window generally yield higher predictivity than last-token selection, and finite impulse response (FIR) expansion improves performance up to a dataset-specific plateau.

4.4 Experiment 3: Mitigating spurious predictivity with contiguous train-test splits

Time-contiguous stimuli induce strong temporal autocorrelation in both features and BOLD responses, which can inflate cross-validated predictivity if training and test splits are mixed in time. Incorrect folding strategies unfortunately remain a major problem in the literature [14]. Following Hadidi et al. [14], we implement *Autocorrelation Control Vectors (ACVs)* whose adjacent samples are intentionally similar to quantify the extent to which models can exploit temporal structure alone (see Appendix A).

LITcoder standardizes this through a `folding_type` parameter, which supports three strategies: shuffled folds (randomized splits that break temporal contiguity), contiguous folds (non-overlapping time blocks that preserve temporal structure), and contiguous folds with boundary trimming [16] to reduce leakage from neighboring samples.

We apply this analysis to **Narratives** [26] and **Little Prince** [27], the two datasets in our suite that rely on cross-validation rather than a separate held-out story. Results (Figure 4) show substantially higher scores under *Shuffled* folds, consistent with temporal leakage; *contiguous* folds reduce this effect, and *trimmed* folds further mitigates boundary bleed. These results highlight the crucial importance of respecting the temporal structure in fMRI data during cross-validation. Shuffled folds artificially inflate model performance by letting temporally adjacent samples leak across train-test boundaries, creating a misleading sense of predictivity. Contiguous folds provide a more conservative and realistic benchmarks, and trimmed folds add further protection against boundary bleed.
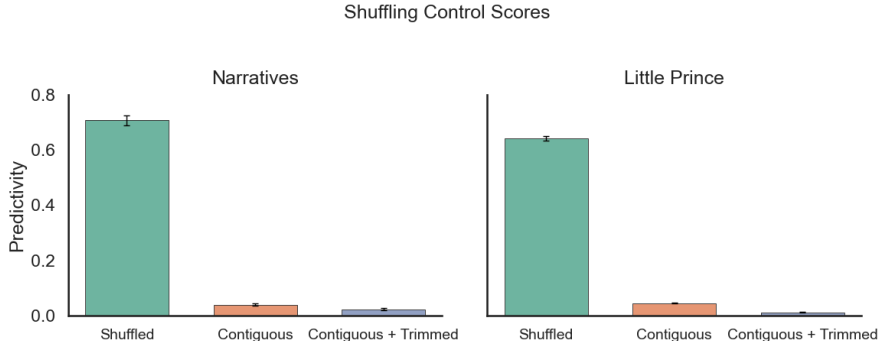

Figure 4: **Effect of train-test splitting choices.** Bars show average voxel-wise correlations across the full cortical surface for **Narratives** [26] (left) and **Little Prince** [27] (right) under three cross-validation schemes: **Shuffled** (randomized folds that ignore temporal order), **Contiguous** (non-overlapping time blocks), and **Contiguous + Trimmed** (contiguous folds with boundary trimming).
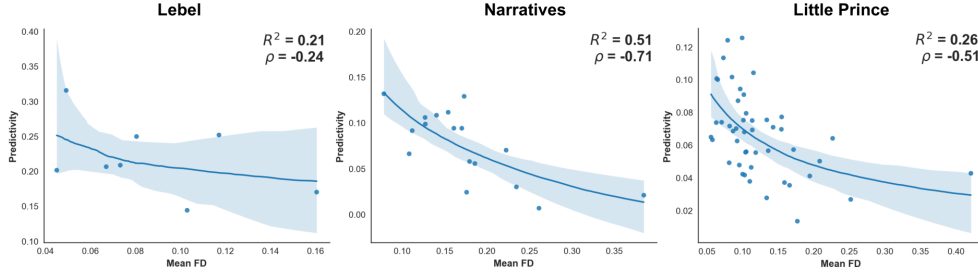
Figure 5: **Head motion vs. encoding model predictivity across datasets.** Each panel shows subject-level predictivity (language areas' mean voxel correlation from the best-performing layer of GPT-2 small) versus mean framewise displacement (FD) for **LeBel** [28] (left), **Narratives** [26] (middle), and **Little Prince** [27] (right). Points are individual subjects; the solid curve shows a nonlinear power-law fit, and the shaded region indicates 95% confidence intervals. R^2 values quantify variance explained by the fitted power-law, and ρ denote the Spearman rank correlations between FD and predictivity.

4.5 Experiment 4: Head motion negatively predicts encoding performance

Naturalistic fMRI is susceptible to motion-related artifacts that reduce signal reliability and bias cross-subject comparisons. To quantify the extent to which motion explains variance in our encoding scores, we examined the association between head motion and model predictivity in each dataset. For each participant, we computed *mean framewise displacement (FD)* from the realignment parameters provided by `fMRIPrep` [29].

We focused on predictivity scores from the best-performing layer of GPT-2 small, evaluated within the language-network. For each dataset, voxel-wise performance values were averaged to obtain a subject-level predictivity score. We then examined the relationship between mean FD and these subject-level scores by plotting scatterplots with a nonlinear power-law fit of the form $y = a \cdot x^b + c$.

Across all three datasets, we observe a consistent negative relationship between motion and predictivity: subjects who moved more (higher mean FD) exhibited lower model performance (Figure 5). The effect is most pronounced in datasets with wider motion ranges (Narratives, Little Prince), and is present though slightly reduced in LeBel (fewer subjects, narrower FD range).

These results align with the known impact of motion on BOLD signal quality [30], and they indicate that motion accounts for a meaningful fraction of the between-subject variance in encoding scores. Motion-related variance can confound individual differences, which underscores the need for careful quality control of the neural data that serves as the encoding model target.

5 Conclusion

LITcoder is a flexible open-source library for building and evaluating neural encoding models. By standardizing core components (stimulus-brain alignment, feature extraction, downsampling, mapping, and evaluation) while remaining modular and extensible, LITcoder lowers the barriers to reproducible analysis and makes methodological choices transparent. Experiments across three story listening fMRI datasets showed how decisions about downsampling, FIR window length, cross-validation strategy, and subject motion substantially impact reported predictivity, underscoring the need for principled pipelines. LITcoder therefore provides a toolkit for systematic comparisons across models, datasets, and brain regions and an infrastructure for methodological rigor.

Acknowledgments

References

[1] Daniel L. K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, June 2014. doi: 10.1073/pnas.1403112111. URL `https://www.pnas.org/doi/10.1073/pnas.1403112111`. Publisher: Proceedings of the National Academy of Sciences.

[2] U. Guclu and M. A. J. Van Gerven. Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral Stream. *Journal of Neuroscience*, 35(27):10005–10014, July 2015. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.5023-14.2015. URL `https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.5023-14.2015`.

[3] N. Apurva Ratan Murty, Pouya Bashivan, Alex Abate, James J. DiCarlo, and Nancy Kanwisher. Computational models of category-selective brain regions enable high-throughput tests of selectivity. *Nature Communications*, 12(1):5540, September 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-25409-6. URL `https://www.nature.com/articles/s41467-021-25409-6`. Publisher: Nature Publishing Group.

[4] Kendrick N. Kay, Thomas Naselaris, Ryan J. Prenger, and Jack L. Gallant. Identifying natural images from human brain activity. *Nature*, 452(7185):352–355, March 2008. ISSN 1476-4687. doi: 10.1038/nature06713. URL `https://www.nature.com/articles/nature06713`. Publisher: Nature Publishing Group.

[5] Seyed-Mahdi Khaligh-Razavi and Nikolaus Kriegeskorte. Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation. *PLOS Computational Biology*, 10(11):e1003915, November 2014. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1003915. URL `https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003915`. Publisher: Public Library of Science.

[6] Martin Schrimpf, Idan Asher Blank, Greta Tuckute, Carina Kauf, Eghbal A. Hosseini, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. The neural architecture of language: Integrative modeling converges on predictive processing. *Proceedings of the National Academy of Sciences*, 118(45):e2105646118, November 2021. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2105646118. URL `https://pnas.org/doi/full/10.1073/pnas.2105646118`.

[7] Alexander G. Huth, Wendy A. de Heer, Thomas L. Griffiths, Frédéric E. Theunissen, and Jack L. Gallant. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532(7600):453–458, April 2016. ISSN 0028-0836. doi: 10.1038/nature17637. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4852309/`.

[8] Richard Antonello, Aditya Vaidya, and Alexander G. Huth. Scaling laws for language encoding models in fMRI, January 2024. URL `http://arxiv.org/abs/2305.11863`. arXiv:2305.11863 [cs].

[9] Charlotte Caucheteux and Jean-Rémi King. Brains and algorithms partially converge in natural language processing. *Communications Biology*, 5(1):1–10, February 2022. ISSN 2399-3642. doi: 10.1038/s42003-022-03036-1. URL `https://www.nature.com/articles/s42003-022-03036-1`. Publisher: Nature Publishing Group.

[10] Mariya Toneva and Leila Wehbe. Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain), November 2019. URL `http://arxiv.org/abs/1905.11833`. arXiv:1905.11833 [cs].

[11] Refael Tikochinski, Ariel Goldstein, Yoav Meiri, Uri Hasson, and Roi Reichart. Incremental accumulation of linguistic context in artificial and biological neural networks. *Nature Communications*, 16(1):803, January 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-56162-9. URL `https://www.nature.com/articles/s41467-025-56162-9`. Publisher: Nature Publishing Group.

[12] Badr AlKhamissi, Greta Tuckute, Yingtian Tang, Taha Binhuraib, Antoine Bosselut, and Martin Schrimpf. From Language to Cognition: How LLMs Outgrow the Human Language Network, March 2025. URL `http://arxiv.org/abs/2503.01830`. arXiv:2503.01830 [cs].

[13] Gabriel Kressin Palacios, Zhuoyang Li, and Kristijan Armeni. Executable science: Research software engineering practices for replicating neuroscience findings. In *Proceedings of the ACM Conference on Reproducibility and Replicability (ACM REP '25)*, pages 1–14, Vancouver, BC, Canada, 2025. Association for Computing Machinery. doi: 10.1145/3736731.3746147. URL `https://doi.org/10.1145/3736731.3746147`. ACM, New York, NY, USA. 14 Pages.

[14] Nima Hadidi, Ebrahim Feghhi, Bryan H. Song, Idan A. Blank, and Jonathan C. Kao. Illusions of Alignment Between Large Language Models and Brains Emerge From Fragile Methods and Overlooked Confounds, March 2025. URL `https://www.biorxiv.org/content/10.1101/2025.03.09.642245v1`. Pages: 2025.03.09.642245 Section: Contradictory Results.

[15] Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, Kailyn Schmidt, Daniel L. K. Yamins, and James J. DiCarlo. Brain-Score: Which Artificial Neural Network for Object Recognition is most Brain-Like?, September 2018. URL `http://biorxiv.org/lookup/doi/10.1101/407007`.

[16] Subba Reddy Oota, Manish Gupta, and Mariya Toneva. Joint processing of linguistic properties in brains and language models, November 2023. URL `http://arxiv.org/abs/2212.08094`. arXiv:2212.08094 [cs].

[17] Idan Blank, Nancy Kanwisher, and Evelina Fedorenko. A functional dissociation between language and multiple-demand systems revealed in patterns of BOLD signal fluctuations. *Journal of Neurophysiology*, 112(5):1105–1118, September 2014. ISSN 0022-3077, 1522-1598. doi: 10.1152/jn.00884.2013. URL `https://www.physiology.org/doi/10.1152/jn.00884.2013`.

[18] Francisco Pereira, Bin Lou, Brianna Pritchett, Samuel Ritter, Samuel J. Gershman, Nancy Kanwisher, Matthew Botvinick, and Evelina Fedorenko. Toward a universal decoder of linguistic meaning from brain activation. *Nature Communications*, 9(1):963, March 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-03068-4. URL `https://www.nature.com/articles/s41467-018-03068-4`. Publisher: Nature Publishing Group.

[19] Charlotte Caucheteux, Alexandre Gramfort, and Jean-Rémi King. Evidence of a predictive coding hierarchy in the human brain listening to speech. *Nature Human Behaviour*, 7(3): 430–441, March 2023. ISSN 2397-3374. doi: 10.1038/s41562-022-01516-2. URL `https://www.nature.com/articles/s41562-022-01516-2`. Publisher: Nature Publishing Group.

[20] Matthew F Glasser, Timothy S Coalson, Emma C Robinson, Carl D Hacker, John Harwell, Essa Yacoub, Kamil Ugurbil, Jesper Andersson, Christian F Beckmann, Mark Jenkinson, Stephen M Smith, and David C Van Essen. A multi-modal parcellation of human cerebral cortex. *Nature*, 536(7615):171–178, August 2016. ISSN 0028-0836. doi: 10.1038/nature18933. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4990127/`.

[21] Shailee Jain and Alexander Huth. Incorporating Context into Language Encoding Models for fMRI. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/hash/f471223d1a1614b58a7dc45c9d01df19-Abstract.html`.

[22] Alexander G. Huth, Shinji Nishimoto, An T. Vu, and Jack L. Gallant. A Continuous Semantic Space Describes the Representation of Thousands of Object and Action Categories across the Human Brain. *Neuron*, 76(6):1210–1224, December 2012. ISSN 0896-6273. doi: 10.1016/j.neuron.2012.10.014. URL `https://www.cell.com/neuron/abstract/S0896-6273(12)00934-8`. Publisher: Elsevier.

[23] Alexandre Abraham, Fabian Pedregosa, Michael Eickenberg, Philippe Gervais, Andreas Mueller, Jean Kossaifi, Alexandre Gramfort, Bertrand Thirion, and Gael Varoquaux. Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, 8, February 2014. ISSN 1662-5196. doi: 10.3389/fninf.2014.00014. URL `https://www.frontiersin.org/journals/neuroinformatics/articles/10.3389/fninf.2014.00014/full`. Publisher: Frontiers.

[24] Ahmad Samara, Zaid Zada, Tamara Vanderwal, Uri Hasson, and Samuel A. Nastase. Cortical language areas are coupled via a soft hierarchy of model-based linguistic features. *bioRxiv: The Preprint Server for Biology*, page 2025.06.02.657491, June 2025. ISSN 2692-8205. doi: 10.1101/2025.06.02.657491.

[25] Benjamin Lipkin, Greta Tuckute, Josef Affourtit, Hannah Small, Zachary Mineroff, Hope Kean, Olessia Jouravlev, Lara Rakocevic, Brianna Pritchett, Matthew Siegelman, Caitlyn Hoeflin, Alvincé Pongos, Idan A. Blank, Melissa Kline Struhl, Anna Ivanova, Steven Shannon, Aalok Sathe, Malte Hoffmann, Alfonso Nieto-Castañón, and Evelina Fedorenko. Probabilistic atlas for the language network based on precision fMRI data from >800 individuals. *Scientific Data*, 9(1):529, August 2022. ISSN 2052-4463. doi: 10.1038/s41597-022-01645-3. URL `https://www.nature.com/articles/s41597-022-01645-3`. Publisher: Nature Publishing Group.

[26] Samuel A. Nastase, Yun-Fei Liu, Hanna Hillman, Asieh Zadbood, Liat Hasenfratz, Neggin Keshavarzian, Janice Chen, Christopher J. Honey, Yaara Yeshurun, Mor Regev, Mai Nguyen, Claire H. C. Chang, Christopher Baldassano, Olga Lositsky, Erez Simony, Michael A. Chow, Yuan Chang Leong, Paula P. Brooks, Emily Micciche, Gina Choe, Ariel Goldstein, Tamara Vanderwal, Yaroslav O. Halchenko, Kenneth A. Norman, and Uri Hasson. The "Narratives" fMRI dataset for evaluating models of naturalistic language comprehension. *Scientific Data*, 8 (1):250, September 2021. ISSN 2052-4463. doi: 10.1038/s41597-021-01033-3. URL `https://www.nature.com/articles/s41597-021-01033-3`. Publisher: Nature Publishing Group.

[27] Jixing Li, Shohini Bhattasali, Shulin Zhang, Berta Franzluebbers, Wen-Ming Luh, R. Nathan Spreng, Jonathan R. Brennan, Yiming Yang, Christophe Pallier, and John Hale. Le Petit Prince multilingual naturalistic fMRI corpus. *Scientific Data*, 9(1):530, August 2022. ISSN 2052-4463. doi: 10.1038/s41597-022-01625-7. URL `https://www.nature.com/articles/s41597-022-01625-7`. Publisher: Nature Publishing Group.

[28] Amanda LeBel, Lauren Wagner, Shailee Jain, Aneesh Adhikari-Desai, Bhavin Gupta, Allyson Morgenthal, Jerry Tang, Lixiang Xu, and Alexander G. Huth. A natural language fMRI dataset for voxelwise encoding models. *Scientific Data*, 10(1):555, August 2023. ISSN 2052-4463. doi: 10.1038/s41597-023-02437-z. URL `https://www.nature.com/articles/s41597-023-02437-z`. Publisher: Nature Publishing Group.

[29] Oscar Esteban, Christopher J. Markiewicz, Ross W. Blair, Craig A. Moodie, A. Ilkay Isik, Asier Erramuzpe, James D. Kent, Mathias Goncalves, Elizabeth DuPre, Madeleine Snyder, Hiroyuki Oya, Satrajit S. Ghosh, Jessey Wright, Joke Durnez, Russell A. Poldrack, and Krzysztof J. Gorgolewski. fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nature Methods*, 16(1):111–116, January 2019. ISSN 1548-7105. doi: 10.1038/s41592-018-0235-4. URL `https://www.nature.com/articles/s41592-018-0235-4`. Publisher: Nature Publishing Group.

[30] Jonathan D Power, Kelly A Barnes, Abraham Z Snyder, Bradley L Schlaggar, and Steven E Petersen. Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion. *Neuroimage*, 59(3):2142–2154, February 2012. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2011.10.018. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3254728/`.

A Temporal Baseline Features

To quantify the extent to which temporal autocorrelation alone can drive predictivity (independent of stimulus content), we construct a *content-agnostic temporal baseline*. The baseline assigns each stimulus timepoint a feature vector whose similarity depends only on temporal proximity.

Construction. Let n be the number of stimulus timepoints (e.g., tokens or TR-aligned bins). We define an $n \times n$ temporal autocorrelation matrix A with entries

$$A_{ij} = \exp\Big(-\frac{|i-j|}{\ell} \Big),$$

where $\ell > 0$ is a correlation length (in stimulus units). This kernel induces exponentially decaying similarity as a function of temporal distance. Figure 6 illustrates this kernel for two choices of ℓ.

We compute the truncated SVD of A,

$$A = U\Sigma V^\top,$$

and form d-dimensional baseline features by taking the top d left singular vectors scaled by the square roots of their singular values:

$$F = U_{[:,1:d]}\, \Sigma_{[1:d,1:d]}^{1/2} \in \mathbb{R}^{n\times d}.$$

Each row F_t is the baseline feature for timepoint t. By construction, pairwise similarities of $\{F_t\}$ reflect only temporal distance; content is never used.

Hyperparameters. The correlation length ℓ controls the decay rate of temporal similarity (larger ℓ $\Rightarrow$ longer-range correlation). The feature width d controls the rank/complexity of the baseline.

Use in analyses. These features serve as *autocorrelation controls*: training an encoding model on F estimates the upper bound of predictivity obtainable from temporal structure alone. Comparing model scores to this baseline helps detect temporal leakage and overoptimistic cross-validation (cf. Section 4.3).

A.1 Extracting Brain Regions (V1, A1/AC1, and Language Network)

We derive cortical ROIs on the `fsaverage` template from two sources: the HCP–MMP1 (Glasser) atlas for V1 and A1/AC1 (`lh.HCP-MMP1.annot`, `rh.HCP-MMP1.annot`), and the LanA probabilistic language atlas for the language network [25].

V1 and A1/AC1. From the Glasser annotations, we select parcels that correspond to V1 and A1/AC1. The resulting parcel indices are converted into boolean masks per hemisphere and then concatenated into whole-cortex masks on `fsaverage5`.

Language network (LanA). LanA provides a per-vertex probability of language selectivity [25]. After transferring the LanA map to `fsaverage5`, we define the language ROI as the *top 10% highest-probability vertices* (default setting used throughout). This produces a reproducible, thresholded mask for the language network.

B   Model Specifications

We evaluated both language models (via TransformerLens) and speech models (via Hugging Face). For **language models**, we relied on the `HookedTransformer` interface from Transformer-Lens to expose internal activations. Unless otherwise specified, features were extracted from the `blocks.{L}.hook_resid_pre` hook (pre-MLP residual stream) at a given layer L. By default, we used the representation of the *last token* for any given input string, though the framework also supports mean-over-tokens pooling. The models included several members of the Pythia family as well as GPT-2 small (Table 2).

For **speech models**, we used Hugging Face implementations of OpenAI Whisper (Tiny) and HuBERT (Base). In Whisper, which has 4 encoder and 4 decoder transformer blocks, we extracted features from the final *encoder* layer only. In HuBERT, which has a 12-layer transformer encoder, we extracted features from the final encoder layer. These choices follow prior work using speech representations as predictors of fMRI activity (Table 3).

C   Dataset-Based Evaluation Protocols

Evaluation procedures were dataset-specific to align with established practices in the literature. For the **Narratives** [26] dataset, following [16], we employed k-fold cross-validation with boundary trimming to mitigate autocorrelation effects.

For the **Little Prince** [27] dataset, we excluded run 1 because the first chapter incorporated visual cues (drawings) to highlight the story's theme of differences between adults and children [27]. These
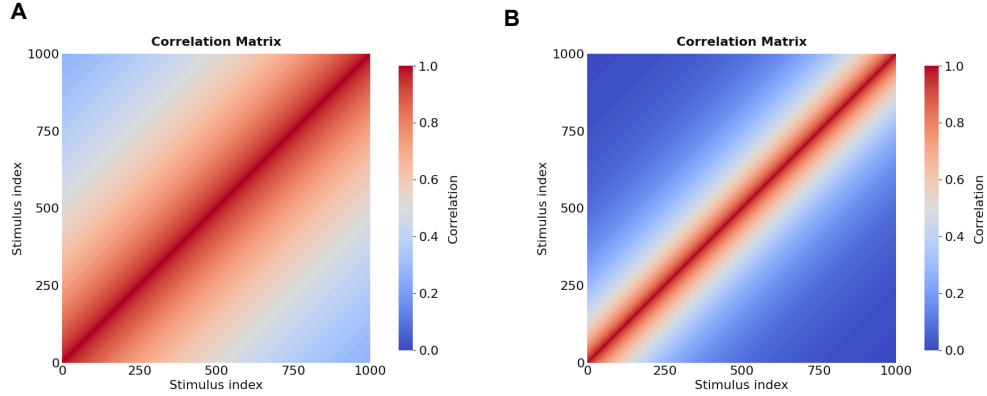
Figure 6: **Temporal autocorrelation kernels used to construct content-agnostic baselines.** Each panel shows an example of the exponential kernel $A_{ij} = \exp(-|i - j|/\ell)$ that defines similarity between timepoints. **(A)** With a longer correlation length ($\ell = 750$), similarity decays slowly, producing a broad diagonal band (long-range temporal correlation). **(B)** With a shorter correlation length ($\ell = 300$), similarity decays faster, yielding a narrower band (short-range correlation). These kernels drive the Autocorrelation Control Vectors used in Section 4.3: larger ℓ sets a stronger/longer-range temporal baseline, smaller ℓ a weaker/shorter-range one.

Table 2: **Language models used and TransformerLens hook.** Unless otherwise noted, we extract features from `blocks.{L}.hook_resid_pre` (pre-MLP residual stream) at a specified layer L, using the last token within each text segment (configurable to mean-over-tokens).

Family	HF ID	Layers	Hook (TransformerLens)	Token pooling
Pythia	EleutherAI/pythia-14m	6	blocks.{L}.hook_resid_pre	last token
Pythia	EleutherAI/pythia-31m	6	blocks.{L}.hook_resid_pre	last token
Pythia	EleutherAI/pythia-70m	6	blocks.{L}.hook_resid_pre	last token
Pythia	EleutherAI/pythia-160m	12	blocks.{L}.hook_resid_pre	last token
Pythia	EleutherAI/pythia-410m	24	blocks.{L}.hook_resid_pre	last token
Pythia	EleutherAI/pythia-1b	16	blocks.{L}.hook_resid_pre	last token
GPT-2	gpt2 ("gpt2-small")	12	blocks.{L}.hook_resid_pre	last token

Notes: The hook type is configurable in our code (`hook_type`), defaulting to `hook_resid_pre`. We also support mean-over-tokens pooling by setting `last_token=False`.

Table 3: **Speech models used (Hugging Face).** We extract encoder-side representations for mapping to fMRI.

Family	HF ID	Architecture (layers)	Feature layer used
Whisper (Tiny)	openai/whisper-tiny	Encoder 4, Decoder 4	Final *encoder* layer
HuBERT (Base)	facebook/hubert-base-ls960	Encoder 12	Final encoder layer

additional stimuli introduce visual confounds not present in later runs; excluding run 1 therefore ensures that our analyses focus on auditory comprehension and remain comparable across runs. For runs 2-9, we trained encoding models on each run separately using k-fold cross-validation with boundary trimming. This approach follows the design philosophy outlined in [11], where neural encoder models are trained individually on each story within the Narratives dataset.

For the larger-scale **LeBel** [28] dataset, we used held-out evaluation: models were trained on a subset of stories and evaluated on an entirely separate story to assess generalization. Following the evaluation protocol in LeBel et al. [28], we used the *wheretheressmoke* story as our test story, which was presented once in each of the five scanning sessions. As recommended in the original paper, we averaged responses across these five repetitions to increase signal-to-noise ratio and obtain less biased estimates of model performance.

D   Data Processing and Selection

We report here the dataset-specific preprocessing steps applied prior to model fitting. These include temporal trimming of runs and motion-based subject exclusion.

D.1   Temporal Trimming

Trimming the beginning and end of fMRI runs is standard practice in naturalistic neuroimaging due to several sources of noise and artifact [28]. Additionally, many experimental protocols include silent periods or delays that contain no stimulus content relevant to the cognitive processes of interest.

For the Narratives dataset, we removed the first 14 TRs because they are silent, and the last 9 TRs, following [16].

For the Little Prince dataset, each run began with a 4 TR silent period between the trigger and the audiobook onset [27]. Beyond removing these silent periods, we applied additional trimming of the first 5 and last 5 TRs.

For the LeBel dataset, we followed the original protocol, removing 10 TRs from the beginning and end of each story. This eliminated the 10-second silent periods as well as some portion of the story. In addition, following [8] we removed the first 50 TRs from the test story to eliminate long context artifacts.

D.2   Subject Exclusion Based on Motion

Consistent with our finding that head motion is negatively correlated with encoding model performance (Section 4.5), we excluded participants whose mean framewise displacement (FD) exceeded 0.2 mm. No participants were excluded from the LeBel dataset. The excluded subjects for the other datasets are listed below.

Table 4: Subjects excluded based on mean FD > 0.2 mm.

Dataset	Subjects Excluded	# Excluded
LeBel	None	0
Narratives	Sub-268, Sub-266, Sub-259, Sub-254	4
Little Prince	Sub-EN099, Sub-EN075, Sub-EN097, Sub-EN093	4