LENS: LINKING PHONEME, WORD, AND SENTENCE SUPERVISION FOR BRAIN-TO-TEXT DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Brain–computer interfaces offer a path to restoring communication by decoding intended speech from neural activity. High-performing brain-to-text systems combine connectionist temporal classification (CTC)-based phoneme decoding with language models, yet standard CTC neither explicitly encodes phoneme relations nor directly supervises word- and sentence-level neural representations. Pretrained linguistic representations could provide that direct supervision, but connecting words to neural features is challenging without annotated word timings. We propose **LENS**, a framework that **l**inks phon**e**me-, word-, and se**n**tence-level **s**upervision for brain-to-text decoding. LENS combines classifier geometry with neural representation covariance to assign bounded credit to phoneme alternatives during CTC supervision. Building on phoneme-level training, we introduce Posterior-Guided Multiscale Linguistic Alignment: model-derived soft alignment couples neural features with pretrained word embeddings without annotated word timings, complemented by utterance-level alignment with pretrained sentence representations. LENS achieves state-of-the-art word error rates on the Brain-to-Text '24 and '25 benchmarks without changing the established decoding pipeline. Together, these findings support combining phoneme supervision informed by neural evidence and phoneme relations with word- and sentence-level linguistic guidance for more accurate brain-to-text decoding.

1 INTRODUCTION

Speech brain-computer interfaces (BCIs) aim to restore communication for people with severe speech impairment by decoding intended speech from neural activity. By enabling users to express needs and participate in conversation, these systems could support greater independence and social connection (Willett et al., 2023; Card et al., 2024). Clinical demonstrations have progressed from restricted-vocabulary word decoding to large-vocabulary systems that combine phonetic predictions with lexicons and language models (Moses et al., 2021; Willett et al., 2023; Metzger et al., 2023; Card et al., 2024). To support systematic comparison of decoding methods, the Brain-to-Text Benchmark '24 provided a shared evaluation framework with a CTC-trained phoneme baseline and n-gram language-model decoding, highlighting substantial WER gains from ensembling and language-model refinement (Willett et al., 2024).

Recent work advances both neural decoding and language-model integration. DCoND models context-dependent diphones and combines phoneme decoding with n-gram and large language model (LLM) refinement (Li et al., 2025). Time-Masked Transformers combine extensive temporal masking, a compact encoder, and lightweight test-time adaptation for efficient decoding and robustness across held-out days (Feghhi et al., 2026b). BIT uses cross-species neural pretraining and evaluates both cascaded n-gram decoding and end-to-end audio-language-model integration through contrastive alignment (Zhang et al., 2026b). LightBeam reduces decoding memory requirements through non-WFST beam search with delayed LLM fusion (Feghhi et al., 2026a). These methods advance neural representations, adaptation, and decoding. A complementary direction is to enrich the supervision used to train the neural phoneme decoder while retaining its inference pipeline.

At the phoneme level, standard CTC marginalizes over unknown alignments but assigns no explicit similarity-dependent credit between distinct classes (Graves et al., 2006). Related approaches modify targets, transfer teacher knowledge, or constrain alignments: label smoothing regularizes cate-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

gorical targets (Müller et al., 2019), and CTC knowledge distillation transfers teacher information while addressing posterior-alignment difficulties (Tian et al., 2022). Other work incorporates multiple weighted transcription alternatives for semi-supervised handwriting recognition (Kišš et al., 2024). Closer to phoneme supervision, phoneme similarity has been incorporated into CTC and an auxiliary mapping objective (Zhou et al., 2025), while similarity-aware local alignments have been used to constrain CTC paths (Ye et al., 2025). These approaches motivate incorporating uncertainty and phoneme relations into sequence supervision. For neural speech decoding, a central design challenge is determining which phoneme alternatives are supported by the observed neural representations and how much credit to assign them without eroding distinctions needed for accurate transcription.

Beyond phoneme-level supervision, pretrained linguistic representations offer a way to directly guide neural representations at word and sentence scales. Cross-modal alignment with language is an effective source of supervision that has been explored at multiple scales, from global image–text matching (Radford et al., 2021) to patch–word correspondences (Yao et al., 2022). For speech and neural recording, applying such supervision at the word level requires identifying temporal correspondence between continuous signals and discrete linguistic units. Speech studies illustrate supervision at both scales: SAMU-XLSR aligns utterance representations with pretrained sentence embeddings (Khurana et al., 2022),while WACO aligns speech segments with transcript-word representations using an external forced aligner for low-resource speech translation (Ouyang et al., 2023). Model-derived temporal alignment can also support localized supervision: DyPCL updates CTC-based phoneme alignments during training for contrastive learning across typical and dysarthric speech samples (Lee et al., 2025).

For neural recordings, recent decoders link neural activity to linguistic units with different mechanisms and goals. Brain2Speech-Net uses a differentiable deep-HMM to align intracortical activity with contextual phoneme representations from a pretrained text-to-speech model, enabling speech synthesis without intermediate text decoding (Chandra et al., 2026). Brain2Qwerty v2 segments MEG representations recorded during typing into word-like units using CTC-predicted spaces, contrastively aligns them with LLM word embeddings, and conditions a fine-tuned LLM on the neural tokens and CTC-decoded text for sentence generation (Zhang et al., 2026a).These studies demonstrate the potential of linguistic representations to guide neural decoding across different tasks. A complementary direction is to jointly supervise word- and sentence-level representations within a neural-to-phoneme decoder. This requires connecting word-level targets to temporally uncertain neural features and combining them with utterance-level guidance while retaining the established phoneme-to-word decoding pipeline.

To address these challenges, we introduce **LENS**, a framework that **l**inks phon**e**me-, word-, and se**n**tence-level **s**upervision for brain-to-text decoding. LENS combines covariance-aware phoneme relaxation with posterior-guided linguistic alignment, enriching neural training without changing the phoneme-to-word inference pipeline. Our contributions are as follows:

- **Covariance-aware phoneme supervision (CAPS).** We relax exact-label CTC using phoneme affinities derived from a frozen teacher's classifier weights and hidden-state covariance. A column-stochastic kernel turns these affinities into bounded alternative credit while retaining an exact-phoneme anchor and a standard CTC alignment topology.

- **Posterior-guided multiscale linguistic alignment (PMLA).** We derive soft word-to-frame correspondence from reference-conditioned CTC posteriors of a pretrained phoneme decoder, without annotated word timings or an external aligner. This correspondence supports alignment of word-level neural representations with pretrained word embeddings, complemented by utterance-level alignment with pretrained sentence embeddings. Both objectives are used only during training, preserving the phoneme-to-word decoding pipeline at inference.

- **State-of-the-art brain-to-text decoding.** On the test sets, LENS achieves single-model WERs of 5.94% on Brain-to-Text '24 and 3.28%/3.11% on the '25 public/private splits. Its ensemble achieves 4.02%, 1.55%, and 1.43%, respectively, improving over the strongest prior ensemble results by 19.3%, 11.9%, and 19.2% in relative WER.

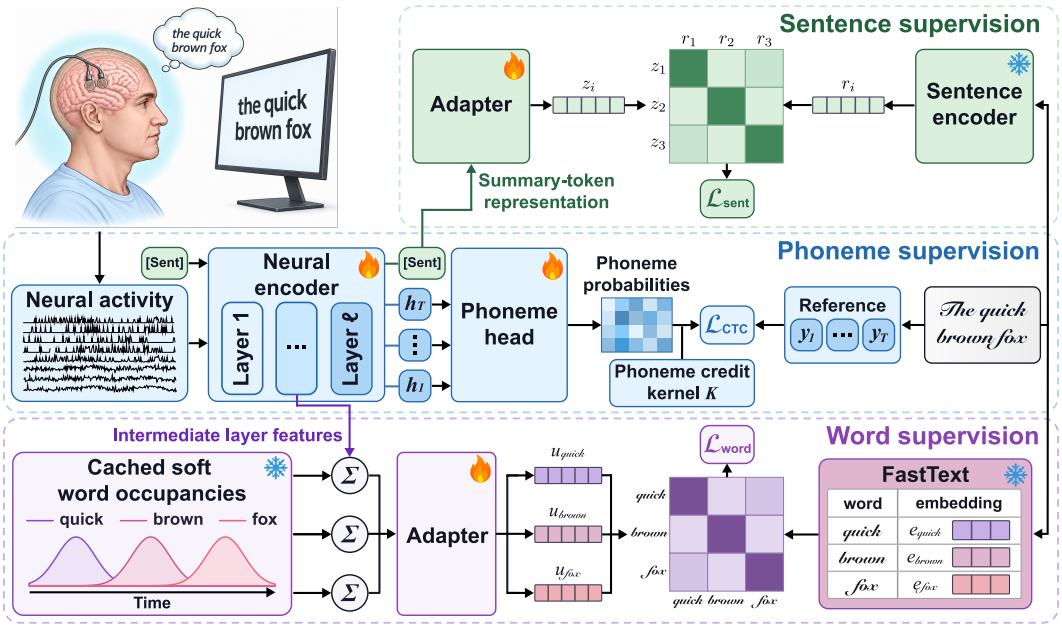

Figure 1: **LENS links phoneme, word, and sentence supervision through a shared neural encoder.** Covariance-aware phoneme training uses a credit kernel K to allocate alternative credit while retaining an exact-phoneme anchor. Subsequent linguistic fine-tuning uses cached, reference-conditioned CTC posteriors to pool word-level features, and a summary token to represent the utterance. Adapters align these representations with frozen FastText word and E5 sentence embeddings through contrastive objectives, alongside standard CTC.

2 Methodology

2.1 Problem Formulation

Let $\mathcal{D} = \{(x_i, y_i, s_i)\}_{i=1}^{N}$ contain neural recordings $x_i \in \mathbb{R}^{U_i \times F}$, with U_i time bins and equal numbers of threshold-crossing and spike-band-power features, together with reference phoneme sequences $y_i = (y_{i,1}, \ldots, y_{i,M_i}) \in \mathcal{P}^{M_i}$ with M_i phoneme labels, and transcripts $s_i = (w_{i,1}, \ldots, w_{i,K_i})$ with K_i words. The phoneme inventory $\mathcal{P}$ includes silence, which is distinct from the CTC blank $\varnothing$. Let θ denote the parameters of the neural encoder and phoneme heads. The encoder partitions the U_i input bins into temporal patches with stride r and produces layer-ℓ representations $h_{i,t}^{(\ell)}$, where $t \in \{1, \ldots, T_i\}$ indexes the $T_i = \lceil U_i / r \rceil$ encoder frames. The final phoneme head maps the neural representations to probabilities $p_{i,t}(c) \equiv p_\theta(c \mid x_i, t)$ over $\mathcal{C} = \mathcal{P} \cup \{\varnothing\}$, where θ is suppressed in framewise quantities unless needed. Standard CTC defines:

$$P_\theta(y_i \mid x_i) = \sum_{\pi \in \mathcal{B}^{-1}(y_i)} \prod_{t=1}^{T_i} p_{i,t}(\pi_t), \qquad \mathcal{L}_{\mathrm{CTC}}(x_i, y_i; \theta) = -\log P_\theta(y_i \mid x_i) \tag{1}$$

where $\mathcal{B}$ merges repeated labels and removes blanks (Graves et al., 2006). At inference, a fixed phoneme-to-word decoder produces $\widehat{s}_i = \mathrm{Dec}_{\mathrm{LM}}(p_{i,1:T_i})$, where $\mathrm{Dec}_{\mathrm{LM}}(\cdot)$ is a language model decoder.

2.2 LENS

Motivation & Overview Although brain-to-text decoding ultimately targets words and sentences, its standard phoneme-level CTC objective enforces exact label matching without explicitly incorporating phoneme relations or supervising word- and sentence-level neural representations. **LENS** enriches supervision at these levels, linking phoneme relations with multiscale linguistic supervision through two complementary mechanisms (Fig. 1). **Covariance-Aware Relaxation of Phoneme Supervision** uses classifier geometry under neural-representation covariance to assign bounded credit to related phonemes while preserving an exact-phoneme anchor (Section 2.3, Fig. 2). **Posterior-**

Guided Multiscale Linguistic Alignment supervises neural features with pretrained word and sentence representations, using model-derived temporal posteriors to associate individual words with neural frames without annotated timings (Section 2.4, Fig. 1).

Architecture & Pipeline LENS uses a TMT neural encoder (Feghhi et al., 2026b) and word- and sentence-level adapters targeting frozen FastText word embeddings (Bojanowski et al., 2017) and E5 sentence representations (Wang et al., 2022), respectively (Fig. 1). Training proceeds in three stages: **(1) Phoneme training:** train the neural model with covariance-aware phoneme supervision. **(2) Adapter initialization:** freeze the neural model and train the adapters and their temperatures. **(3) Joint fine-tuning:** unfreeze the neural model and optimize standard CTC alongside word and sentence objectives. At inference, final-head framewise phoneme probabilities feed WFST search combining CTC topology, a pronunciation lexicon, and an n-gram language model (Povey et al., 2011), followed by OPT-6.7B scoring (Zhang et al., 2022).

2.3 Covariance-Aware Relaxation of Phoneme Supervision

Standard CTC provides no explicit graded credit between distinct phoneme classes. We introduce **Covariance-Aware relaxation of Phoneme Supervision (CAPS)**, a fixed training-time emission mapping that allocates bounded alternative credit using a frozen teacher's classifier geometry and hidden-state covariance. The mapping retains standard CTC alignment topology and the inference pipeline 2.1.

Phoneme Relations from Neural Geometry To guide this relaxation, we derive phoneme relations from the teacher's classifier, whose weights describe which hidden directions support each phoneme. Weight similarity alone, however, ignores how these directions vary across neural recordings. We therefore measure the co-variation of class-relative teacher responses on training representations, using their regularized correlation as a prior for alternative credit.

Let $\mathbf{w}_c \in \mathbb{R}^d$ denote the frozen teacher's classifier vector for phoneme class $c \in \mathcal{C}$, and let $\{\mathbf{v}_n\}_{n=1}^{N_h}$ be its pre-classifier hidden representations sampled from training recordings, with mean $\bar{\mathbf{v}}$. Here, N_h is the number of sampled utterance–frame pairs. We center classifier vectors across classes, $\widetilde{\mathbf{w}}_c = \mathbf{w}_c - |\mathcal{C}|^{-1} \sum_{m \in \mathcal{C}} \mathbf{w}_m$, so that a shared logit component does not determine class similarity. We then define the regularized covariance and affinity $C_{cc'}$ between phoneme classes $c, c' \in \mathcal{C}$:

$$\widehat{\Sigma} = \frac{1}{N_h - 1} \sum_{n=1}^{N_h} (\mathbf{v}_n - \bar{\mathbf{v}})(\mathbf{v}_n - \bar{\mathbf{v}})^\top, \quad \Sigma_\lambda = \widehat{\Sigma} + \lambda_\Sigma \frac{\mathrm{tr}(\widehat{\Sigma})}{d} I_d, \quad C_{cc'} = \frac{\widetilde{\mathbf{w}}_c^\top \Sigma_\lambda \widetilde{\mathbf{w}}_{c'}}{\sqrt{(\widetilde{\mathbf{w}}_c^\top \Sigma_\lambda \widetilde{\mathbf{w}}_c)(\widetilde{\mathbf{w}}_{c'}^\top \Sigma_\lambda \widetilde{\mathbf{w}}_{c'})}} \tag{2}$$

Here, $\lambda_\Sigma \geq 0$ controls trace-scaled ridge regularization; the affinity is defined for classifier directions with positive variance under Σ_λ. Without ridge regularization, $C_{cc'}$ is the sample correlation between the class-relative responses $\widetilde{\mathbf{w}}_c^\top \mathbf{v}_n$ and $\widetilde{\mathbf{w}}_{c'}^\top \mathbf{v}_n$. More generally, $C_{cc'} = \cos(\Sigma_\lambda^{1/2} \widetilde{\mathbf{w}}_c, \Sigma_\lambda^{1/2} \widetilde{\mathbf{w}}_{c'})$; replacing Σ_λ with I_d recovers centered classifier cosine similarity. We estimate the covariance and affinities once from training recordings and keep them fixed throughout subsequent phoneme supervision.

Bounded Alternative Credit Correlated classifier responses do not imply interchangeable phonemes. We therefore separate alternative preference from total credit: temperature controls its distribution, while an exact-phoneme anchor limits departure from standard supervision. Let $\mathcal{O} = \mathcal{P} \setminus \{\mathrm{SIL}\}$ contain the ordinary phonemes. Rows index reference classes c and columns index emitted classes c'. For distinct $c, c' \in \mathcal{O}$, define:

$$G_{cc'} = \frac{\exp(C_{cc'}/\tau_{\mathrm{ph}})}{\sum_{m \in \mathcal{O} \setminus \{c'\}} \exp(C_{mc'}/\tau_{\mathrm{ph}})}, \qquad K^{(0)}_{cc'} = \rho G_{cc'}, \qquad K^{(0)}_{c'c'} = 1 - \rho \tag{3}$$

The temperature $\tau_{\mathrm{ph}} > 0$ controls alternative concentration, while $0 \leq \rho < 1$ fixes the total alternative credit per emitted phone. Blank and silence retain identity-only: $K_{\varnothing\varnothing} = K_{\mathrm{SIL,SIL}} = 1$, with all other entries in those rows and columns zero. Consequently, K is nonnegative and column stochastic. With G fixed, ρ is either constant or linearly increased from zero to its target value after an initial standard-CTC phase.

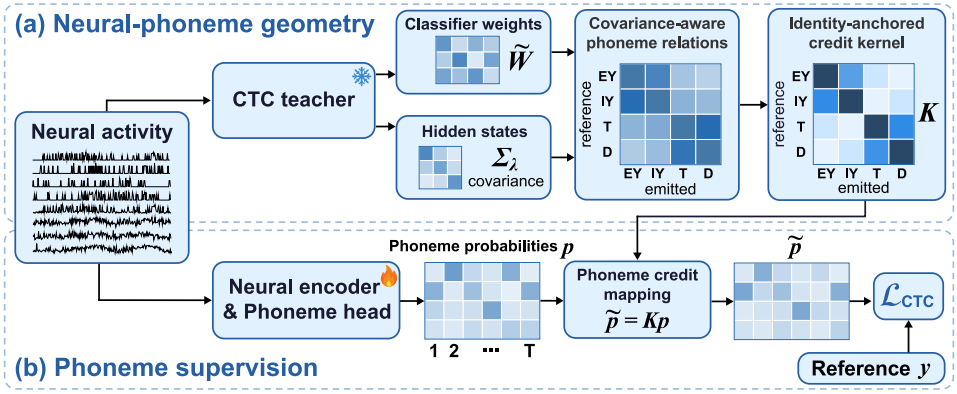

Figure 2: **Covariance-aware relaxation of phoneme supervision in LENS.** (a) Centered classifier weights and hidden-state covariance from a frozen CTC teacher define a regularized response affinity. This affinity guides a phoneme credit kernel K, which allocates bounded alternative credit while preserving an exact-phoneme anchor. (b) During training, K maps the neural model's framewise phoneme probabilities as $\widetilde{p} = Kp$ before CTC.

Sequence Training & Inference To incorporate phoneme relations without committing to a frame-level alignment, we map framewise emissions through K and retain CTC's marginalization over reference-compatible paths:

$$\widetilde{p}_{i,t}(c) = \sum_{c' \in \mathcal{C}} K_{cc'} p_{i,t}(c'), \qquad \mathcal{L}_{\text{CAPS}}(x_i, y_i; \theta) = -\log \sum_{\pi \in \mathcal{B}^{-1}(y_i) \cap \mathcal{C}^{T_i}} \prod_{t=1}^{T_i} \widetilde{p}_{i,t}(\pi_t) \tag{4}$$

Column stochasticity ensures $\sum_{c \in \mathcal{C}} \widetilde{p}_{i,t}(c) = 1$, so standard CTC forward–backward applies to the mapped emissions with the original target-state topology (Graves et al., 2006). When intermediate phoneme supervision is enabled, we apply the same credit kernel K to intermediate heads. At inference, the established decoder consumes the unmapped final-head probabilities $p_{i,t}$; neither the teacher nor the credit mapping is required.

2.4 Posterior-guided multiscale linguistic alignment

Linguistic fine-tuning aligns neural representations with pretrained word and sentence representations, using standard CTC Eq. (1) as the phoneme term. We omit the utterance index i for single-utterance quantities.

Soft Temporal Alignment We obtain word-to-frame correspondence from a frozen pretrained neural-to-phoneme decoder with parameters θ_0, referred to as the source model. Its emissions depend only on the neural input; the reference phoneme sequence constrains the subsequent alignment computation. For a reference sequence $y = (y_1, \ldots, y_M)$, we construct the blank-extended target $\widetilde{y} = (\varnothing, y_1, \varnothing, \ldots, y_M, \varnothing)$, with states indexed from 0 to $2M$. Using CTC forward-backward (Graves et al., 2006), we compute the posterior state occupancy $\gamma_t(r) = P_{\theta_0}(z_t = r \mid x, y)$, where z_t denotes the target state at frame t; see Appendix B for details. Silence labels separate the reference phoneme sequence into word-level groups. For word occurrence w_k spanning reference positions a_k to b_k, its framewise occupancy is $q_{k,t} = \sum_{r=2a_k-1}^{2b_k-1} \gamma_t(r)$. This includes the word's phoneme states and internal blanks, excluding boundary silence and surrounding blanks. These soft occupancies are precomputed and fixed during fine-tuning.

A target-conditioned oracle adapted from EM-Network (Yoon et al., 2023) combines neural features with reference phonemes on the training set. The same alignment procedure yields oracle occupancies $q^{\text{O}}_{k,t}$ (see Appendix C). Their agreement with source occupancies estimates alignment reliability:

$$\text{IoU}_k = \frac{\sum_t \min\left(q_{k,t}, q^{\text{O}}_{k,t}\right)}{\sum_t \max\left(q_{k,t}, q^{\text{O}}_{k,t}\right)} \tag{5}$$

We set $c_k = c_{\min} + (1 - c_{\min})\,\text{IoU}_k$, where $0 < c_{\min} < 1$. These fixed weights control the relative contributions of occurrences within each word type; $q_{k,t}$ alone determines temporal pooling.

Word-Level Linguistic Supervision We pool intermediate neural features using the source-derived occupancies:

$$\bar{h}_k = \frac{\sum_{t=1}^{T} q_{k,t} h_t^{(\ell_w)}}{\sum_{t=1}^{T} q_{k,t}}, \tag{6}$$

where ℓ_w is an intermediate encoder layer. A learned nonlinear projection adapter produces the unit-normalized embedding $u_k = g_w(\bar{h}_k)/\|g_w(\bar{h}_k)\|_2$. Words with negligible occupancy mass are excluded. Gradients pass through the neural features and adapter; alignment weights remain fixed.

Each unique word v has a frozen, unit-normalized FastText embedding e_v. For a candidate vocabulary $\mathcal{V}$ containing w_k, the occurrence loss is

$$\mathcal{L}_k^O(\mathcal{V}) = -\log \frac{\exp\big(u_k^\top e_{w_k}/\tau_w\big)}{\sum_{v\in\mathcal{V}} \exp\big(u_k^\top e_v/\tau_w\big)}, \tag{7}$$

where τ_w is a learned temperature. Candidates are unique word types, so repeated words share a positive prototype and do not become false negatives. We retain content words with usable pretrained embeddings and valid temporal support.

For minibatch aggregation, we restore the utterance index i, (i,k) denotes word occurrence k in utterance i. For an occurrence set $\mathcal{A}$, let $\mathcal{V}(\mathcal{A})$ contain its distinct word types and $\mathcal{A}_v$ its occurrences of type v. We balance word types and weight occurrences within each type by confidence:

$$\mathcal{J}(\mathcal{A}) = \frac{1}{|\mathcal{V}(\mathcal{A})|} \sum_{v\in\mathcal{V}(\mathcal{A})} \frac{\sum_{(i,k)\in\mathcal{A}_v} c_{i,k}\, \mathcal{L}_{i,k}^O(\mathcal{V}(\mathcal{A}))}{\sum_{(i,k)\in\mathcal{A}_v} c_{i,k}}. \tag{8}$$

The word objective combines minibatch-wide and within-utterance candidates:

$$\mathcal{L}_{\text{word}} = \eta\, \mathcal{J}(\mathcal{A}_{\text{batch}}) + (1-\eta)\frac{1}{|\mathcal{I}_{\text{word}}|} \sum_{i\in\mathcal{I}_{\text{word}}} \mathcal{J}(\mathcal{A}_i), \tag{9}$$

where $\mathcal{A}_i$ contains eligible occurrences in utterance i, $\mathcal{A}_{\text{batch}}$ contains those across the minibatch, $\mathcal{I}_{\text{word}}$ includes utterances with at least two eligible word types and $\eta \in [0,1]$ balances minibatch-wide and within-utterance supervision. We set $\mathcal{J}(\mathcal{A}) = 0$ when fewer than two types are available and the within-utterance term to zero when $\mathcal{I}_{\text{word}}$ is empty. Minibatch-wide candidates encourage discrimination across recordings; within-utterance candidates distinguish words sharing the same utterance context.

Sentence-Level Linguistic Supervision We append a learned summary token, denoted `[Sent]`, to the neural sequence and extract its final-layer representation h_i^{sent}. Let z_i and r_i denote the unit-normalized outputs of the sentence adapter $g_s(h_i^{\text{sent}})$ and frozen E5 encoder $E_s(s_i)$, respectively.

For a minibatch of size B, define $S_{i,j} = z_i^\top r_j/\tau_s$, where τ_s is a separately learned temperature. We use a symmetric contrastive loss:

$$\mathcal{L}_{\text{sent}} = -\frac{1}{2B}\sum_{i=1}^{B}\left[\log\frac{\exp(S_{i,i})}{\sum_{j\in\mathcal{N}_i}\exp(S_{i,j})} + \log\frac{\exp(S_{i,i})}{\sum_{j\in\mathcal{N}_i}\exp(S_{j,i})}\right], \tag{10}$$

where $\mathcal{N}_i = \{i\} \cup \{j \in \{1,\dots,B\} : s_j \neq s_i\}$. This excludes off-diagonal pairs with identical transcripts from the negative set. Sentence supervision complements word-level supervision by aligning a representation of the complete neural sequence.

Optimization and Inference Training begins with projection-head initialization. We freeze the pretrained neural encoder, phoneme heads, and summary-token embedding, optimizing only the two projection heads and their temperatures. We then unfreeze the neural model and jointly optimize

$$\mathcal{L} = \mathcal{L}_{\text{CTC}}^{\text{final}} + \frac{\lambda_{\text{int}}}{|\mathcal{H}|}\sum_{\ell\in\mathcal{H}}\mathcal{L}_{\text{CTC}}^{(\ell)} + \lambda_w^{(n)}\mathcal{L}_{\text{word}} + \lambda_s^{(n)}\mathcal{L}_{\text{sent}}, \tag{11}$$

where $\mathcal{H}$ indexes intermediate phoneme heads and n is the joint fine-tuning epoch. Each CTC term averages target-length-normalized per-example losses. The auxiliary weights decay to zero, leaving only CTC supervision in the final training phase. Neural augmentations preserve the temporal coordinates of the cached alignments. At inference, the summary token remains in the encoder, while the linguistic adapters are discarded.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Datasets We evaluate LENS on the Brain-to-Text '24 (Willett et al., 2024) and '25 (Card et al., 2025) benchmarks, each comprising intracortical recordings of attempted speech from one participant with amyotrophic lateral sclerosis. The datasets contain 12,100 and 10,948 sentence trials recorded with 128 and 256 electrodes, respectively, and provide threshold-crossing and spike-band-power features in 20-ms bins. We follow the official preprocessing and data splits, and train separate models for each benchmark. Ablations use the validation sets; competition holdouts are reserved for final evaluation.

Baselines Our primary baseline is the Time-Masked Transformer (TMT) (Feghhi et al., 2026b) trained with standard CTC, using the same backbone as LENS to assess the benefit of our supervision. We additionally compare with published results from DCoND (Li et al., 2025), which models context-dependent diphones; BIT (Zhang et al., 2026b), which uses cross-species neural pretraining; and LightBeam (Feghhi et al., 2026a), which integrates an LLM into non-WFST beam search.

Evaluation Metrics We evaluate transcription using word error rate (WER), defined as the total number of word substitutions, deletions, and insertions divided by the total number of reference words. For our single-model and ablation results, we report the mean $\pm$ standard error of the mean across ten random seeds for Brain-to-Text '24 and five for '25, using matched seeds for controlled comparisons. Baseline uncertainties are shown as originally reported, with ensemble results reported as point estimates. In the ablation studies, CAPS variants are compared against standard CTC, while PMLA variants are compared against the CAPS$\rightarrow$CTC fine-tuning control. Unless otherwise specified, hypothesis tests use one-tailed paired t-tests on matched experimental units at $\alpha = 0.05$.

3.2 RESULTS

Table 1 compares WER on the Brain-to-Text '24 competition set and the '25 public and private competition sets, separately for single-model and ensemble configurations.

Single-Model Performance Before linguistic fine-tuning, CAPS achieves 6.10% WER on '24 and 3.43%/3.35% on the '25 public/private sets, below the strongest listed non-LENS single-model result on each split. Posterior-guided multiscale linguistic fine-tuning further lowers WER to 5.94%, 3.28%, and 3.11%, respectively—absolute reductions of 0.16, 0.15, and 0.24 percentage points from CAPS. Full LENS thus achieves relative WER reductions of 6.5%, 19.2%, and 32.2% against the strongest listed non-LENS single-model comparators: BIT on '24 and '25 public, and LightBeam on '25 private. These results show lower mean WER after linguistic fine-tuning on all three splits while retaining the single-model decoding pipeline.

Table 1: Word error rate (%, $\downarrow$) on the Brain-to-Text '24 competition set and the Brain-to-Text '25 public and private competition sets. Results are grouped by single-model and ensemble configurations. **BOLD**/UNDERLINE indicates the best/second-best results. – indicates unavailable results.

Type	Model	B2T '24	B2T '25	
		Comp.	Public	Private
Single Model	DCoND	$8.39_{\pm 0.10}$	–	–
	TMT	$8.18_{\pm 0.22}$	–	–
	BIT	6.35	4.06	8.30
	LightBeam	$8.08_{\pm 0.12}$	$4.44_{\pm 0.21}$	$4.59_{\pm 0.14}$
	CAPS	$\underline{6.10}_{\pm 0.05}$	$\underline{3.43}_{\pm 0.14}$	$\underline{3.35}_{\pm 0.09}$
	LENS	$\mathbf{5.94}_{\pm 0.06}$	$\mathbf{3.28}_{\pm 0.12}$	$\mathbf{3.11}_{\pm 0.05}$
Ensemble	DCoND	5.77	–	–
	TMT	5.68	–	–
	BIT	5.10	<u>1.76</u>	<u>1.77</u>
	LightBeam	<u>4.98</u>	–	–
	LENS	**4.02**	**1.55**	**1.43**

Ensemble Decoding The LENS ensemble achieves 4.02% WER on '24 and 1.55%/1.43% on the '25 public/private sets. Relative to the strongest listed ensemble comparators—LightBeam on '24 (4.98%) and BIT on '25 (1.76%/1.77%)—these correspond to reductions of 19.3%, 11.9%, and 19.2%, respectively. As detailed in Appendix D, this extension combines checkpoints before and after linguistic fine-tuning using learned language-model voting. These are system-level improvements that include hypothesis aggregation and learned refinement.

3.3 Ablation Studies

Table 2 reports validation WER for the ablations of phoneme supervision, extended CTC-only fine-tuning, and the components of linguistic fine-tuning. UNIFORM assigns equal off-diagonal phoneme credit; CLF uses centered classifier-weight cosine affinities; and CLF+COV (CAPS) incorporates hidden-state covariance. The $-\mathcal{L}_{\text{word}}$ variant sets $\lambda_w = 0$ throughout fine-tuning, retaining only sentence-level supervision alongside CTC. Conversely, $-\mathcal{L}_{\text{sent}}$ sets $\lambda_s = 0$, retaining only word-level supervision alongside CTC. The $-$ORACLE variant retains both objectives but sets $c_k = 1$ for every retained word occurrence, removing source–oracle agreement weighting; occurrences within each word type are therefore weighted equally, while word-type balancing and source-derived soft temporal pooling are preserved.

Table 2: Ablation study on the Brain-to-Text '24 and '25 validation sets. We report word error rate (%, ↓). **Bold** indicates the best result. * denotes it is significantly better than its corresponding control.

Type	Model	B2T '24	B2T '25
	CTC	$9.90_{\pm 0.19}$	$3.16_{\pm 0.09}$
CAPS	UNIFORM	$9.66_{\pm 0.08}$	$3.28_{\pm 0.13}$
	CLF	$9.47^{*}_{\pm 0.06}$	$3.21_{\pm 0.13}$
	CLF+COV	$9.34^{*}_{\pm 0.07}$	$3.12_{\pm 0.14}$
FT	CTC→CTC	$9.37_{\pm 0.11}$	$3.06_{\pm 0.12}$
	CAPS→CTC	$9.06_{\pm 0.07}$	$2.95_{\pm 0.11}$
PMLA	$-\mathcal{L}_{\text{word}}$	$9.03_{\pm 0.06}$	$2.92_{\pm 0.10}$
	$-\mathcal{L}_{\text{sent}}$	$9.04_{\pm 0.06}$	$2.96_{\pm 0.10}$
	$-$ORACLE	$9.01_{\pm 0.06}$	$2.90_{\pm 0.12}$
	LENS	$\mathbf{8.92}^{*}_{\pm 0.05}$	$\mathbf{2.85}^{*}_{\pm 0.12}$

Covariance-Aware Phoneme Supervision CAPS (CLF+COV) reduces WER from 9.90%/3.16% to 9.34%/3.12% on '24 and '25. The alternative-credit ablations examine whether these gains reflect relaxation alone or the relations used to distribute credit. UNIFORM yields 9.66%/3.28% WER, whereas CLF yields 9.47%/3.21%, suggesting that classifier-derived preferences provide more useful supervision than uniform allocation. Incorporating covariance further gives CAPS the lowest WER among the phoneme-supervision variants on both benchmarks. Unlike classifier-weight similarity alone, the covariance-aware affinity makes alternative preferences dependent on the observed neural representations. On '25, UNIFORM and CLF worsen WER relative to CTC, whereas CAPS modestly improves it. Thus, the findings favor structured, representation-informed credit over indiscriminate relaxation.

Extended CTC-Only Fine-Tuning To examine whether the phoneme-training advantage persists after further optimization, we compare two 1,200-epoch schedules on '24: 600 epochs of standard or covariance-aware CTC, each followed by 600 epochs of standard CTC without linguistic losses. Standard-CTC initialization reaches 9.37% WER, whereas covariance-aware initialization reaches 9.06%, a difference of 0.31% (3.3% relative). Relative to their respective initial-stage results, these schedules improve WER by 0.53% and 0.30%. Thus, extended training accounts for substantial gains, while the covariance-aware training schedule retains a lower mean WER under the common standard-CTC continuation.

Linguistic Objectives & Agreement Weighting Relative to the CAPS→CTC control, full LENS reduces WER from 9.06% to 8.92% on '24 and from 2.95% to 2.85% on '25, achieving significant improvements on both benchmarks. Word-only and sentence-only supervision produce smaller and less consistent changes, while their combination yields the lowest mean WER on both datasets. This pattern supports complementary benefits from localized lexical targets and utterance-level linguistic guidance. Oracle-derived agreement weighting further lowers mean WER by 0.09% on '24 and 0.05% on '25. Since source-derived temporal pooling remains unchanged, these numerical gains suggest that weighting word occurrences by alignment reliability strengthens the benefit of multi-scale supervision.

4 Discussion

Phoneme-context organization Fig. 3 groups phonemes into vowels and consonants with (prevocalic) or without (coda) a later vowel in the reference word. Layer-wise projections show organization under CAPS, particularly at layer 6, that remains visually apparent after PMLA. At layer 6, CAPS significantly reduces normalized within-group distances for all three groups and increases vowel–consonant distances relative to CTC. The remaining prevocalic–coda distance decreases slightly. This context-associated structure emerges without explicit group-label or prevocalic-versus-coda supervision. These findings complement the WER ablations by revealing structured phoneme representations under CAPS.

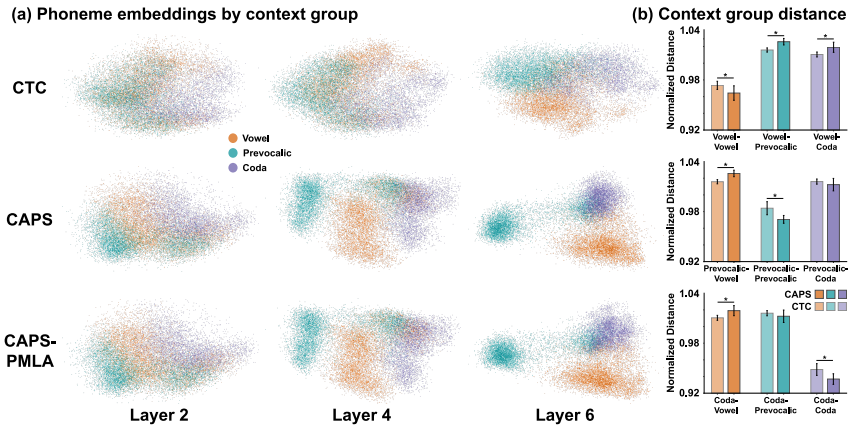

Figure 3: **Phoneme-context organization under CAPS and PMLA** (a) layer-wise PCA projections and (b) normalized layer-6 group distances ($*$ denotes $p < 0.05$); details in Appendix E.

Linguistic supervision shapes sentence-token attention. Fig. 4 compares models that both retain the appended `[Sent]` token but differ in whether linguistic losses are applied. Blue summarizes normalized `[Sent]`-to-frame attention, while pink shows CTC-derived word-occupancy posteriors. Without linguistic supervision, attention is strongly concentrated near the sequence ending. With supervision, this concentration decreases and more pronounced peaks appear within several word-associated intervals, although a terminal peak remains. This redistribution matters because `[Sent]` supplies the utterance representation: the observed pattern is consistent with gathering temporally localized word-related information to represent the complete utterance. Since both models contain the same token, the comparison suggests that linguistic supervision shapes how this aggregation occurs. Its presence in both training and validation examples further suggests that this behavior can extend to unseen utterances

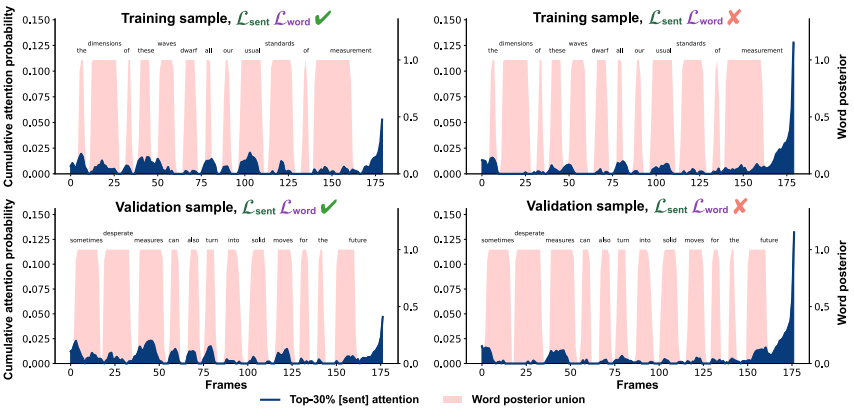

Figure 4: **Sentence-token attention with and without linguistic supervision** Blue shows normalized `[Sent]` attention and pink word-occupancy posteriors, with linguistic supervision enabled (left) or disabled (right); visualization details in Appendix F.

5 Conclusion

We introduced LENS, combining covariance-aware phoneme relaxation with posterior-guided word-level alignment and sentence-level supervision, without annotated word timings. LENS improves single-model WER on the Brain-to-Text '24 and '25 benchmarks while retaining the established phoneme-to-word decoding pipeline. Ablations favor representation-informed alternative credit over uniform allocation and show additional gains from combined linguistic supervision beyond continued CTC training. Together, these findings support enriching neural supervision across linguistic scales as a route to more accurate brain-to-text decoding. Evaluation across more participants and in online settings remains necessary to establish broader applicability.

AI USE STATEMENT

We used generative AI tools to assist with drafting and polishing the manuscript, identifying and summarizing related literature, research ideation, methodological and experimental design, implementation and debugging, interpretation of results, and hyperparameter tuning, including suggesting parameter settings and follow-up experiments. AI assistance also supported the presentation of results and preparation of scientific figures.

The authors reviewed AI-assisted text, mathematical formulations, code, and analyses, and checked the cited literature and reported results against the corresponding sources and experimental outputs. All methodological decisions and conclusions remained the responsibility of the authors. We take full responsibility for the final content of this work, including text, claims, code, and other artifacts produced with generative AI assistance.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide the model architecture, training procedures, hyperparameters, optimization schedules, checkpoint-selection criterion, and computing resources in Appendix G. Section 3.1 describes the datasets, data splits, and evaluation protocol. Appendices B and C detail the soft temporal alignment computation and oracle construction, respectively, while Appendix D specifies the ensemble decoding procedure and language-model fine-tuning settings.

REFERENCES

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Nicholas Card, Maitreyee Wairagkar, Carrina Iacobacci, Xianda Hou, Tyler Singer-Clark, Francis R. Willett, Erin M. Kunz, Chaofei Fan, Maryam Vahdati Nia, Darrel R. Deo, Aparna Srinivasan, Eun Young Choi, Matthew F. Glasser, Leigh R. Hochberg, Jaimie M. Henderson, Kiarash Shahlaie, Sergey D. Stavisky, and David M. Brandman. Brain-to-Text '25. `https://www.kaggle.com/competitions/brain-to-text-25`, 2025. Kaggle.

Nicholas S Card, Maitreyee Wairagkar, Carrina Iacobacci, Xianda Hou, Tyler Singer-Clark, Francis R Willett, Erin M Kunz, Chaofei Fan, Maryam Vahdati Nia, Darrel R Deo, et al. An accurate and rapidly calibrating speech neuroprosthesis. *New England Journal of Medicine*, 391(7):609–618, 2024.

Shreeram Suresh Chandra, Zexin Cai, Yu Tsao, Simon King, and Berrak Sisman. Brain2SpeechNet: Intelligible, real-time brain-to-speech synthesis without text decoding. *arXiv preprint arXiv:2609.04455*, 2026.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36:10088–10115, 2023.

Ebrahim Feghhi, Junlin Hu, Nima Hadidi, and Jonathan C Kao. Lightbeam: An accurate and memory-efficient ctc decoder for speech neuroprostheses. *arXiv preprint arXiv:2603.14002*, 2026a.

Ebrahim Feghhi, Shreyas Kaasyap, Nima Hadidi, and Jonathan Kao. Time-masked transformers with lightweight test-time adaptation for neural speech decoding. *Advances in Neural Information Processing Systems*, 38:93047–93072, 2026b.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376, 2006.

Sameer Khurana, Antoine Laurent, and James Glass. SAMU-XLSR: Semantically-aligned multimodal utterance-level cross-lingual speech representation. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1493–1504, 2022.

Martin Kišš, Michal Hradiš, Karel Beneš, Petr Buchal, and Michal Kula. Softctc—semi-supervised learning for text recognition using soft pseudo-labels: M. ki et al. *International Journal on Document Analysis and Recognition (IJDAR)*, 27(2):177–193, 2024.

Wonjun Lee, Solee Im, Heejin Do, Yunsu Kim, Jungseul Ok, and Gary Lee. DYPCL: Dynamic phoneme-level contrastive learning for dysarthric speech recognition. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4701–4712, 2025.

Jingyuan Li, Trung Le, Chaofei Fan, Mingfei Chen, and Eli Shlizerman. Brain-to-text decoding with context-aware neural representations and large language models. *Journal of Neural Engineering*, 22(5):056026, 2025.

Sean L Metzger, Kaylo T Littlejohn, Alexander B Silva, David A Moses, Margaret P Seaton, Ran Wang, Maximilian E Dougherty, Jessie R Liu, Peter Wu, Michael A Berger, et al. A high-performance neuroprosthesis for speech decoding and avatar control. *Nature*, 620(7976):1037–1046, 2023.

David A Moses, Sean L Metzger, Jessie R Liu, Gopala K Anumanchipalli, Joseph G Makin, Pengfei F Sun, Josh Chartier, Maximilian E Dougherty, Patricia M Liu, Gary M Abrams, et al. Neuroprosthesis for decoding speech in a paralyzed person with anarthria. *New England Journal of Medicine*, 385(3):217–227, 2021.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Siqi Ouyang, Rong Ye, and Lei Li. WACO: Word-aligned contrastive learning for speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3891–3907, 2023.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Sanli Tian, Keqi Deng, Zehan Li, Lingxuan Ye, Gaofeng Cheng, Ta Li, and Yonghong Yan. Knowledge distillation for ctc-based speech recognition via consistent acoustic representation learning. In *Interspeech*, pp. 2633–2637, 2022.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Francis R Willett, Erin M Kunz, Chaofei Fan, Donald T Avansino, Guy H Wilson, Eun Young Choi, Foram Kamdar, Matthew F Glasser, Leigh R Hochberg, Shaul Druckmann, et al. A high-performance speech neuroprosthesis. *Nature*, 620(7976):1031–1036, 2023.

Francis R Willett, Jingyuan Li, Trung Le, Chaofei Fan, Mingfei Chen, Eli Shlizerman, Yue Chen, Xin Zheng, Tatsuo S Okubo, Tyler Benster, et al. Brain-to-text benchmark'24: Lessons learned. *arXiv preprint arXiv:2412.17227*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. In *International Conference on Learning Representations*, 2022.

Zongli Ye, Jiachen Lian, Akshaj Gupta, Xuanru Zhou, Haodong Li, Krish Patel, Hwi Joo Park, Dingkun Zhou, Chenxu Guo, Shuhe Li, et al. Lcs-ctc: Leveraging soft alignments to enhance phonetic transcription robustness. In *2025 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. IEEE, 2025.

Ji Won Yoon, Sunghwan Ahn, Hyeonseung Lee, Minchan Kim, Seok Min Kim, and Nam Soo Kim. EM-network: Oracle-guided self-distillation for sequence learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40111–40128. PMLR, 2023.

Mingfang Zhang, Jarod Lévy, Cedric Rommel, Jérémy Rapin, Corentin Bel, Julie Bonnaire, Daniel Nieto, Pierre Bourdillon, Svetlana Pinet, Stéphane d'Ascoli, et al. Accurate decoding of natural sentences from non-invasive brain recordings. *arXiv preprint arXiv:2608.18114*, 2026a.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Yizi Zhang, Linyang He, Chaofei Fan, Tingkai Liu, Han Yu, Trung Le, Jingyuan Li, Scott Linderman, Lea Duncker, Francis Willett, et al. A cross-species neural foundation model for end-to-end speech decoding. In *International Conference on Learning Representations*, volume 2026, pp. 30773–30797, 2026b.

Xuanru Zhou, Jiachen Lian, Cheol Jun Cho, Tejas Prabhune, Shuhe Li, William Li, Rodrigo Ortiz, Zoe Ezzes, Jet Vonk, Brittany Morin, et al. Towards accurate phonetic error detection through phoneme similarity modeling. *arXiv preprint arXiv:2507.14346*, 2025.

A Interpretation of Covariance-Aware Phoneme Affinity

We show that the affinity in Eq. (2) is an empirical correlation of class-relative teacher responses without ridge regularization, and a covariance-weighted cosine similarity with or without regularization.

Class-Relative Teacher Responses Let $\{\mathbf{v}_n\}_{n=1}^{N_h}$, with $N_h > 1$, be the sampled pre-classifier hidden representations from the frozen teacher's training recordings, with mean $\bar{\mathbf{v}}$ and sample covariance $\widehat{\Sigma}$. For $c \in \mathcal{C}$, write its affine logit as $z_c(\mathbf{v}) = \mathbf{w}_c^\top \mathbf{v} + b_c$. Define $\bar{\mathbf{w}} = |\mathcal{C}|^{-1} \sum_m \mathbf{w}_m$ and $\bar{b} = |\mathcal{C}|^{-1} \sum_m b_m$, where the sums run over $\mathcal{C}$. Subtracting the across-class mean logit gives

$$a_c(\mathbf{v}) = z_c(\mathbf{v}) - \frac{1}{|\mathcal{C}|} \sum_{m \in \mathcal{C}} z_m(\mathbf{v}) = \widetilde{\mathbf{w}}_c^\top \mathbf{v} + (b_c - \bar{b}), \qquad \widetilde{\mathbf{w}}_c = \mathbf{w}_c - \bar{\mathbf{w}}. \tag{12}$$

This removes the same scalar from every logit at a given frame and therefore leaves softmax probabilities unchanged. Writing $\bar{a}_c = N_h^{-1} \sum_n a_c(\mathbf{v}_n)$, centering across sampled frames yields $a_c(\mathbf{v}_n) - \bar{a}_c = \widetilde{\mathbf{w}}_c^\top (\mathbf{v}_n - \bar{\mathbf{v}})$. Thus, class centering removes the shared logit component, whereas sample centering removes each response's mean; constant classifier biases cancel in the latter step.

Empirical Response Correlation Using sample covariance with denominator $N_h - 1$, we obtain

$$\begin{aligned} \widehat{\mathrm{Cov}}_n\big(a_c(\mathbf{v}_n), a_{c'}(\mathbf{v}_n)\big) &= \frac{1}{N_h - 1} \sum_{n=1}^{N_h} \big(a_c(\mathbf{v}_n) - \bar{a}_c\big)\big(a_{c'}(\mathbf{v}_n) - \bar{a}_{c'}\big) \\ &= \widetilde{\mathbf{w}}_c^\top \left[\frac{1}{N_h - 1} \sum_{n=1}^{N_h} (\mathbf{v}_n - \bar{\mathbf{v}})(\mathbf{v}_n - \bar{\mathbf{v}})^\top \right] \widetilde{\mathbf{w}}_{c'} \\ &= \widetilde{\mathbf{w}}_c^\top \widehat{\Sigma} \widetilde{\mathbf{w}}_{c'}. \end{aligned} \tag{13}$$

Consequently, when both response variances are positive,

$$\widehat{\mathrm{Corr}}_n\big(a_c(\mathbf{v}_n), a_{c'}(\mathbf{v}_n)\big) = \frac{\widetilde{\mathbf{w}}_c^\top \widehat{\Sigma} \widetilde{\mathbf{w}}_{c'}}{\sqrt{(\widetilde{\mathbf{w}}_c^\top \widehat{\Sigma} \widetilde{\mathbf{w}}_c)(\widetilde{\mathbf{w}}_{c'}^\top \widehat{\Sigma} \widetilde{\mathbf{w}}_{c'})}} = C_{cc'}\big|_{\lambda_\Sigma = 0}. \tag{14}$$

Because constant offsets do not affect correlation, the same identity holds for the responses $\widetilde{\mathbf{w}}_c^\top \mathbf{v}_n$ and $\widetilde{\mathbf{w}}_{c'}^\top \mathbf{v}_n$ used in the main text. This is an algebraic identity for the sampled frames; it does not require those frames to be independent. It describes class-relative logit responses, not softmax probabilities.

Regularization & Geometric Interpretation. Let $d = \dim(\mathbf{v}_n)$, $\lambda_\Sigma \geq 0$, $\delta = \lambda_\Sigma \operatorname{tr}(\widehat{\Sigma})/d$ and $\Sigma_\lambda = \widehat{\Sigma} + \delta I_d$. The ridge term adds $\delta \widetilde{\mathbf{w}}_c^\top \widetilde{\mathbf{w}}_{c'}$ to the response covariance and $\delta \|\widetilde{\mathbf{w}}_c\|_2^2$ to the variance of class c. Accordingly, $C_{cc'}$ is a regularized response correlation, not generally the empirical correlation of the original responses when $\delta > 0$. For the positive-semidefinite matrix Σ_λ, let $\mathbf{q}_c = \Sigma_\lambda^{1/2} \widetilde{\mathbf{w}}_c$, using its symmetric square root. Whenever the projected variances are positive,

$$C_{cc'} = \frac{\mathbf{q}_c^\top \mathbf{q}_{c'}}{\|\mathbf{q}_c\|_2 \|\mathbf{q}_{c'}\|_2} = \cos\Big(\Sigma_\lambda^{1/2} \widetilde{\mathbf{w}}_c, \Sigma_\lambda^{1/2} \widetilde{\mathbf{w}}_{c'}\Big). \tag{15}$$

Replacing Σ_λ by I_d recovers the centered classifier-only cosine affinity. Thus, covariance introduces the observed representation variability into the weight geometry; it does not apply inverse-covariance weighting. Positive ridge ensures positive projected variance for nonzero centered classifier vectors when $\operatorname{tr}(\widehat{\Sigma}) > 0$; degenerate cases still require explicit checks.

Implication for Phoneme Supervision The affinity measures whether class-relative teacher responses co-vary across training representations, providing a data-dependent prior for alternative credit. It is not a confusion probability or evidence that phonemes are interchangeable: representation variability can also reflect nuisance factors. We therefore combine this prior with an exact-phoneme anchor and a bounded alternative budget, and assess its utility through controlled downstream evaluations.

B CTC FORWARD–BACKWARD FOR SOFT TEMPORAL ALIGNMENT

We describe the standard CTC forward–backward algorithm (Graves et al., 2006) using the notation of Section 2.4. The frozen source model provides framewise probabilities $p_t(c) = P_{\theta_0}(\pi_t = c \mid x)$, where π_t is the emitted symbol at frame t. The reference sequence $y = (y_1, \dots, y_M)$ constrains the valid alignments but does not enter the source model.

Alignment States and Transitions We insert CTC blanks to form $\widetilde{y} = (\varnothing, y_1, \varnothing, \dots, y_M, \varnothing)$, with states indexed by $r \in \{0, \dots, 2M\}$. Each reference phoneme position has its own state, so repeated phonemes at different positions remain distinct. A valid state path starts at state 0 or 1 and ends at state $2M-1$ or $2M$. At each subsequent frame, it may remain in the same state, advance by one state, or advance by two states if the destination is a nonblank symbol different from the current symbol. The last condition prevents consecutive identical reference phonemes from merging under CTC collapse.

Let $\mathcal{S}(r)$ denote the allowed successor states: it contains r, $r+1$ when $r+1 \le 2M$, and $r+2$ when $r+2 \le 2M$, $\widetilde{y}_{r+2} \neq \varnothing$, and $\widetilde{y}_{r+2} \neq \widetilde{y}_r$. The corresponding predecessor set is $\mathcal{P}(r) = \{u : r \in \mathcal{S}(u)\}$.

Forward Recursion The forward variable $\alpha_t(r)$ sums the probabilities of all valid partial paths reaching state r at frame t, including the emission at that frame. At $t = 1$, only the initial blank and first phoneme can be occupied:

$$\alpha_1(r) = \begin{cases} p_1(\widetilde{y}_r), & r \in \{0, 1\}, \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

For $t = 2, \dots, T$, we sum over the allowed predecessors:

$$\alpha_t(r) = p_t(\widetilde{y}_r) \sum_{u \in \mathcal{P}(r)} \alpha_{t-1}(u). \tag{17}$$

A complete alignment ends at the final phoneme or final blank, giving

$$P_{\theta_0}(y \mid x) = \alpha_T(2M-1) + \alpha_T(2M). \tag{18}$$

Backward Recursion The backward variable $\beta_t(r)$ sums the probabilities of all valid continuations from state r at frame t to the end of the reference sequence. It includes emissions at frames $t+1, \dots, T$, excluding the emission at t. Accordingly, its terminal values are

$$\beta_T(r) = \begin{cases} 1, & r \in \{2M-1, 2M\}, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

For $t = T-1, \dots, 1$, the recursion is

$$\beta_t(r) = \sum_{u \in \mathcal{S}(r)} p_{t+1}(\widetilde{y}_u) \beta_{t+1}(u). \tag{20}$$

Posterior State Occupancy Every complete alignment passing through state r at frame t decomposes uniquely into a prefix ending at that state and a continuation after that frame. Let $\mathcal{Z}(y)$ denote the set of valid complete state paths. Under the CTC factorization,

$$\alpha_t(r)\beta_t(r) = \sum_{\substack{z \in \mathcal{Z}(y) \\ z_t = r}} \prod_{\tau=1}^{T} p_\tau(\widetilde{y}_{z_\tau}). \tag{21}$$

Dividing by the total probability of the reference sequence gives

$$\gamma_t(r) = P_{\theta_0}(z_t = r \mid x, y) = \frac{\alpha_t(r)\beta_t(r)}{P_{\theta_0}(y \mid x)}. \tag{22}$$

Thus, $\sum_{r=0}^{2M} \gamma_t(r) = 1$ at each frame, provided the reference has nonzero probability.

C Oracle Architecture and Alignment Extraction

The target-conditioned oracle is initialized from the pretrained neural-to-phoneme decoder and augmented with a reference-phoneme encoder and an attention-based fusion module, adapting the architecture of EM-Network (Yoon et al., 2023).We again omit the utterance index i for clarity. The oracle's neural encoder maps the neural recording x to a sequence of neural features H. In parallel, the reference phoneme sequence y is represented using learned token and positional embeddings, followed by a Transformer encoder and layer normalization to produce reference-phoneme representations R. The fusion block first applies self-attention to H, allowing neural frames to exchange temporal information. It then performs cross-attention, using the updated neural features as queries and R as keys and values. This allows each neural frame to incorporate information from the reference phoneme sequence while preserving the neural sequence's temporal resolution. A feed-forward network further transforms the fused features. The attention and feed-forward sublayers use residual connections and layer normalization. The fused representations pass through the phoneme output head, yielding frame-level probabilities over $\mathcal{C} = \mathcal{P} \cup \{\varnothing\}$.

For offline alignment extraction, we freeze the oracle and apply CTC forward–backward to its emissions, restricting paths to those that collapse to the reference phoneme sequence. We aggregate the target-state posteriors $\gamma_t^{\mathrm{O}}(r)$ into word occupancies:

$$q_{k,t}^{\mathrm{O}} = \sum_{r=2a_k-1}^{2b_k-1} \gamma_t^{\mathrm{O}}(r), \tag{23}$$

where a_k and b_k are the first and last reference-phoneme positions of word occurrence k, as defined in Section 2.4. As for the source model, this sum includes the word's phoneme states and internal blank states, excluding surrounding blanks and inter-word silence. The oracle occupancies are used only to compute agreement weights; the source-model occupancies determine temporal pooling.

D Deep-Ensemble Decoding with Learned Language-Model Voting

The single-model setting in the main text decodes each neural model's framewise phoneme probabilities using WFST search followed by OPT-6.7B rescoring, yielding one sentence prediction. Here, we describe a deep-ensemble extension that combines hypotheses from multiple LENS checkpoints through a separately fine-tuned Qwen model (Yang et al., 2025). Following Li et al. (2025), we provide decoded phoneme sequences alongside sentence hypotheses as inputs to the language model. Qwen serves as a learned voting model: it conditions on these hypotheses and their scores to reconstruct a final transcript and phoneme sequence. Its output can reproduce a candidate or correct the candidates using their combined evidence. Following Section 2.1, x_i denotes a neural recording, y_i its reference phoneme sequence, and s_i its transcript. The aggregation model is trained after the neural checkpoints have been fixed and supplies no gradients to the neural encoder or the linguistic projection heads of Section 2.4.

Checkpoint Pool and Hypothesis Construction For each of ten seeds, we retain the checkpoint trained with covariance-aware phoneme supervision (Section 2.3) and its corresponding checkpoint after posterior-guided multiscale linguistic fine-tuning (Section 2.4). This gives a pool of 20 checkpoints, comprising ten before and ten after linguistic fine-tuning; the two checkpoints within each seed share a training lineage. Each checkpoint processes the same recording x_i and contributes one phoneme hypothesis and one sentence hypothesis.

Phoneme hypotheses are obtained from the unmapped final-head probabilities using CTC prefix beam search with beam width 50, restricting expansion to the 20 highest-scoring classes at each frame. For each checkpoint, we retain the best collapsed phoneme sequence and its CTC log score. Sentence hypotheses are obtained separately from the same framewise outputs using WFST search with an OpenWebText 5-gram language model, followed by OPT-6.7B rescoring. The decoding configuration uses a 100-best WFST list, acoustic scale 0.325, blank penalty 90, and OPT interpolation weight 0.55. The final sentence score combines the acoustic, 5-gram, and OPT scores with weights 0.325, 0.45, and 0.55, respectively. We retain the decoder's highest-ranked sentence and its total score for each checkpoint.

We pool the 20 checkpoint outputs and rank them by descending sentence score. The phoneme list follows this same ordering, preserving the correspondence between each checkpoint's phoneme and sentence hypotheses. Repeated hypotheses are retained, and checkpoint-family and seed identifiers are omitted from the Qwen prompt. Thus, each prompt contains 20 phoneme hypotheses and 20 sentence hypotheses, with one of each per checkpoint. The 100-best list is internal to WFST decoding; the 20 candidates supplied to Qwen come from the 20 neural checkpoints. Because phoneme search and sentence decoding are performed separately, a checkpoint's phoneme hypothesis need not equal the pronunciation of its selected sentence.

Training Data for Learned Voting We construct the refinement dataset from the labeled utterances available for model development. We randomly divide these utterances into 90% for Qwen training and 10% for Qwen validation. All hypotheses for a given recording remain together in the same subset. The validation subset is held out from parameter updates and used for Qwen checkpoint selection. This subdivision concerns the downstream refinement model; it does not redefine the official partitions used for neural-model training and evaluation. The held-out evaluation partition is used without reference answers in the refinement input.

Each example contains a system instruction, a user prompt, and a supervised assistant response. The user prompt serializes the phoneme and sentence hypotheses as JSON. Each hypothesis includes its rank, predicted sequence, and numerical score. Sentences are lowercase, and phoneme sequences use space-separated ARPAbet symbols with | as the boundary marker. The assistant target consists of the reference transcript s_i on the first line and the reference phoneme sequence y_i on the second line, prefixed by `PHONEMES:`. Reference phonemes provide auxiliary output supervision and are absent from the input hypothesis lists.

Voting Objective & Optimization We fine-tune Qwen3-30B-A3B (Yang et al., 2025) using QLoRA (Dettmers et al., 2023) on one NVIDIA A100 GPU with 40 GB memory. The pretrained language-model weights are frozen and loaded in 4-bit NF4 format with double quantization and bfloat16 computation. LoRA adapters are applied to the attention projections and feed-forward projections, with no trainable bias terms. These language-model adapters are separate from the neural linguistic projection heads used during LENS training.

Let d_i denote the serialized system and user prompt, and let o_i denote the tokenized assistant response containing both s_i and y_i. With frozen language-model parameters ψ_0 and trainable LoRA parameters ϕ, the supervised objective is

$$\mathcal{L}_{\mathrm{LLM}}(\phi) = -\frac{1}{\sum_{i\in\mathcal{I}_{\mathrm{train}}} |o_i|} \sum_{i\in\mathcal{I}_{\mathrm{train}}} \sum_{j=1}^{|o_i|} \log P_{\psi_0,\phi}(o_{i,j} \mid d_i, o_{i,<j}), \tag{24}$$

where $\mathcal{I}_{\mathrm{train}}$ indexes the refinement-training examples. We use the model's chat template and mask system, user, and padding tokens from the loss. The sentence and phoneme outputs share the same autoregressive objective, without a separately weighted phoneme loss. The ensemble is formed at the hypothesis level, with all 20 outputs preserved as inputs to Qwen. Optimization settings are listed in Table 3.

Checkpoint Selection and Inference We evaluate the refinement model on the held-out 10% validation subset and save checkpoints every 100 optimizer updates. At the end of training, we reload and export the checkpoint with the lowest validation assistant-token cross-entropy. Selection therefore accounts for both the transcript and phoneme portions of the response. At inference, the selected model receives the system and user messages without the reference answer and generates greedily, using one beam and at most 64 new tokens. The first nonempty generated line is extracted as the sentence prediction for WER evaluation; the phoneme line is an auxiliary generated output. Qwen does not require the reference transcript, reference phonemes, cached alignments, oracle, or frozen text encoders used for neural-model supervision.

Example Prompt and Supervised Response Table 4 presents a refinement-training example. The displayed candidates are excerpts from the two JSON lists; the complete prompt contains all 20 entries in each list. The example illustrates disagreement across checkpoints: the highest-ranked sentence is singular, whereas the reference transcript is plural. The target rows show the supervised response, rather than a measured Qwen prediction.

Table 3: Core settings for Qwen refinement.

Setting	Value
Base model	Qwen3-30B-A3B-Instruct-2507
Training / validation split	90% / 10%
Fine-tuning	QLoRA; 4-bit NF4 with double quantization; bfloat16 computation
LoRA configuration	Rank 8; scaling parameter 16; dropout 0
Optimization	Paged AdamW with 8-bit states; learning rate 1×10^{-4}; cosine schedule with 5% warm-up; weight decay 0.01
Training duration	5 epochs
Effective batch size	16 examples
Maximum sequence length	4,096 tokens, including prompt and target
Training seed	42
Checkpoint selection	Minimum validation assistant-token loss
Generation	Greedy

Table 4: A refinement-training example. Candidate ranks follow descending sentence score. Phoneme and sentence scores are shown beside their respective hypotheses. The actual input is JSON; selected ranks are reformatted here for readability.

Component	Content
System instruction	You are a brain-to-text correction model. Given candidate phoneme decodings and sentence hypotheses with scores, reconstruct the most likely intended sentence and its phoneme path. Return the sentence in lowercase and the phoneme path as space-separated ARPAbet phonemes with `|` marking word boundaries. No explanation.
User instruction	Predict the intended sentence and its phoneme path from this brain-to-text decoder output. Use both phoneme hypotheses and sentence hypotheses. Answer with the sentence on the first line, then `PHONEMES: <path>`.
Rank 1	**Sentence:** the commercial district (-48.5188). **Phonemes:** `DH AH | F AH M ER SH AH L | S IH S T R IH K T S |` (-0.7634).
Rank 15	**Sentence:** the commercial districts (-54.0483). **Phonemes:** `DH AH | F M ER SH AH L | S IH S T R IH K T S |` (-1.0665).
Rank 19	**Sentence:** the commercial trucks (-56.0773). **Phonemes:** `DH AH | K AH M ER SH AH L | T IH S T R IH K T S |` (-1.1846).
Target, line 1	`the commercial districts`
Target, line 2	`PHONEMES: DH AH | K AH M ER SH AH L | D IH S T R IH K T S |`

E Phoneme Embedding Analysis

Figure 3 visualizes validation-set phoneme embeddings for the original CTC-based model, CAPS, and LENS. A point corresponds to one ground-truth phoneme occurrence. For occurrence i at layer ℓ, its embedding is the CTC forward–backward posterior-weighted average of the frame-level hidden states. We L2-normalize each occurrence embedding before fitting a separate two-dimensional PCA projection for each model and layer. The left panels show embeddings of layers 2, 4, and 6. Each color encode a phoneme context label. Vowels are assigned to the *vowel* group. A consonant is *prevocalic* if at least one vowel appears later in the same word's ground-truth phoneme sequence; otherwise it is assigned to the *coda* group. For example, the sequence `S T R IY T` produces three prevocalic consonants (`S`, `T`, `R`) and one coda consonant (the final `T`).

The right panels quantify the layer-6 embedding geometry of the CTC model and CAPS. Let $z_i = \frac{h_i^{(6)}}{\|h_i^{(6)}\|_2}$, $d(i,j) = \|z_i - z_j\|_2$. For context groups A and B, the between-group distance is the exact

mean across all occurrence pairs,

$$D(A,B) = \frac{1}{|A||B|} \sum_{i \in A} \sum_{j \in B} d(i,j), \quad A \neq B. \tag{25}$$

For within-group distances, self-pairs are excluded:

$$D(A,A) = \frac{1}{|A|(|A|-1)} \sum_{i \neq j;\, i,j \in A} d(i,j). \tag{26}$$

To remove model-specific differences in overall embedding spread, every seed/model matrix is divided by its global mean distance over all validation occurrence pairs,

$$G = \frac{1}{N(N-1)} \sum_{i \neq j} d(i,j), \qquad \widetilde{D}(A,B) = \frac{D(A,B)}{G}. \tag{27}$$

Consequently, a value above one denotes a group-pair distance larger than the model's overall distance, whereas a value below one denotes a more compact pair. The bars summarize the mean of $\widetilde{D}(A,B)$ across ten paired runs, and error bars show the sample standard deviation across seeds.

F [SENT]-TO-FRAME ATTENTION MAP ANALYSIS

Figure 4 compares the direct attention from the `[Sent]` token to neural-frame tokens for the PMLA model and an identical model trained without word- and sentence-level linguistic supervision from two utterances in the training and validation set of B2T'24 respectively. For seed s, layer ℓ, attention head h, and valid neural frame t, let $A_{s,\ell,h}[\texttt{[Sent]}, t]$ denote the post-softmax attention probability. We first average over heads,

$$a_{s,\ell}(t) = \frac{1}{H} \sum_{h=1}^{H} A_{s,\ell,h}[\texttt{[Sent]}, t], \tag{28}$$

and then average over the ten seeds to obtain $\bar{a}_\ell(t) = \frac{1}{10} \sum_s a_{s,\ell}(t)$. For each layer $\ell \in \{2, \ldots, 7\}$, we retain only the 30% of valid frames with the largest values of $\bar{a}_\ell(t)$. Denoting this set by $\mathcal{T}_\ell$, the displayed blue curve is

$$r(t) = \frac{\sum_{\ell=2}^{7} \bar{a}_\ell(t)\, \mathbf{1}[t \in \mathcal{T}_\ell]}{\sum_u \sum_{\ell=2}^{7} \bar{a}_\ell(u)\, \mathbf{1}[u \in \mathcal{T}_\ell]}. \tag{29}$$

Thus, the blue trace summarizes direct `[Sent]`-to-frame attention over layers. The red region is the word posterior obtained in B averaged over seeds.

G IMPLEMENTATION DETAILS

G.1 COMPUTE SOURCES

CAPS training was performed on NVIDIA H100 GPUs with 80 GB of memory for both Brain-to-Text '24 and '25. PMLA training used NVIDIA A100 GPUs with 40 GB of memory for Brain-to-Text '24 and NVIDIA H100 GPUs for Brain-to-Text '25. The Qwen3-30B-A3B aggregation model was fine-tuned on NVIDIA H200 GPUs with 141 GB of memory. The A100 nodes were equipped with two AMD EPYC 7302 16-core processors, while the H100 and H200 nodes used two AMD EPYC 9454 48-core processors. The Brain-to-Text '25 PMLA job configuration requested one GPU, 20 CPU units, and 96 GiB of host memory, with a memory limit of 128 GiB.

G.2 TRAINING PROCEDURE

Initial training and linguistic fine-tuning. We distinguish initial phoneme-decoder training from subsequent posterior-guided multiscale linguistic alignment (PMLA). The neural architecture and initial training settings are reported in Table 5, while Table 6 specifies the CAPS credit kernel. PMLA starts from the corresponding pretrained phoneme decoder and uses standard CTC together with the word and sentence objectives defined in Section 2.4. Text embeddings and source-model CTC alignment posteriors are precomputed and remain fixed during fine-tuning.

Projection-head preparation. We first train the word and sentence projection heads and their separate temperature parameters for 20 epochs. The pretrained neural encoder, phoneme heads, and newly appended `[Sent]` token remain frozen. Both auxiliary objectives have unit weight during this stage; CTC does not contribute to parameter updates.

Full fine-tuning. We subsequently unfreeze the neural encoder, phoneme heads, and `[Sent]` token and fine-tune for 600 epochs. Separate learning-rate groups cover the front end, early Transformer blocks 1–4, late Transformer blocks 5–7, phoneme heads, and linguistic projection heads. The late-encoder group also contains the final encoder normalization and `[Sent]` token; the projection-head group includes the contrastive temperature parameters.

Each group's learning rate increases linearly to its peak over the first five epochs and then follows cosine decay toward 10% of its peak. Gradient clipping is disabled during full fine-tuning on both datasets. We select the checkpoint with the lowest validation phoneme error rate (PER). The fine-tuning learning rates and weight decay are distinct from those used during initial phoneme training.

Auxiliary-loss scheduling. Using the notation of Eq. equation 11, both $\lambda_w^{(n)}$ and $\lambda_s^{(n)}$ increase linearly during epochs 1–5 and remain at their peak through epoch 60. They subsequently follow cosine decay, with the word weight becoming exactly zero from epoch 301 and the sentence weight becoming exactly zero from epoch 421. The final 180 epochs therefore optimize only the phoneme objective. The word-loss mixture uses $\eta = 0.7$, and the intermediate-CTC coefficient is $\lambda_{\text{int}} = 0.3$.

All epoch numbers are one-based, and schedules advance by epoch. For Brain-to-Text '25, each fine-tuning epoch visits all 8,072 training trials once and retains the final incomplete batch, giving 127 optimizer updates at batch size 64 without gradient accumulation. The '24 loader drops the final incomplete training batch.

Oracle preparation. The target-conditioned oracle uses the reference-phoneme encoder and attention-based fusion described in Appendix C. Preparation uses 10 warm-up epochs with the neural backbone frozen, followed by 100 joint-fitting epochs. We use AdamW, batch size 64, weight decay 10^{-5}, and learning rates of 5×10^{-4} for the reference encoder and fusion module and 10^{-4} for the unfrozen neural branch. Joint fitting uses cosine decay toward 10^{-5} and a posterior-matching weight ramped to 2 over five epochs. We retain the final checkpoint for alignment extraction. PMLA starts from the original pretrained decoder and uses only the oracle's cached agreement information.

G.3 Hyperparameter Settings

We list shared PMLA hyperparameters once and use separate dataset columns only for settings that differ. The reported PMLA values correspond to the default configuration.

Table 5: Neural encoder architecture and initial phoneme-training hyperparameters.

Type	Hyperparameter	Brain-to-Text '24	Brain-to-Text '25
Neural Encoder	Input feature dimension	256	512
	Temporal patch size / stride	4 / 4	4 / 4
	Transformer layers	7	7
	Hidden dimension	384	512
	Attention heads	6	8
	Feed-forward expansion ratio	4	4
	Positional encoding	RoPE	RoPE
	Input dropout	0.20	0.20
	Transformer dropout	0.35	0.35
	Stochastic-depth rate	0.20	0.20
	Intermediate supervision layers	$\{2, 4, 6\}$	$\{2, 4, 6\}$
Optimization	Optimizer	AdamW	AdamW
	Adam coefficients (β_1, β_2)	$(0.9, 0.999)$	$(0.9, 0.999)$
	Weight decay	10^{-5}	10^{-5}
	Initial learning rate	10^{-3}	10^{-4}
	Scheduler	Multi-step	Multi-step
	Training epochs	600	600
	Learning-rate milestone	300	300
	Learning-rate decay factor	0.1	0.3
	Training microbatch size	64	16
	Gradient accumulation steps	1	4
	Effective batch size	64	64
	Final-head CTC weight	1.0	1.0
	Intermediate CTC weight	0.3	0.3
Augmentation	Additive Gaussian noise, SD	0.20	0.20
	Per-trial feature offset, SD	0.05	0.05
	Temporal masks per example	20	20
	Maximum individual mask fraction	0.075	0.075

Table 6: Hyperparameters for covariance-aware relaxation of phoneme supervision.

Hyperparameter	Brain-to-Text '24	Brain-to-Text '25
Trace-scaled ridge coefficient λ_Σ	10^{-3}	10^{-3}
Alternative temperature τ_{ph}	0.03	0.03
Alternative-credit budget ρ	0.04	0.02
Kernel normalization	Column-stochastic	Column-stochastic
Blank and SIL	Identity-only	Identity-only
Intermediate-head kernel	Same as final head	Same as final head

Table 7: PMLA optimization settings shared by Brain-to-Text '24 and '25. These settings apply after initial phoneme training and are separate from offline oracle fitting.

Hyperparameter	Shared value
Both PMLA stages	
Optimizer	AdamW
Adam coefficients (β_1, β_2)	$(0.9, 0.999)$
Weight decay	10^{-4}
Training / validation batch size	$64/32$
Gradient accumulation steps	1
Learning-rate decay	Cosine
Final-to-peak learning-rate ratio	0.1
Projection-head preparation	
Training epochs	20
Peak learning rate	10^{-3}
Learning-rate warm-up epochs	0
Trainable parameters	Word and sentence projections; both temperatures
Encoder, phoneme heads, and `[Sent]`	Frozen
Word / sentence loss weights	$1.0/1.0$
Full fine-tuning	
Training epochs	600
Learning-rate warm-up epochs	5
Gradient clipping	Disabled
Final-head CTC coefficient	1.0
Intermediate CTC coefficient λ_{int}	0.3, applied to the mean over heads
Intermediate supervision layers $\mathcal{H}$	$\{2, 4, 6\}$
Checkpoint selection	Lowest validation PER

Table 8: PMLA representation and auxiliary-loss settings shared by Brain-to-Text '24 and '25. Auxiliary schedules refer to full fine-tuning; epoch numbers are one-based.

Hyperparameter	Shared value
Representations and confidence	
Word representation layer ℓ_w	5
Sentence representation layer	7
Frozen word embeddings	FastText, 300 dimensions
Frozen sentence embeddings	E5-large-v2, 1024 dimensions
Word projection hidden dimension	384
Word projection dropout	0.1
Initial temperatures τ_w, τ_s	0.07 each; learned separately
Word-pooling weights	Cached source-model CTC posteriors $q_{k,t}$
Agreement measure	Soft IoU of source and oracle occupancies
Confidence floor $c_{\min}$	0.5
Auxiliary objectives and schedules	
Global / within-utterance word coefficients	$\eta = 0.7$ / $1 - \eta = 0.3$
Sentence objective	Symmetric contrastive loss
Repeated words / identical transcripts	Not treated as false negatives
Auxiliary-weight ramp epochs	5
Peak-weight hold ends after epoch	60
Auxiliary-weight decay	Cosine
Word loss zero from epoch	301
Sentence loss zero from epoch	421
Final CTC-only epochs	180

Table 9: Dataset-specific settings for the default PMLA fine-tuning configuration. Learning rates are peak values, rather than the learning rates at the first warm-up epoch.

Hyperparameter	Brain-to-Text '24	Brain-to-Text '25
Projection-head learning rate	2×10^{-4}	10^{-5}
Late-encoder learning rate	2×10^{-4}	10^{-5}
Phoneme-head learning rate	10^{-4}	5×10^{-6}
Early-encoder learning rate	7.5×10^{-5}	3.75×10^{-6}
Front-end learning rate	5×10^{-5}	2.5×10^{-6}
Peak word weight $\lambda_w^{\max}$	1.0	0.25
Peak sentence weight $\lambda_s^{\max}$	0.5	0.125