DeMEGa: Decoding Listened Phonemes via MEG Signals by Stacking Conformers with DeBERTa-Style Disentangled Attention

Ihor Stepanov , Aleksandr Smechov , Alexander Yavorskyi , and Shivam Chaudhary
September Labs (S8L.io)

November 9, 2025

ABSTRACT

Magnetoencephalography (MEG), first demonstrated in 1968, is a cleaner—albeit much bulkier and more expensive—alternative to Electroencephalography (EEG) for non-invasive brain signal measurement. MEG and EEG have historically suffered from poor signal-to-noise ratio (SNR) and limited data, both at the individual level and across participants, which makes high-precision tasks such as attempted, imagined, or listened speech detection extremely challenging. In 2025, the PNPL team at the University of Oxford released LibriBrain, the largest listened-phoneme dataset to date, with > 50 hours of high-quality MEG of a single subject listening to audiobooks. LibriBrain enables rigorous validation of methods for decoding listened speech.

For the PNPL 2025 competition, we introduce *DeMEGa* (Disentangled MEG Attention), a three-stage stacking pipeline that leverages multiple grouped-averaging levels of MEG data, Conformers with DeBERTa-style disentangled attention, IPA (International Phonetic Alphabet) feature supervision, and loss functions tailored to severe class imbalance and low SNR. Our stack achieves a macro-F1 of 0.62 on an the competition holdout set. The design also supports adding external MEG data for further gains. We detail the methodology and provide an end-to-end, reproducible pipeline on GitHub [link].

Keywords MEG · phoneme classification · stacking · grouped averaging · disentangled attention · Conformer · focal loss · supervised contrastive · IPA features · PNPL · LibriBrain

1 Introduction

1.1 Problem and motivation

Listened phoneme decoding from MEG is challenging due to low SNR, short time windows, and severe class imbalance. While less directly practical than attempted or imagined speech decoding, listened phoneme classification remains an important stepping stone toward non-invasive communication aids.

Although "decoding thoughts" can sound like telepathy, brain-signal decoding has clear potential to improve the lives of people who have lost the ability to communicate. Invasive BMIs often yield higher fidelity by measuring closer to neuronal sources, whereas non-invasive BCIs (MEG/EEG) face limited SNR and comparatively less hardware innovation [Frontiers in Neuroscience, 2016]. Nonetheless, non-invasive approaches are critical for patients who cannot or will not undergo surgery.

We view the challenge as resting on three pillars: Data, Models, and Hardware. LibriBrain contributes data scale; here we focus on models, introducing DeMEGa, a three-stage stacking pipeline that converts preprocessing diversity into a learning signal.

1.2 Core idea

Stage 1 trains several base learners on identical LibriBrain splits but at different grouped-averaging levels, so each model realizes a distinct SNR/data-count trade-off. Instead of naïvely averaging probabilities, we convert base outputs to logits and learn a shallow meta-model (Stage 3) that mixes them per class. Thus the stack can lean on high-K models when stability helps and on low-K models when fine detail matters. The base learner is a Conformer with DeBERTa-style disentangled attention to separate *what* happens from *when* it happens. This echoes multi-scale ideas in speaker diarization, where complementary temporal views aid downstream decisions [**?**].

1.3 Contributions

- **DeMEGa Stack:** A three-stage framework for MEG phoneme decoding that consistently improves macro-F1 over the best single base model and over probability-averaging ensembles.
- **Strong base learner:** Conformer with DeBERTa-style disentangled attention, present-class averaged focal loss, supervised NT-Xent, and an auxiliary IPA-feature head.
- **Logit-level stacking:** Meta-features from concatenated logits (not softmax), preserving margins and calibration cues for the stacker.
- **Practical recipe:** Median-IQR normalization, temperature-scaled class reweighting, and strict sample alignment across grouping levels for clean meta-datasets.
- **Inference add-ons:** Optional db4 wavelet denoising that improves holdout performance; standard test-time augmentation did not help here.

2 Related Work

2.1 MEG/EEG speech decoding

Recent studies demonstrate that deep learning can decode speech from non-invasive recordings. Défossez et al. [2023] achieve 41% accuracy on speech-segment identification from MEG/EEG via contrastive learning; Gwilliams et al. [2022] show the brain processes multiple phonemes simultaneously with position-invariant encoding; Dash et al. [2020] decode imagined phrases from MEG using CNNs. Sensor-space deep models increasingly outperform traditional source-reconstruction pipelines.

2.2 Conformer architectures for time series

The Conformer marries convolutions for local patterns with self-attention for global dependencies and excels in ASR [Gulati et al., 2020]. Adaptations to biosignals (e.g., EEG) show benefits over pure CNNs on motor imagery and related tasks [Anonymous, 2023].

2.3 DeBERTa disentangled attention

DeBERTa separates content and position into distinct vectors, allowing it to learn more complex positional patterns bound to content information. [He et al., 2021] achieved remarkable results with a disentangled attention-based model, beating human performance on SuperGLUE by computing attention with content-to-content, content-to-position, and position-to-content components. [Shaw et al., 2018] established that the use of relative positions can be even more beneficial for attention as compared to the classical approach. For MEG, separating what (content) from when (position) matches how phonemes are encoded in temporal sequences, moreover, due to relative positional encoding, representations are less dependent on global positional information, making them more stable to temporal perturbations.

2.4 Class imbalance techniques

Phoneme distributions are severely imbalanced. Focal loss down-weights easy examples [Lin et al., 2017]; class-balanced losses reweight by effective sample count [Cui et al., 2019]. Our present-class averaging addresses per-batch class sparsity common in phoneme data.

2.5 Stacking and ensembles

Stacking learns to combine multiple models by training a meta-learner on their outputs. [Corsi et al., 2022] demonstrated that combining phoneme and articulatory features can enhance both articulatory feature estimation and phoneme recognition. We use IPA features as auxiliary targets to inject this inductive bias.

2.6 Articulatory features as auxiliary supervision

The superior temporal gyrus encodes phonetic features (e.g., voicing, place) [Mesgarani et al., 2014]. Joint training with articulatory targets aids phoneme recognition through inductive bias and positive transfer [Rasipuram and Magimai-Doss, 2011].

2.7 Wavelet denoising

Wavelet analysis is a well-known and efficient technique used across various signal data modalities, including the EEG, spanning for more than 20 years such as in [Hazarika et al., 1997], where the authors used it along with neural networks for the classification task. In our case, we lean heavily towards one of the classical wavelet use-cases, which is denoising, similar to how it was used in [Patil and Pawar, 2012] for the EEG signals.

3 Dataset and Grouped Preprocessing

3.1 LibriBrain dataset

We build and evaluate on LibriBrain: $\sim$52.3 h of within-subject MEG while a single participant listens to Sherlock Holmes audiobooks, with dense time-locked word/phoneme annotations. Recordings were obtained on a 306-channel Elekta/MEGIN TRIUX system and minimally preprocessed by the authors. The release ships with standard train/val/test splits and a PNPL Python API [PNPL, 2025a]. We additionally modified this library for faster pre-averaged HDF5 loading [Labs, 2025].

Spatial snapshots across grouped-averaging levels (e.g., $K \in \{5, 10, 25, 50, 100\}$) reveal class imbalance: the top-5 phonemes comprise $\approx$35% of all labels; the rarest (`ZH`) appears only 100 times, while the most common (`AH`) appears 16,471 times. Median per-class count is 3,256 (mean 4,333; SD 3,607). Imbalance persists across audio books (some double the counts of others).

Sensors and sampling. The TRIUX system uses 204 planar gradiometers and 102 magnetometers (306 sensors). Raw data were sampled at 1 kHz and downsampled to 250 Hz; we use 4 ms per sample.

Splits and format. Data are distributed as one HDF5 and one event TSV per session, with 91 training sessions ($\sim$51.6 h), 1 validation ($\sim$0.36 h), and 1 test ($\sim$0.38 h). Additional held-out sessions are reserved for competition evaluation.

Labels. We follow the competition convention: 39 CMU/ARPAbet phoneme classes. Event files also include B/I/E/S tags; we retain them as metadata but train/evaluate on the 39-class mapping unless noted.

Windowing. Unless stated otherwise, each example is a $[0.0, 0.5]$ s window (125 time points at 250 Hz) starting at phoneme onset. The sample tensor shape is channels $= 306$ by time $= 125$.

Access. The dataset is hosted on Hugging Face under `pnpl/LibriBrain` [PNPL, 2025b]; the PNPL library is available via PyPI [PNPL, 2025a].

3.2 Grouped averaging

A single 500 ms snippet aligned to a phoneme has low SNR. Grouped averaging is therefore first-class in DeMEGa. For class c and group size K, we construct sensor-space averages

$$\bar{x}^{(c)} = \frac{1}{K} \sum_{k=1}^{K} x_k^{(c)} \in \mathbb{R}^{306 \times 125}.$$

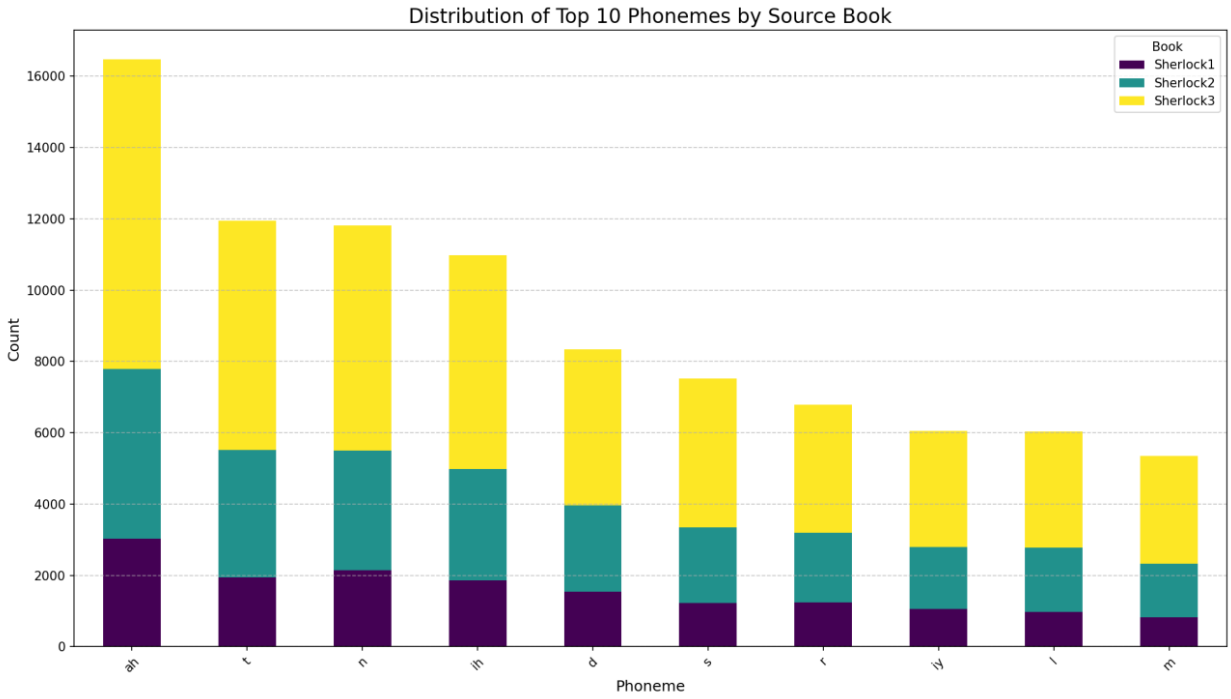

Figure 1: Top-10 most common phonemes per book

Windows share $t_{\min} = 0.0$, $t_{\max} = 0.5$, onset-locked. We group by the 39-class ARPAbet label; B/I/E/S can be enforced for stricter homogeneity (off by default). If a class has fewer than K instances, we use K_{eff} and record it for calibration analysis.

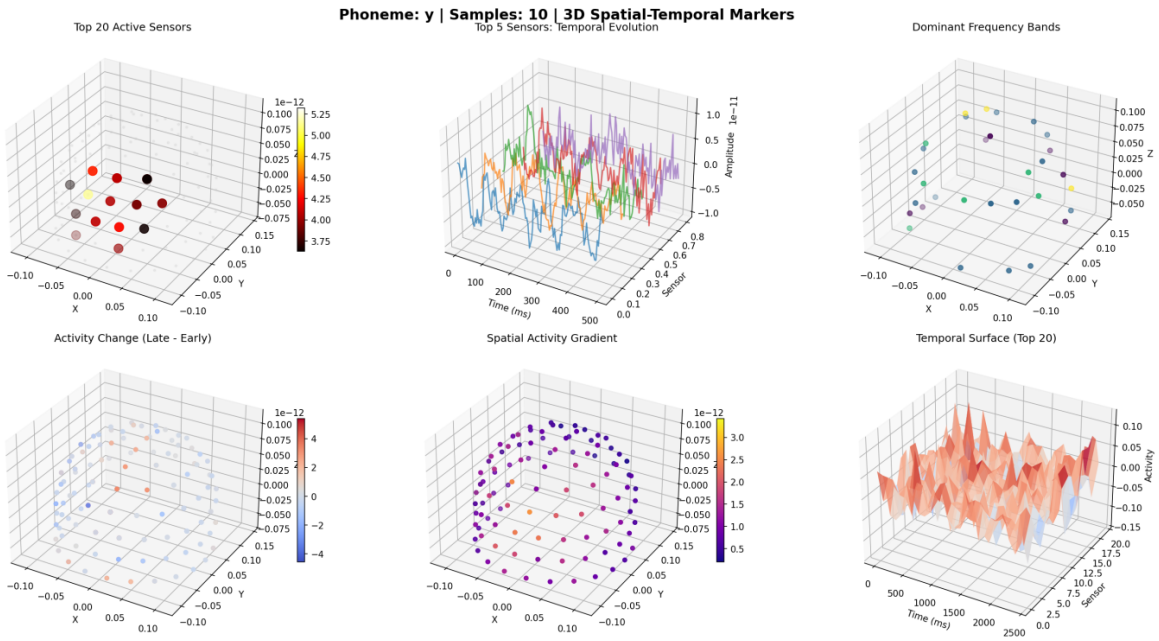

Figure 2: Y-phoneme spatial analysis at $K = 10$ (higher variability; more detail, lower SNR).

Why multiple K's? High K improves SNR but risks over-smoothing fast transitions; low K yields noisier but more diverse samples. Rather than picking one K, we train distinct bases for several K and stack their logits so the meta-model can learn class-specific mixtures. This is partly inspired by multi-scale diarization [Park et al., 2022].

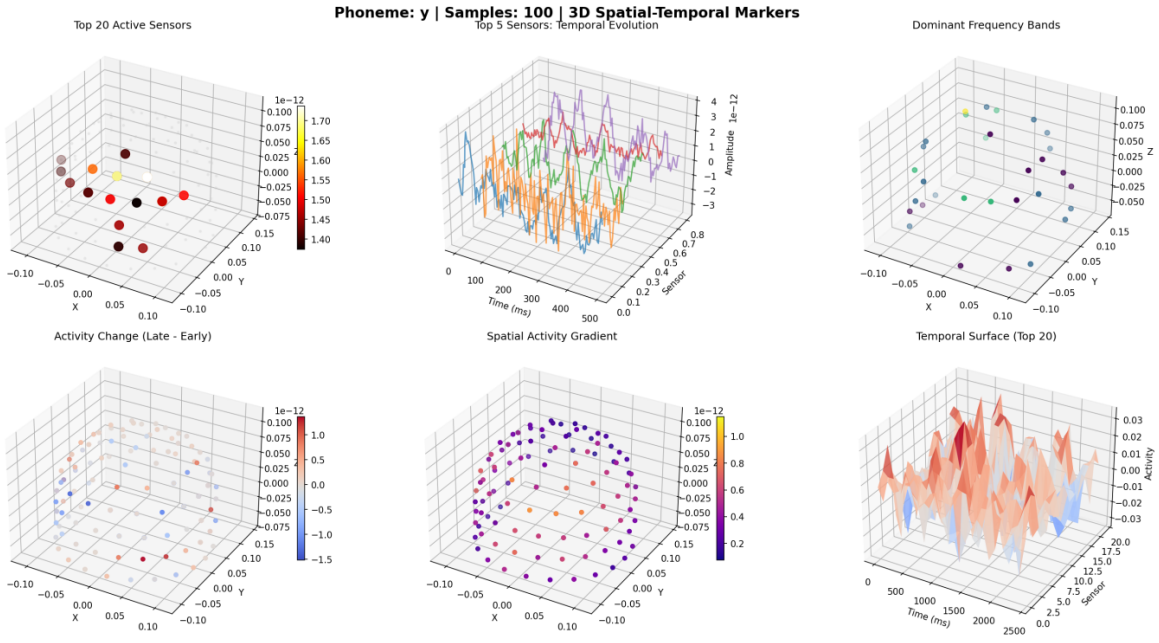

Figure 3: Y-phoneme spatial analysis at $K = 100$ (stable spatial pattern; higher SNR, more smoothing).

What has worked for us. Robust per-channel median-IQR normalization; class-balanced batching for low K, uniform shuffles for high K; careful bookkeeping so per-K predictions align row-wise when concatenated; evaluation parity with $K = 100$ where feasible (using K_{eff} masks for rares).

3.3 Extended track and external data

For the PNPL extended track [PNPL, 2025c], we explored external MEG corpora: Radboud listened-speech MEG [University, 2022] and MEG-SCANS (German) [NEMAR, 2025]. Cross-dataset transfer is difficult due to sensor variability, preprocessing differences, and inter-subject factors; domain adaptation and pseudo-labels gave modest but real gains: $0.58 \rightarrow 0.62$ macro-F1 (relative $\sim$7%). Success requires careful alignment and normalization rather than naïve pooling.

4 Method

4.1 DeMEGa architecture

DeMEGa targets MEG phoneme classification with $\approx$5.2M parameters and four components:

Input processing. A residual input projection maps 306×125 tensors to hidden size $H = 128$: $\text{Conv1d } 306 \rightarrow H$ (kernel 5) $\rightarrow$ SiLU $\rightarrow$ $\text{Conv1d } H \rightarrow H$ (kernel 3), plus a skip $\text{Conv1d } 306 \rightarrow H$ (kernel 1).

MEG encoder. The encoder stacks $N = 4$ Conformer layers. Each layer uses (i) a depthwise separable convolution module; (ii) DeBERTa-style disentangled self-attention with content$\leftrightarrow$position terms and 32 relative-position buckets (max distance 128); and (iii) a SiLU feed-forward network. Pre-norm and residual connections wrap each sub-module; the overall flow is: Norm $\rightarrow$ Conv $\rightarrow$ Norm $\rightarrow$ Attention $\rightarrow$ Norm $\rightarrow$ FFN, with residuals around each block.

Feature aggregation. Temporal outputs are flattened and passed through an MLP:

$$\text{Dropout} \rightarrow \text{Linear}(H \times T \rightarrow 2d_{\text{emb}}) \rightarrow \text{SiLU} \rightarrow \text{Linear}(2d_{\text{emb}} \rightarrow d_{\text{emb}}) \rightarrow \text{Dropout},$$

with $d_{\text{emb}} = 128$.

Prediction heads. (1) A 39-way classifier; (2) an IPA-feature head predicting 14 binary articulatory features (consonantal, syllabic, sonorant, voice, nasal, continuant, labial, coronal, dorsal, high, low, back, round, diphthong) per sample; (3) a projection head for supervised contrastive learning.

Signal normalization. We apply median-IQR scaling per channel for robustness:

$$y = \frac{x - \text{median}(x)}{\text{IQR}(x) + \varepsilon}, \qquad \varepsilon = 10^{-6}.$$

4.2 Training objective

The total loss combines three terms.

Present-class averaged focal loss. Let logits $z \in \mathbb{R}^{N \times C}$, probabilities $p = \text{softmax}(z)$, and labels $y \in \{1, \ldots, C\}^N$. With label smoothing $\tilde{y}_{ij}$, class weights α_j, and focusing $\gamma = 2.0$,

$$L_{\text{focal}} = \frac{1}{|C_{\text{present}}|} \sum_{c \in C_{\text{present}}} \frac{1}{|I_c|} \sum_{i \in I_c} \sum_{j=1}^{C} \tilde{y}_{ij}\, \alpha_j\, (1 - p_{ij})^{\gamma}\, (-\log p_{ij}),$$

where $I_c = \{i : y_i = c\}$ and $C_{\text{present}} = \{c : |I_c| > 0\}$. We compute α_c via temperature-scaled frequency priors, clamped to $[0.5, 5.0]$.

Supervised NT-Xent. On L2-normalized projections u,

$$L_{\text{NT-Xent}} = -\frac{1}{|V|} \sum_{i \in V} \log \left(\frac{\sum_{j \in P(i)} \exp\big(\text{sim}(u_i, u_j)/\tau\big)}{\sum_{k \neq i} \exp\big(\text{sim}(u_i, u_k)/\tau\big)} \right), \quad \tau = 0.03,$$

with $V = \{i : \exists j \neq i,\ y_j = y_i\}$ and $P(i) = \{j \neq i :\ y_j = y_i\}$. Weight $= 0.5$.

IPA auxiliary loss.

$$L_{\text{IPA}} = \text{BCE}\big(\psi(h), F[y]\big),$$

where ψ is the IPA head, h the embedding, and $F[y] \in \{0, 1\}^{14}$ the feature vector. Weight $= 0.2$.

Total.

$$L_{\text{total}} = L_{\text{focal}} + 0.5\, L_{\text{NT-Xent}} + 0.2\, L_{\text{IPA}}.$$

4.3 Stage 1: Base models across grouping levels

We train separate bases per grouping level $K \in \{5, 10, 25, 50, 100\}$. Optimizer: AdamW (base LR 1e-4, weight decay 10^{-3}), cosine schedule with 5-epoch warmup, classifier LR multiplier 1.5, gradient clipping (max_norm $= 1.0$), up to 30 epochs with early stopping on validation macro-F1. Best checkpoints per K proceed to Stage 2. The differential LR helps convergence and mitigates catastrophic forgetting.

4.4 Stage 2: Meta-dataset construction

For each trained base m and split (train/val/test), we run aligned, unshuffled loaders to extract raw logits $L^{(m)} \in \mathbb{R}^{N \times 39}$. Meta-features are horizontal concatenations

$$X_{\text{train}} = \big[L^{(1)}_{\text{train}} \mid L^{(2)}_{\text{train}} \mid \cdots \mid L^{(M)}_{\text{train}}\big] \in \mathbb{R}^{N_{\text{train}} \times 39M},$$

and likewise for validation/test. We retain optional K_{eff} masks for rare classes. Note: Meta-training uses in-sample predictions, which can bias upward; we counter with regularization and multi-seed robustness in Stage 3.

4.5 Stage 3: Stacking classifier

The meta-learner is a shallow MLP:

$$\text{BatchNorm} \rightarrow \text{Linear}(39M \rightarrow 512) \rightarrow \text{SiLU} \rightarrow \text{BatchNorm} \rightarrow \text{Dropout}(0.5) \rightarrow \text{Linear}(512 \rightarrow 39).$$

We train with cross-entropy, AdamW, and ReduceLROnPlateau (factor 0.5, patience 5), early-stopping on validation macro-F1. We run $N = 5$ seeds and average softmax probabilities from the top-K runs.

4.6 Optional wavelet denoising

At inference, we optionally apply db4 wavelet denoising (level 3). Noise σ is estimated by MAD on the finest detail coefficients,

$$\sigma = \mathrm{median}(|\text{details}|)/0.6745, \qquad \lambda = \sigma\sqrt{2\log n},$$

scaled by a factor (0.6) and applied via soft thresholding before reconstruction. Signals are rescaled to preserve original mean/variance. This improves holdout performance but introduces a train-test distribution shift.

5 Limitations

Grouped averaging. Boosts SNR but can smear sub-phonemic transients; gains rely on complementary error profiles across K.

Windowing and causality. Using $[0.0, 0.5]$ s post-onset windows makes the system non-causal; streaming use requires latency-aware decoding.

Label scope. Collapsing to 39 phonemes and ignoring B/I/E/S loses positional/allophonic structure.

Calibration. Present-class focal loss improves rare-class recall but can degrade calibration; the meta-model may inherit this.

External generalization. Cross-dataset gains are modest and sensitive to preprocessing; English-centric pipeline.

Inference-only denoising. Denoising only at test time can shift distributions.

Architectural sensitivity. Disentangled attention bucket counts and kernel sizes are sensitive to window/sampling choices.

Evaluation scope. Single-subject, listened-speech macro-F1 does not capture robustness to task/environment changes.

6 Conclusion and Impact

We presented DeMEGa, a compact, reproducible pipeline that uses grouped-averaging diversity and logit-level stacking to improve non-invasive listened-phoneme decoding. The model combines Conformer backbones with DeBERTa-style attention, present-class averaged focal loss, supervised NT-Xent, and IPA features. While listened phoneme detection is niche, it is a useful stepping stone toward practical non-invasive speech prostheses. Code and configs are available for reproduction [link].

References

link. pnpl-2025-experiments (code, configs, and pipeline). `https://github.com/September-Labs/pnpl-2025-experiments`.

Frontiers in Neuroscience. Article discussing non-invasive bcis and hardware innovation. `https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2016.00295/full`, 2016.

Alexandre Défossez et al. Speech decoding from meg and eeg via contrastive learning. *Nature Machine Intelligence*, 2023. URL `https://www.nature.com/articles/s42256-023-00714-5`.

Laura Gwilliams et al. Position-invariant phoneme encoding in the brain. *Nature Communications*, 2022. URL `https://www.nature.com/articles/s41467-022-34326-1`.

Debadatta Dash et al. Decoding imagined phrases from meg with cnns. *Frontiers in Neuroscience*, 2020. URL `https://www.frontiersin.org/articles/10.3389/fnins.2020.00290/full`.

Anmol Gulati et al. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech*, 2020. URL `https://www.isca-archive.org/interspeech_2020/gulati20_interspeech.pdf`.

Anonymous. Adapting conformers for eeg time series. `https://ieeexplore.ieee.org/document/9991178/`, 2023.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2021. URL `https://arxiv.org/abs/2006.03654`.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations, 2018. URL `https://arxiv.org/abs/1803.02155`.

Tsung-Yi Lin et al. Focal loss for dense object detection, 2017. URL https://arxiv.org/abs/1708.02002.

Yin Cui et al. Class-balanced loss based on effective number of samples, 2019. URL https://arxiv.org/abs/1901.05555.

Marie-Constance Corsi et al. Ensembling diverse feature spaces for bci decoding. https://inria.hal.science/hal-03594331/document, 2022.

Nima Mesgarani et al. Phonetic feature encoding in human superior temporal gyrus. *Science*, 2014. URL https://www.science.org/doi/abs/10.1126/science.1245994.

R. Rasipuram and M. Magimai-Doss. Acoustic modeling with articulatory features. In *Statistical Language and Speech Processing*. 2011. URL https://link.springer.com/chapter/10.1007/978-3-642-21735-7_37.

Nirupam Hazarika, J. Z. Chen, A. C. Tsoi, and A. Sergejew. Classification of eeg signals using the wavelet transform. In *Proceedings of the First Joint BMES/EMBS Conference - Serving Humanity, Advancing Technology*. IEEE, 1997. URL https://dl.acm.org/doi/10.1145/1543834.1543860.

Suhas S. Patil and Minal K. Pawar. Quality advancement of eeg by wavelet denoising for biomedical analysis. In *2012 International Conference on Communication, Information & Computing Technology (ICCICT)*, Mumbai, India, October 2012. IEEE. doi:10.1109/ICCICT.2012.6398151. Suhas S. Patil, Department of Electronics, K. B. N. College of Engineering, Satara, Maharashtra, India; Minal K. Pawar, Department of Electronics, Sinhgad College of Engineering, Pune, Maharashtra, India.

PNPL. Pnpl python library (pypi). https://pypi.org/project/pnpl/, 2025a.

September Labs. September labs fork of pnpl. https://github.com/September-Labs/pnpl, 2025.

PNPL. Libribrain dataset on hugging face. https://huggingface.co/datasets/pnpl/LibriBrain, 2025b.

Tae Jin Park, Nithin Rao Koluguri, Jagadeesh Balam, and Boris Ginsburg. Multi-scale speaker diarization with dynamic scale weighting, 2022. URL https://arxiv.org/abs/2203.15974.

PNPL. 2025 libribrain competition tracks. https://neural-processing-lab.github.io/2025-libribrain-competition/tracks/, 2025c.

Radboud University. A listened-speech meg corpus. *Scientific Data*, 2022. URL https://www.nature.com/articles/s41597-022-01382-7.

NEMAR. Meg-scans dataset page (ds006468). https://nemar.org/dataexplorer/detail?dataset_id=ds006468, 2025.

Appendix

A. Implementation, Training Recipe, and Reproducibility

We release an end-to-end pipeline [link] mapping §4 to runnable code: repository layout and entry points (A.1), software/hardware (A.2), datasets (A.3), preprocessing (A.4), training stages (A.5–A.7), inference/submission (A.8), and logging and artifacts (A.9).

A.1 Repository map and entry points.

Models & components:
`experiments/demega/meg_classifier/models/meg_classifier.py`
`experiments/demega/meg_classifier/models/components/`

Configs:
`experiments/demega/configs/default_config.yaml`

Pipelines:
Stage 1 base learners: `experiments/demega/scripts/train.py`
Stacker (Stages 2–3): `experiments/demega/scripts/train_stacker_v2.py`
Evaluation: `experiments/demega/scripts/evaluate.py`
Submission / inference: `experiments/demega/scripts/generate_submission.py`

Data fetch:
`experiments/demega/download_data.py` (downloads pre-grouped HDF5 snapshots)

A.2 Software and hardware. PyTorch $\geq$ 2.0, Lightning $\geq$ 2.0, TorchMetrics $\geq$ 1.0, PyWavelets 1.8.0, PNPL (PyPI) [PNPL, 2025a] or our fork [Labs, 2025]. Python 3.10–3.12. Determinism via a global seed (777).

A.3 Datasets and on-disk layout. We consume pre-grouped LibriBrain HDF5 shards with $[0.0, 0.5]$ s windows at 250 Hz across 306 channels. Grouping directories like `./data/grouped_100/` contain `train_grouped.h5`, `validation_grouped.h5`, `test_grouped.h5`.

A.4 Preprocessing. Grouped averaging at $K \in \{5, 10, 25, 50, 100\}$; robust scaling inside the model; optional db4 denoising at inference.

A.5 Stage 1 training. Four Conformer layers with disentangled attention (32 buckets, max rel. distance 128), MLP aggregator, and three heads (39-way classifier, 14-feature IPA, projection head). Losses: focal ($\gamma = 2.0$, label smoothing 0.12), NT-Xent ($\tau = 0.03$, weight 0.5), IPA BCE (weight 0.2). AdamW, cosine schedule, warmup 5, clip 1.0, batch size 512.

A.6 Stage 2 logits. Emit raw logits per base and align loaders (no shuffle, no `drop_last`) to guarantee row-wise concatenation.

A.7 Stage 3 stacking. BatchNorm $\rightarrow$ Linear $39M \rightarrow 512 \rightarrow$ SiLU $\rightarrow$ BatchNorm $\rightarrow$ Dropout 0.5 $\rightarrow$ Linear $512 \rightarrow 39$. AdamW + ReduceLROnPlateau, early stopping; $N = 5$ seeds; average top-K runs ($K = 3$).

A.8 Inference and submission. Optional denoising flags (`-use-denoising -wavelet db4 -denoise-level 3`, etc.). Scripts validate output probabilities.

A.9 Logging and checkpoints. CSV and TensorBoard by default; filenames include epoch and score (e.g., `val_f1_macro`). Keep top-k and last checkpoints.