MOST-OPSD: DUAL-LEVEL SPEECH-PRIOR DISTILLATION FOR HIGH-PERFORMANCE LANGUAGE BCIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language brain-computer interfaces (BCIs) offer promising approaches to restoring communication, but accurate speech or text decoding remains challenging. Existing methods mainly improve neural decoding with transcript supervision alone by using phoneme context or enhancing temporal modeling. However, decoding models struggle to learn reasonable continuous speech structures from discrete phoneme label sequences, thereby limiting decoding performance and generalization. Here, we propose the **MOST-OPSD** neural decoding framework, combining **M**ixture **o**f **S**peech **T**eachers with privileged **O**n-**P**olicy **S**elf-**D**istillation, which transfers transcript-generated speech priors to the neural decoder at the representation and prediction levels. Specifically, the MOST module transfers structured speech representations across multiple neural stages via independent projections and length-aware Soft-DTW loss, while the OPSD module distills speech-conditioned phoneme distributions by querying speech features using current neural states and applying a shared phoneme head. On the *Brain-to-Text Benchmark '24*, the complete MOST-OPSD neural decoding system with the language models achieved state-of-the-art performance, achieving a word error rate (WER) of **3.63%** on the hidden test set. Supplementary experiments on the second dataset *Brain-to-Text '25* also show that our method significantly outperforms baselines. The results indicate that the proposed dual-level speech-prior distillation framework provides a promising technical path for high-performance language BCIs.

1 INTRODUCTION

Language brain-computer interfaces (BCIs) aim to restore communication by translating neural activity associated with attempted speech into text or speech for people with severe speech impairments (Willett et al., 2023; Metzger et al., 2023; Card et al., 2024; Tan et al., 2024; Wairagkar et al., 2025; Littlejohn et al., 2025; Jia et al., 2025; Li et al., 2025; Feghhi et al., 2026). Despite progress in language decoding, accurately recovering the attempted phoneme sequences from the limited neural activity remains a central challenge (Li et al., 2025; Feghhi et al., 2026).

Existing approaches mainly enhance speech decoding by improving the sequence modeling from neural activity to phonemes. For example, recurrent decoders trained with connectionist temporal classification (CTC, (Graves et al., 2006)) learn phoneme sequences without frame-level labels and combine their outputs with language models to generate sentences (Willett et al., 2023; Card et al., 2024). To explicitly represent local phonetic context, DCoND-LIFT predicts diphone distributions and marginalizes over them to obtain phoneme probabilities, allowing neighboring phonemes to inform each decision (Li et al., 2025). Time-Masked Transformers (TMT) instead strengthen neural sequence modeling with a compact causal Transformer and temporal masking that reduces overfitting (Feghhi et al., 2026). These advances have gradually improved how neural evidence is mapped to phonemes. However, the neural decoder may still have difficulty in learning the underlying speech process and reasonable phoneme distribution merely by optimizing the neural-to-phoneme mapping alone, limiting decoding performance and generalization.

Pretrained speech models offer such complementary representations that encode acoustic, phonetic, and contextual information (Baevski et al., 2020; Hsu et al., 2021a; Pasad et al., 2021). The key question is how to convert this prior knowledge into effective supervision for the neural decoder. BrainTalker supervises an electrocorticography (ECoG) encoder with Wav2Vec 2.0 encoder's repre-

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.

sentations of recorded speech for low-resource speech synthesis (Kim et al., 2023). More recently, Brain2Speech-Net uses a neural phoneme bottleneck and monotonic alignment to a pretrained text-to-speech (TTS) latent space for intracortical speech synthesis (Chandra et al., 2026). Although these approaches establish the value of certain speech-related representations for neural decoding, speech-prior information has not yet been fully utilized. Existing analyses show that different types of speech-prior information, such as acoustic, phonetic, and contextual information, are emphasized at different depths within pretrained speech encoders, rather than being uniformly preserved in a single representation (Pasad et al., 2021). Moreover, speech-derived phonetic posteriors can retain fine-grained support for acoustically related alternatives (Churchwell et al., 2024). This motivates us to consider two complementary ways of speech-prior distillation simultaneously: multi-level representation alignment and predictive posterior supervision. While the former constrains how neural states represent speech, the latter explicitly guides the phoneme decisions based on those states.

Here, we propose **MOST-OPSD**, **M**ixture **o**f **S**peech **T**eachers with privileged **O**n-**P**olicy **S**elf-**D**istillation, a neural decoding framework for joint representation- and prediction-level speech-prior transfer. At the representation level, the MOST module extracts complementary speech teacher views of transcript-generated speech from a frozen pretrained speech encoder and aligns them with corresponding neural stages through independent spatial projections and length-aware Soft-DTW (Cuturi & Blondel, 2017). This provides constraints on multiple aspects of speech structure while allowing flexible neural-speech correspondence. At the predictive distribution level, the privileged OPSD module uses current neural states to query deep speech features and constructs a CTC-supervised privileged posterior through a shared phoneme head. The privileged posterior trained on the reference phonemes provides additional speech-derived framewise distribution targets for the neural-only students. Together, these two modules provide complementary speech priors, allowing the neural decoder to learn how to represent speech while also making appropriate phoneme decisions. Experiments on the *Brain-to-Text Benchmark '24* [1] show that the complete system consisting of the proposed MOST-OPSD and language models achieves state-of-the-art performance with a word error rate (WER) of **3.63%** on hidden test data. The results demonstrate that speech-prior distillation with complementary hierarchies further enhances neural decoding models and provides a promising technical path for high-performance language BCIs.

2 Related Work

Neural Speech Decoding. Neural speech decoding maps neural activity to linguistic outputs. Willett et al. established a cascaded approach that predicts phoneme probabilities from intracortical activity and converts them into text through language models (Willett et al., 2023). Within this framework, Li et al. introduced context-dependent targets by modelling diphone distributions and marginalizing over them to obtain phoneme predictions (Li et al., 2025). Feghhi et al. improved temporal modelling and computational efficiency with a compact causal Transformer, time masking, and lightweight test-time adaptation (Feghhi et al., 2026). Beyond task-specific training, Zhang et al. pretrained a neural encoder across tasks and species and integrated it with an audio-capable language model for end-to-end text decoding (Zhang et al., 2026). However, the proposed MOST-OPSD complements these advances by enriching the decoder's supervision with pretrained speech knowledge. It transfers speech structure to intermediate neural states and speech-conditioned predictive information to phoneme outputs, thereby strengthening the neural decoder itself.

Cross-Modal Transfer and Self-Distillation. Cross-modal transfer and self-distillation use auxiliary representations or predictive views to provide supervision beyond task labels (Hinton et al., 2015; Gupta et al., 2016). At the representation level, BrainTalker matches ECoG features to speech representations for synthesis (Kim et al., 2023), while Brain2Speech-Net aligns a neural bottleneck to a pretrained TTS latent space (Chandra et al., 2026). At the prediction level, Generalized Knowledge Distillation obtains teacher feedback on student-generated sequences (Agarwal et al., 2024). Self-Distilled Reasoner further constructs teacher and student distributions within a shared language model by varying access to privileged context and distills along student-generated rollouts (Zhao et al., 2026). MOST-OPSD extends representation- and prediction-level transfer principles to neural decoding, where generated speech and neural activity provide asynchronous training views. The two complementary mechanisms jointly transfer speech structure and prediction knowledge to the neural decoder in the training phase, without retaining the speech teachers during inference.

[1] https://eval.ai/web/challenges/challenge-page/2099/leaderboard/4944

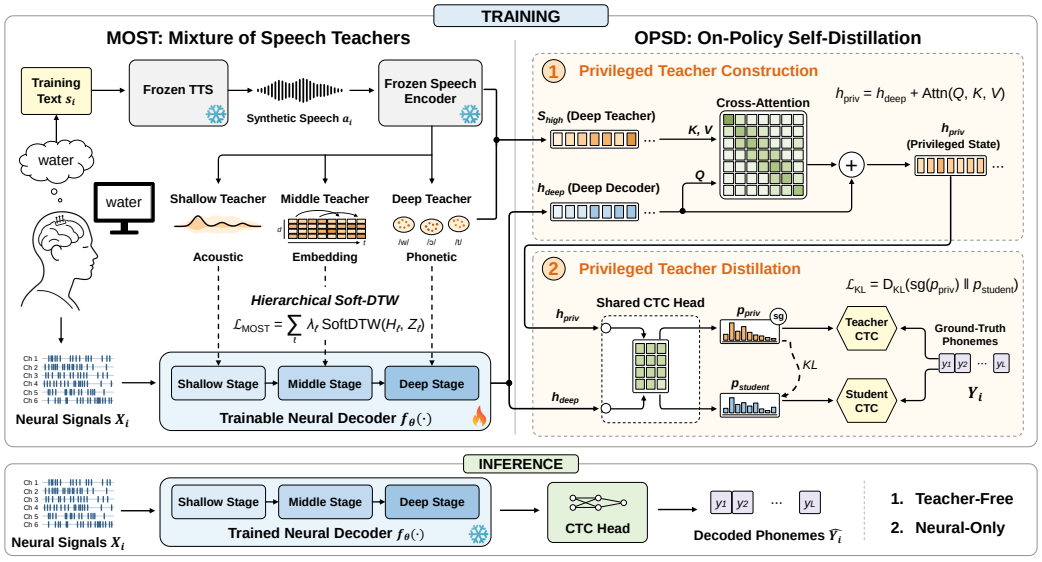

Figure 1: Overview of the MOST-OPSD framework. **Left**: transcript-generated speech supplies frozen speech teachers for multi-stage Soft-DTW alignment. **Right**: deep neural states query speech features to construct a privileged state. Privileged and neural-only views share a CTC head and receive parallel ground-truth CTC supervision. **Bottom**: inference retains only the neural decoder.

3 METHODOLOGY

3.1 OVERVIEW AND PROBLEM SETUP

Figure 1 provides an overview of the MOST-OPSD framework for neural speech decoding. The MOST module organizes intermediate neural states using three complementary frozen speech teachers. And the OPSD module converts deep speech context into a task-trained posterior on the neural time grid and transfers it to the student branch. Structural targets are thus fixed, while predictive targets track the evolving neural decoder. Training uses labelled neural recordings and transcript-generated speech; the deployed predictor receives neither the current transcript nor speech features.

We consider decoding a neural feature sequence $X_i \in \mathbb{R}^{T_i \times C}$ (T_i is the number of frames and C is the feature dimension) into a reference phoneme sequence Y_i, with s_i denoting its training transcript. To obtain the corresponding speech representations for transcription, we first generate pseudo speech using a frozen Piper TTS model (Hansen et al., 2023), and then extract K teacher views via a frozen speech encoder $E(\cdot)$:

$$a_i = \mathrm{TTS}(s_i), \qquad Z_i^{(k)} = E^{(k)}(a_i), \quad k = 1, \ldots, K, \tag{1}$$

where $Z_i^{(k)} \in \mathbb{R}^{U_i^{(k)} \times d_s}$ has valid length $m_i^{(k)}$, and $K = 3$ in our implementation. The teachers share one encoder and remain frozen during neural training, with the main configuration using *Wav2Vec 2.0 base* (Baevski et al., 2020). The neural encoder $f_\theta(\cdot)$ and linear phoneme head $g_\psi(\cdot)$ produce:

$$(H_i^{(1)}, \ldots, H_i^{(L)}) = f_\theta(X_i), \qquad \ell_i^{\mathrm{neu}} = g_\psi(H_i^{\mathrm{neu}}), \qquad p_i^{\mathrm{neu}} = \mathrm{Softmax}(g_\psi(H_i^{(L)})), \tag{2}$$

where H_i represents the neural representations at different stages, and p_i^{neu} is the phoneme probability output by the decoder. Each neural-representation sequence has a valid length n_i after input processing. The output alphabet $\mathcal{V}_{\mathrm{CTC}}$ contains the task labels and the CTC blank.

3.2 MOST: MULTI-STAGE REPRESENTATION TRANSFER

Attempted and synthesized speech share linguistic content but differ in feature coordinates and temporal organization. For each predefined neural-speech stage pairing (l_k, k), an independent trainable linear map P_k projects neural states into the speech latent space, and the distance is defined $D_{i,tu}^{(k)}$:

$$\widehat{H}_i^{(k)} = P_k(H_i^{(l_k)}), \qquad D_{i,tu}^{(k)} = \left\| \widehat{H}_{i,t}^{(k)} - Z_{i,u}^{(k)} \right\|_2^2. \tag{3}$$

Length-aware Temporal Alignment. We use Soft-DTW to accommodate local timing variations between neural and speech sequences while preserving their temporal order (Cuturi & Blondel, 2017). Suppressing sample and view indices, the cost satisfies:

$$\begin{aligned} A_{tu} &= D_{tu} + \mathrm{Softmin}_\gamma\left(A_{t-1,u}, A_{t,u-1}, A_{t-1,u-1}\right), \\ \mathrm{Softmin}_\gamma(a_1,a_2,a_3) &= -\gamma \log \sum_{j=1}^{3} \exp(-a_j/\gamma), \end{aligned} \tag{4}$$

where D_{tu} is the squared Euclidean distance between neural state t and speech feature u, A_{tu} is the accumulated alignment cost, and γ controls the smoothness of the soft minimum. We use $\gamma = 0.2$. We initialize $A_{00} = 0$ and $A_{t0} = A_{0u} = +\infty$ for positive indices.

For each neural-speech pair, we evaluate the recurrence over its valid lengths and obtain:

$$d_i^{(k)} = A^{(i,k)}_{n_i, m_i^{(k)}} = \mathrm{SoftDTW}_\gamma\left(\widehat{H}^{(k)}_{i,1:n_i}, Z^{(k)}_{i,1:m_i^{(k)}}\right), \tag{5}$$

where $\widehat{H}_i^{(k)}$ and $Z_i^{(k)}$ are the projected neural and frozen speech representations, and n_i and $m_i^{(k)}$ are their valid lengths excluding padding. The scalar $d_i^{(k)}$ is the raw alignment cost before the subsequent length and scale normalization.

Both the forward recursion and gradient propagation are restricted to valid time indices. This length-aware formulation provides flexible temporal correspondence for neural and speech sequences with different durations and local rates, and alleviates the potential mismatch between them.

Length and Scale Normalization. To account for sequence duration and view-dependent loss scales, we normalize each alignment cost as:

$$\widetilde{d}_i^{(k)} = \frac{d_i^{(k)}}{n_i + m_i^{(k)}} \frac{c_{\mathrm{ref}}}{c_k}, \tag{6}$$

where $n_i + m_i^{(k)}$ is the sum of the valid neural and speech sequence lengths. The positive scale c_k is estimated as the median absolute length-normalized alignment cost on a training calibration subset, while c_{ref} is the median absolute unnormalized cost of the deepest reference view. These calibration constants remain fixed throughout training.

We combine the normalized costs using fixed stage weights w_k and average over a minibatch B:

$$\mathcal{L}_{\mathrm{MOST}} = \frac{1}{B} \sum_{i=1}^{B} \sum_{k=1}^{K} w_k \widetilde{d}_i^{(k)}. \tag{7}$$

Together, (1) independent projections bridge the neural and speech feature spaces, (2) length-aware alignment accommodates flexible temporal variability, and (3) multi-stage supervision transfers complex speech structures throughout the neural encoder.

3.3 OPSD: Privileged Predictive Transfer

Constructing Task-aligned Privileged State. OPSD complements representation transfer with speech-conditioned supervision in the phoneme prediction space. The final neural states query the frozen deep speech representation Z_i^{deep} through multi-head cross-attention (Vaswani et al., 2017):

$$R_i = W_O\Big[\mathrm{MHA}\Big(W_Q(H_i^{(L)}), W_K(\mathrm{sg}(Z_i^{\mathrm{deep}})), W_V(\mathrm{sg}(Z_i^{\mathrm{deep}}))\Big)\Big], \tag{8}$$

where W_Q, W_K, W_V, and W_O are trainable linear maps, and $\mathrm{sg}(\cdot)$ denotes stop-gradient. The speech-conditioned residual augments the neural representation:

$$H_i^{\mathrm{priv}} = H_i^{(L)} + R_i. \tag{9}$$

The state H_i^{priv} is mapped to phoneme probabilities using the same head as the neural-only branch:

$$\ell_i^{\mathrm{priv}} = g_\psi(H_i^{\mathrm{priv}}), \qquad p_i^{\mathrm{priv}} = \mathrm{softmax}(\ell_i^{\mathrm{priv}}). \tag{10}$$

Neural queries $H_i^{(L)}$ place the privileged representation Z_i^{deep} on the student's time grid, while the shared head $g_\psi(\cdot)$ expresses both views in a common phoneme decision space. Attention masks restrict speech conditioning to valid positions. CTC supervision trains the privileged branch to convert speech priors into task-aligned predictive distributions for guiding the student branch.

On-Policy Self-Distillation. At each training step, the privileged branch constructs its target from the student's current neural states and the corresponding speech features. This defines the on-policy character of OPSD (Agarwal et al., 2024): the predictive target evolves with the neural decoder and cross-attention adapter, while the speech features remain fixed. For temperature $\tau > 0$, we define:

$$p_{it}^{\tau} = \text{Softmax}(\ell_{it}^{\text{neu}}/\tau), \qquad q_{it}^{\tau} = \text{Softmax}(\text{sg}(\ell_{it}^{\text{priv}})/\tau). \tag{11}$$

Then, we transfer the privileged distribution q_{it}^{τ} to the neural-only student p_{it}^{τ} by minimizing forward KL divergence over all valid neural frames:

$$\begin{aligned} \mathcal{L}_{\text{KD}} &= \frac{\tau^2}{\sum_{i=1}^{B} n_i} \sum_{i=1}^{B} \sum_{t=1}^{n_i} D_{\text{KL}}\left(q_{it}^{\tau} \,\|\, p_{it}^{\tau}\right) \\ &= \frac{\tau^2}{\sum_{i=1}^{B} n_i} \sum_{i=1}^{B} \sum_{t=1}^{n_i} \sum_{v \in \mathcal{V}_{\text{CTC}}} q_{it}^{\tau}(v) \log \frac{q_{it}^{\tau}(v)}{p_{it}^{\tau}(v)}. \end{aligned} \tag{12}$$

The τ^2 factor follows temperature-scaled distillation (Hinton et al., 2015). Matching the full CTC distribution transfers both phoneme preferences and blank/nonblank probability allocation. Stop-gradient holds the privileged target fixed within each backward pass. The KL objective updates the neural encoder and shared head through the student distribution, while privileged CTC supervision updates the adapter, neural decoder, and shared head through the detached privileged branch.

3.4 Learning Objective and Neural-Only Inference

Both the neural-only and privileged views receive connectionist temporal classification (CTC) supervision (Graves et al., 2006). Let $\Pi_i(y_i)$ denote the set of length-n_i paths over $\mathcal{V}_{\text{CTC}}$ that map to y_i under the collapse operator $\mathcal{B}$, which merges consecutive repeated labels and removes blanks. For branch $b \in \{\text{neu}, \text{priv}\}$, the per-example CTC loss is:

$$\mathcal{J}_{i,\text{CTC}}^{b} = -\log \left[\sum_{\pi \in \Pi_i(y_i)} \prod_{t=1}^{n_i} p_{it}^{b}(\pi_t) \right]. \tag{13}$$

Then, denoting their minibatch-reduced losses by $\mathcal{L}_{\text{CTC}}^{\text{neu}}$ and $\mathcal{L}_{\text{CTC}}^{\text{priv}}$, we optimize:

$$\mathcal{L} = \frac{\mathcal{L}_{\text{CTC}}^{\text{neu}} + \beta \mathcal{L}_{\text{CTC}}^{\text{priv}}}{1+\beta} + \lambda \mathcal{L}_{\text{MOST}} + \mu \mathcal{L}_{\text{KD}}. \tag{14}$$

The nonnegative coefficients β, λ, and μ control privileged CTC supervision, representation transfer, and predictive distillation, respectively. The factor $1+\beta$ forms a weighted average of the two CTC losses. Together, the MOST and OPSD modules transfer complementary speech-prior knowledge into intermediate neural representations and phoneme predictions, coupling representation transfer learning with task-aligned predictive supervision.

At inference, the neural encoder, including its recording-day input transformation, and the shared phoneme head produce p_i^{neu}. TTS model, the speech encoder, alignment projections, and the privileged branch are used only during training. Greedy phoneme decoding is given by:

$$\widehat{y}_i = \mathcal{B}\left(\left[\operatorname*{argmax}_{v \in \mathcal{V}_{\text{CTC}}} p_{it}^{\text{neu}}(v) \right]_{t=1}^{n_i} \right). \tag{15}$$

The same neural outputs also support language-assisted text decoding (Appendix B.5). Thus, dual-level speech supervision enriches training while preserving the neural-only prediction architecture. Implementation details and optimization settings are provided in Appendices B and D.

4 EXPERIMENTS AND RESULTS

4.1 EXPERIMENTAL SETUP

Datasets and Evaluation. We evaluate MOST-OPSD on the *Brain-to-Text Benchmark '24* using intracortical recordings of attempted speech (Willett et al., 2023). The training set contains $8{,}800$ samples, and validation comprises samples across 15 recording days. Each neural time step contains 256 input features, and the output vocabulary consists of 39 phonemes, a silence label, and the CTC blank. Additional experiments on *Brain-to-Text '25* provide a second dataset for evaluating the framework under a within-participant protocol (Card et al., 2024) (Appendix C.1).

Implementation. The neural decoder is a six-layer bidirectional GRU with $1{,}024$ hidden units per direction and recording-day input transformations. Neural inputs undergo Gaussian smoothing and temporal windowing with width 32 and stride 4. Training transcripts are synthesized using the *Piper HFC female* voice (Hansen et al., 2023) and encoded by frozen *Wav2Vec 2.0 Base* (Baevski et al., 2020). MOST aligns the hidden-state sequences after layers 2, 4, and 6 with speech features from Transformer layers 6, 8, and 10, respectively; OPSD uses the final neural states to query the deepest selected speech view. Training uses batch size 32, and evaluation uses the neural-only branch. Further implementation details and hyperparameters appear in Appendices B and D.

Baselines and Metrics. We compare MOST-OPSD with GRU (Willett et al., 2023), DCoND-LIFT (Li et al., 2025), TMT (Feghhi et al., 2026), and BIT (Zhang et al., 2026). Neural decoding is evaluated using greedy phoneme error rate (PER) and CTC loss on the validation set. Text decoding is evaluated using word error rate (WER). For single-acoustic-model evaluation, neural outputs are decoded with a 5-gram WFST language model and rescored by OPT-6.7B (Willett et al., 2023). Complete fusion systems are reported as system-level results (Feghhi et al., 2026). Our repeated measurements are summarized as mean $\pm$ sample standard deviation; published results retain their original uncertainty definitions. Appendices A and B.5 provide more details.

4.2 MAIN BENCHMARK RESULTS

Neural Decoding. We first investigate whether speech-prior supervision strengthens neural-only phoneme decoding. We compare greedy PER and CTC loss on the validation set. MOST-OPSD achieves the best **10.66±0.10%** PER and **0.46±0.01** CTC loss (Table 1), which significantly outperforms the GRU and TMT baselines ($p < 0.0001$, two-sided paired t-test, n=5). Its lower mean PER extends across **all 15 recording days**, demonstrating consistent gains across the evaluated recordings (Figure 3(a)). These results show that dual-level speech supervision yields an effective neural-only decoder, with the benefits of speech-informed training preserved at inference.

Text Decoding with Language Models. We next assess the downstream value of speech-informed neural decoding for text reconstruction. We evaluate hidden-test WER using single-acoustic-model outputs with 5-gram WFST decoding and OPT-6.7B rescoring, alongside complete fusion systems (Appendix B.5). Single-model decoding achieves **6.25±0.14%** WER, compared with 8.18±0.22% for TMT and 8.39±0.22% for DCoND-LIFT, yielding a **1.93%** point reduction relative to TMT. The complete fusion system of MOST-OPSD achieves **3.63%** WER, surpassing BIT's 5.10% and TMT's 5.68% and ranking **1st** among benchmark submissions. Together, these results establish the value of speech-informed neural decoding for high-performance text reconstruction.

We also evaluate MOST-OPSD on the *Brain-to-Text '25* to assess its effectiveness on a second neural speech dataset, which further supports the applicability of dual-level speech-prior transfer, with complete benchmark comparisons and fusion-system results provided in Appendix C.1.

4.3 ABLATION STUDY

Complementarity of Representation and Prediction Transfer. We examine the complementary benefits of representation and prediction transfer for neural decoding. Figure 2(a) compares MOST-OPSD with *CTC-only*, *w/o OPSD*, and *w/o MOST*. MOST-OPSD achieves **10.66±0.10%** PER, compared with 14.90±0.25%, 11.46±0.25%, and 13.50±0.04%, respectively, yielding reductions of **4.25**, **0.80**, and **2.84** percentage points. The complete model also achieves the lowest CTC loss, providing further evidence of improved sequence prediction (Figure 5). These results are consistent

Table 1: Neural decoding performance on the *Brain-to-Text Benchmark '24*. PER and WER are percentages (%); lower is better. Validation metrics, single-model WER, and fusion-system WER are distinct evaluation endpoints. **Bold** denotes the best. Ranks describe submitted systems.

Method	Val PER	Val CTC loss	Hidden WER (single model)	Hidden WER (fusion system)	Rank
GRU	21.70 ± 0.22	0.81 ± 0.01	11.12 ± 0.13	–	25
DCoND-LIFT	–	–	8.39 ± 0.22	5.77	8
TMT	15.77 ± 0.60	0.79 ± 0.06	8.18 ± 0.22	5.68	6
BIT	–	–	–	5.10	5
MOST-OPSD	$\mathbf{10.66 \pm 0.10}$	$\mathbf{0.46 \pm 0.01}$	$\mathbf{6.25 \pm 0.14}$	**3.63**	**1**

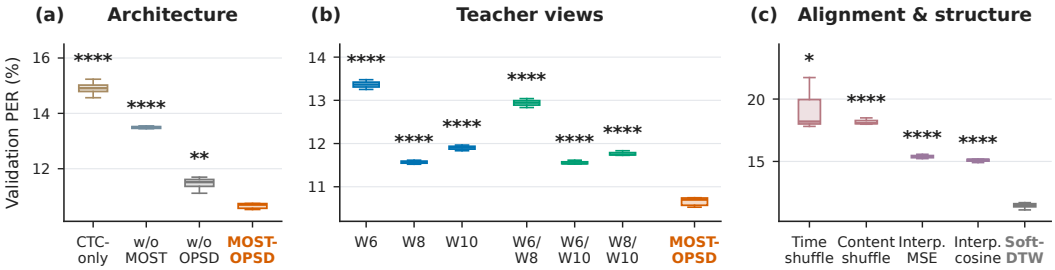

Figure 2: Ablation study. Validation PER for **(a)** four supervision configurations, **(b)** three single- and two-teacher views, and **(c)** two speech-shuffling controls; interpolation with MSE or cosine loss compared with Soft-DTW. Boxes show medians and IQRs with 1.5-IQR whiskers. Significance symbols use two-sided t-tests against MOST-OPSD in (a) & (b) and Soft-DTW in (c).

with the complementary roles of MOST in organizing intermediate neural representations and OPSD in transferring task-aligned predictive distributions.

Contributions of Teacher Views. We next examine how different speech-teacher views contribute through their assigned neural stages. We evaluate three single-view and three two-view configurations, preserving the predefined speech-neural stage pairings (Figure 2(b)). Using W6, W8, or W10 alone yields PERs of 13.37±0.11%, 11.57±0.05%, and 11.90±0.07%, respectively, showing particularly effective supervision from the selected views. Among the two-view configurations, W6/W10 achieves the lowest PER of 11.56±0.05%, compared with 12.94±0.10% for W6/W8 and 11.77±0.06% for W8/W10. The variations highlight the importance of organizing speech supervision through appropriate teacher-view and neural-stage pairings.

Speech Structure and Temporal Correspondence. We next test whether effective transfer depends on correctly associated speech structure (Figure 2(c)). Within the MOST component study, length-aware Soft-DTW achieves **11.46±0.25%** PER, whereas linear interpolation with MSE and cosine loss yields 15.39±0.17% and 15.08±0.15%, respectively. Shuffling speech-time order or substituting length-matched speech content increases PER to 19.24±2.16% and 18.18±0.27%, which is comparable to the performance of the GRU baseline, indicating that the mismatched speech structure can damage the transfer completely. Therefore, effective supervision depends on the transcript-associated speech structure and benefits from flexible temporal correspondence.

Additional ablations examine teacher-layer ordering, neural-stage assignments, transfer weights, speech encoders, and synthesis voices (See Appendix C.2). The results favor structured stage-wise supervision with fixed transfer weights and show strong decoding performance across the evaluated speech encoders and synthesis voices. These findings further support the framework's design choices and its flexibility in constructing training-time speech priors.

4.4 Prediction and Representation Analysis

Coverage Across Days and Phonemes. We first assess how broadly the decoding gains extend across recordings and phoneme decisions. We examine day-wise PER and sequence confusion (Figure 3(a,b)), and the latter evaluates eight typical phonemes which cover a wide range of occurrence

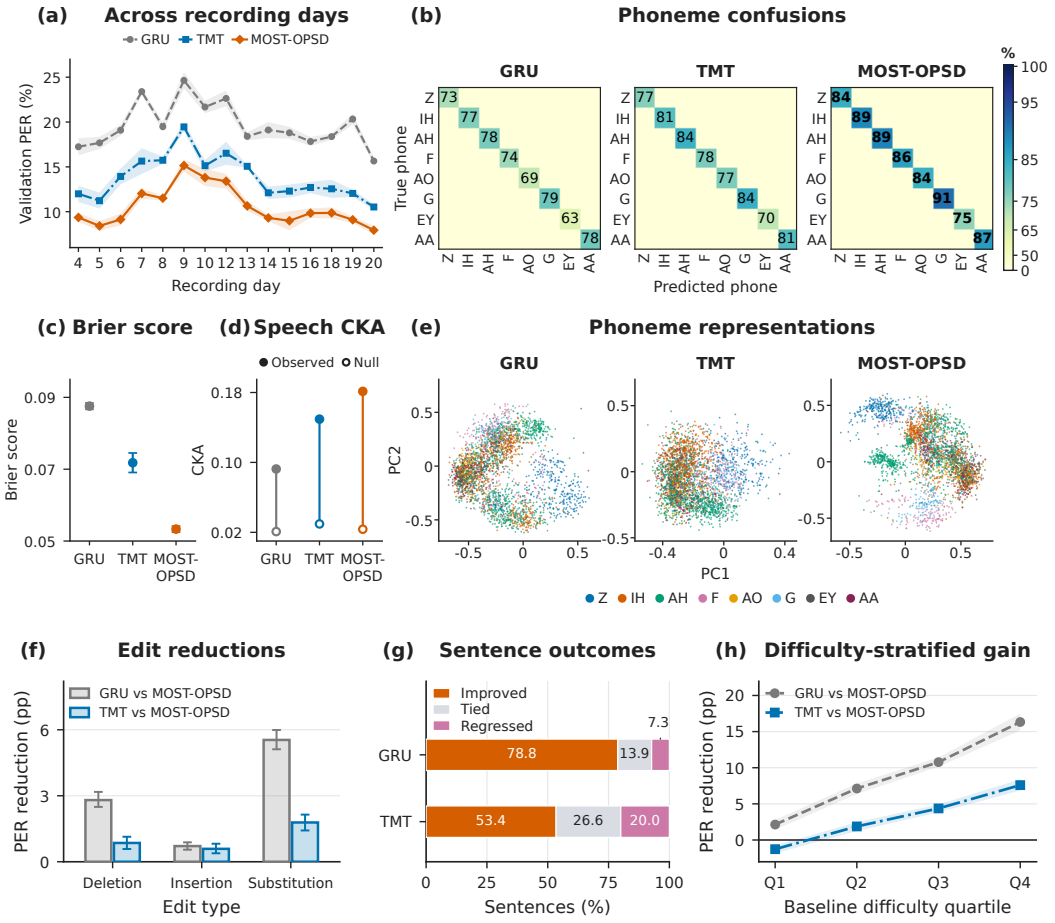

Figure 3: Prediction, representation, and error analyses on the validation set. **(a)** Day-wise PER. **(b)** Sequence confusion for eight typical phonemes. **(c)** Emitted-token Brier scores. **(d)** Temporal CKA. **(e)** Training-fitted PCA in each model's latent space. **(f)** Edit-type reductions. **(g)** Sentence-level edit-count changes. **(h)** PER reductions across baseline-difficulty quartiles. Panels (a,c) summarize means and sample SDs; intervals in (f,h) are descriptive 95% day-bootstrap intervals.

frequencies, with each class containing 148-1,068 occurrences, all covering every validation day. MOST-OPSD achieves lower mean PER on **all 15 recording days** and higher accuracy for all 8 phonemes, including an increase from **81% to 89%** for IH relative to TMT. These improvements demonstrate broad recording coverage and connect the PER gains to concrete phoneme decisions.

Posterior Quality. We next examine whether improved hard decisions are accompanied by better predictive probabilities through proper scoring Gneiting & Raftery (2007). Our method has the lowest emitted-token Brier score of **0.0533±0.0008**, below TMT's 0.0718±0.0027 and GRU's 0.0876±0.0009 (Figure 3(c)). The complementary reference-sequence evaluation also yields a lower CTC proper score: **0.663**, compared with 0.986 for TMT, in bits per reference phoneme (Table 5). Agreement between these token- and sequence-level measures supports improved predictive quality, which may be due to the fact that the OPSD module provides a better privileged posterior teacher to help the model learn a more reasonable predictive distribution.

Speech Association and Representation Geometry. We measure the relationship between neural and speech representations using centered kernel alignment (CKA) (Kornblith et al., 2019). MOST-OPSD reaches an observed CKA of **0.181**, compared with 0.150 for TMT and 0.092 for GRU; its within-sentence temporal-null mean is 0.023 (Figure 3(d), Table 9). This pattern is consistent with a stronger association with temporally organized speech representations. Training-fitted Principal Component Analysis (PCA, (Abdi & Williams, 2010)) complements this statistic by displaying local phoneme geometry (Figure 3(e)); its independently fitted coordinates are interpreted within

each model. These results show that the multi-stage representation alignment of the MOST module enables the decoder to learn more realistic speech structure to improve decoding.

Sources of Decoding Gains. We further identify which errors and sentences contribute to the overall improvement. We decompose phoneme edits by type and examine sentence-level outcomes and baseline-difficulty quartiles (Figure 3(f-h)). MOST-OPSD reduces all three edit types, with substitution reductions of **5.54/1.79%** points and fewer edits in **78.8/53.4%** of sentences relative to GRU/TMT, respectively. The largest reductions occur in the hardest quartile, reaching **16.33/7.59** percentage points, identifying difficult-sentence recovery as a major source of the PER gains.

The appendix extends these results with complete phoneme statistics, posterior reliability and selective-risk analyses, temporal-matching visualizations, and qualitative cases (Appendix C). These complementary analyses connect the aggregate improvements to detailed phoneme predictions, confidence behavior, and neural-speech correspondence.

4.5 Data Efficiency

By providing additional supervisory structure through speech priors, MOST-OPSD may have superior data efficiency and generalization, motivating an evaluation under limited training-data availability. We compare its Wav2Vec 2.0 and HuBERT variants using 25%, 50%, and 100% of the neural training data, measuring validation PER and CTC loss against full-data GRU and TMT references (Figure 4). With 50% of the neural training data, the Wav2Vec 2.0 variant achieves **15.45%** PER compared with 15.77% for full-data TMT; using all available data, the Wav2Vec 2.0 and HuBERT variants reach **10.66%** and **11.07%** PER, with CTC losses of **0.457** and **0.479**, respectively. These results are consistent with the hypothesis that speech-prior supervision supports data-efficient neural decoding, while the continued improvements at larger training fractions suggest that these priors complement the information learned from neural observations.

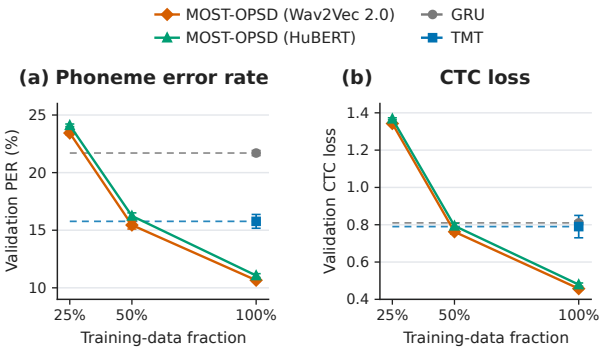

Figure 4: Data efficiency of MOST-OPSD framework with Wav2Vec 2.0 and HuBERT teachers. **(a)** Validation PER and **(b)** validation CTC loss across training-data fractions. Error bars indicate sample standard deviations; dashed lines mark full-data GRU and TMT references.

5 Conclusion and Limitations

Conclusion. We presented MOST-OPSD, a dual-level speech-prior transfer framework that connects transcript-generated speech structure to neural representations and phoneme predictions. Multi-stage, length-aware alignment and speech-conditioned privileged predictive transfer jointly achieve **10.66±0.10%** validation PER on the *Brain-to-Text Benchmark '24*, improving over either transfer level alone. A separately evaluated single neural decoder with language decoding attains **6.25±0.14%** hidden-test WER, while the final complete fusion system reaches **3.91%** WER and ranks **1st** in the benchmark submissions. Prediction and representation analyses support broadly distributed decoding gains and improved predictive quality. Data-efficiency experiments further support the value of reusable speech-prior supervision. These results indicate that our dual-level speech-prior distillation framework provides a promising way for high-performance language BCIs.

Limitations. First, the evaluations address within-participant, offline decoding; generalization to unseen participants and streaming decoding remain to be established. Second, speech-derived priors may only partially capture the neural representations of attempted speech, and their transferability may vary across languages, speaking intentions, and clinical populations. Additionally, neural nonstationarity raises the question of how to sustain effective speech-prior transfer over extended use, warranting further investigation into transfer strategies that adapt to evolving neural representations.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing, including improving grammar, clarity, and presentation. We have not used generative AI tools to generate research results, conduct experiments, analyze data, or produce scientific claims. All AI-assisted text was reviewed and revised by the authors, who verified its correctness and consistency with the underlying work. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work aims to support communication for people with severe speech impairments, using existing neural recordings and transcript-generated speech for computational experiments, building on the data-collection protocols documented in the original studies (Willett et al., 2023; Card et al., 2024). As BCI technology advances, this framework offers a path toward more reliable and transparent neural decoding systems. We believe that advancing such technologies should be guided by ethical principles, particularly regarding user data privacy and transparency. This research is primarily focused on scientific methodology and presents no immediate negative social consequences.

REPRODUCIBILITY STATEMENT

Section 3 specifies the model components, learning objective, and neural-only inference procedure. Appendix A documents the datasets, preprocessing, evaluation protocols, and metrics, while Appendices B and D provide implementation details and training configurations. Appendix C presents complete experimental results and the procedures underlying the quantitative and qualitative analyses. Appendix E provides mathematical details and derivations for the alignment and distillation objectives.

REFERENCES

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

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, Aparna Srinivasan, Eun Young Choi, Matthew F. Glasser, Leigh R. Hochberg, Jaimie M. Henderson, Kiarash Shahlaie, Sergey D. Stavisky, and David M. Brandman. An accurate and rapidly calibrating speech neuroprosthesis. *New England Journal of Medicine*, 391(7):609–618, 2024. doi: 10.1056/NEJMoa2314132. URL https://doi.org/10.1056/NEJMoa2314132.

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.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

Cameron Churchwell, Max Morrison, and Bryan Pardo. High-fidelity neural phonetic posteriorgrams. In *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pp. 823–827. IEEE, 2024.

Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pp. 894–903. PMLR, 2017.

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, 2026.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

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.

Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2827–2836, 2016.

Michael Hansen et al. Piper: A fast, local neural text to speech system. *Piper: A fast, local neural text to speech system*, 2023.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021a.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021b.

Yao Jia, Qi Lian, Lei Wang, Yueming Wang, and Yu Qi. Learning discrete neural latent spaces for high-performance speech decoding. *Journal of Neural Engineering*, 22(6):066039, 2025. doi: 10.1088/1741-2552/ae2ccd. URL https://doi.org/10.1088/1741-2552/ae2ccd.

Miseul Kim, Zhenyu Piao, Jihyun Lee, and Hong-Goo Kang. Braintalker: Low-resource brain-to-speech synthesis with transfer learning using wav2vec 2.0. In *2023 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 1–5. IEEE, 2023.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

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.

Kaylo T. Littlejohn, Cheol Jun Cho, Jessie R. Liu, Alexander B. Silva, Bohan Yu, Vanessa R. Anderson, Cady M. Kurtz-Miott, Samantha Brosler, Anshul P. Kashyap, Irina P. Hallinan, Adit Shah, Adelyn Tu-Chan, Karunesh Ganguly, David A. Moses, Edward F. Chang, and Gopala K. Anumanchipalli. A streaming brain-to-voice neuroprosthesis to restore naturalistic communication. *Nature Neuroscience*, 28(4):902–912, 2025. doi: 10.1038/s41593-025-01905-6. URL https://doi.org/10.1038/s41593-025-01905-6.

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, Inga Zhuravleva, Adelyn Tu-Chan, Karunesh Ganguly, Gopala K. Anumanchipalli, and Edward F. Chang. A high-performance neuroprosthesis for speech decoding and avatar control. *Nature*, 620(7976):1037–1046, 2023. doi: 10.1038/s41586-023-06443-4. URL https://doi.org/10.1038/s41586-023-06443-4.

Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. Layer-wise analysis of a self-supervised speech representation model. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 914–921. IEEE, 2021.

Xianhan Tan, Qi Lian, Junming Zhu, Jianmin Zhang, Yueming Wang, and Yu Qi. Effective phoneme decoding with hyperbolic neural networks for high-performance speech BCIs. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 32:3432–3441, 2024. doi: 10.1109/TNSRE.2024.3457313. URL https://doi.org/10.1109/TNSRE.2024.3457313.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Maitreyee Wairagkar, Nicholas S. Card, Tyler Singer-Clark, Xianda Hou, Carrina Iacobacci, Lee M. Miller, Leigh R. Hochberg, David M. Brandman, and Sergey D. Stavisky. An instantaneous voice-synthesis neuroprosthesis. *Nature*, 644(8075):145–152, 2025. doi: 10.1038/s41586-025-09127-3. URL https://doi.org/10.1038/s41586-025-09127-3.

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, Krishna V. Shenoy, and Jaimie M. Henderson. A high-performance speech neuroprosthesis. *Nature*, 620 (7976):1031–1036, 2023. doi: 10.1038/s41586-023-06377-x. URL https://doi.org/10.1038/s41586-023-06377-x.

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, 2026.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Appendix Organization. The appendix documents the protocols and evidence supporting the dual-level speech-prior transfer framework in Sections 3 and 4. Appendix A specifies the datasets, information available at each stage, evaluation metrics, and baseline comparisons. Appendix B details the neural and speech modules, representation and predictive transfer, training and inference algorithms, privileged-prediction diagnostics, and final text-decoding systems. Appendix C extends the benchmark results and the analyses in Figures 2 and 3, covering component ablations, speech teachers, prediction and posterior quality, representations and temporal correspondence, data efficiency, edit errors, qualitative cases, and computational cost. Appendix D reports model and optimization settings and hyperparameter sensitivity. Appendix E derives the length-aware alignment objective, posterior decomposition and gradient paths, and training and deployment complexity.

A Data and Evaluation Protocols

Our evaluation separates neural phoneme prediction, language-assisted text decoding, and offline diagnostic analyses. This distinction allows us to examine how training-time speech priors affect the neural decoder while identifying the additional contributions of language decoding and system fusion. We first define the datasets and preprocessing, then specify the metrics and statistical units, and finally describe the comparison protocols.

A.1 Datasets and Neural Preprocessing

Brain-to-Text Benchmark '24. Our primary benchmark uses the intracortical attempted-speech recordings associated with Willett et al. (2023). Each neural time step contains 256 input features. We train on 8,800 sentences and evaluate local neural decoding on a fixed validation set, comprising 40 sentences from each of 15 recording days. The validation references contain 15,997 task labels, including 12,312 occurrences of the 39 non-silence phonemes. These full reference counts define the denominators used in sequence and phoneme-level analyses; selecting a subset of phonemes for display does not redefine the evaluation population. Text-decoding performance is additionally evaluated on the benchmark hidden test set.

Brain-to-Text '25. The second dataset uses the T15 copy-task recordings associated with Card et al. (2024) and the expanded Brain-to-Text '25 release. Recordings span August 11, 2023 to April 13, 2025 and use four 64-electrode arrays. Threshold crossings and spike-band power from 256 channels form a 512-dimensional feature vector in each 20-ms bin. The expanded protocol contains 9,336 training sentences from 45 recording days, 162 validation sentences from 41 days, and 1,450 test sentences from 41 days, for a total of 10,948 sentences. Features are z-scored using statistics from the first 20 trials of each session. The original released GRU baseline follows its original data split; the local Kord (#10) reproduction and MOST-OPSD use the common expanded training split, as specified in Appendix B.

Both datasets are evaluated within participant, rather than by transferring a trained decoder between participants. The data information is shown in Table 2.

Table 2: Dataset details. Sentence counts specify the protocol.

Dataset	Task	Neural input	Training / Validation	Hidden test
Brain-to-Text '24	Attempted speech	256 features	8,800 / 600	1,200
Brain-to-Text '25	Attempted speech	512 features	9,336 / 162	1,450

Training-time Speech Information. We synthesize speech from training transcripts using Piper (Hansen et al., 2023) and extract hidden representations with a frozen speech encoder. The main Brain-to-Text '24 configuration uses the HFC female voice, and Brain-to-Text '25 uses the HFC male voice. Wav2Vec 2.0 Base (Baevski et al., 2020) supplies the default multi-layer speech features. Training reference phonemes supervise the CTC objectives; transcript-generated speech supervises MOST module and conditions the privileged branch used by OPSD module. Validation PER and CTC loss are computed from the neural-only branch. At deployment, the decoder receives neural features and the applicable recording-day transformation, without the current transcript or its

synthesized speech. Offline reference-conditioned diagnostics are a separate analysis path: they may use reference alignments and frozen speech features to inspect fixed models, but these quantities are not inputs to the reported neural-only predictions.

Neural Preprocessing. The MOST-OPSD neural path combines Gaussian smoothing, recording-day affine transformations, Softsign nonlinearities, and temporal windowing before recurrent encoding (Willett et al., 2023). For Brain-to-Text '24, padded input positions are cleared before smoothing and again afterward; valid lengths are retained through windowing and packed recurrent computation. Brain-to-Text '25 additionally uses a Gaussian smoothing kernel of 100 bins and random-walk augmentation with standard deviation 0.1 (Card et al., 2024). The ordering of model operations and full optimization settings are provided in Appendices B.1 and D.1.

A.2 Evaluation Protocols and Metrics

Greedy Phoneme Decoding. The CTC output vocabulary contains 39 English phonemes, SIL, and the blank symbol, giving 41 classes with blank index zero. Greedy decoding takes the framewise maximum-probability label, merges consecutive repetitions, and removes blanks. For an evaluation set $\mathcal{I}$, let S_i, D_i, and I_i denote substitution, deletion, and insertion counts after edit alignment, and let $r_i = |y_i|$ be the reference length. We compute:

$$\mathrm{PER}(\mathcal{I}) = 100 * \frac{\sum_{i\in\mathcal{I}}(S_i + D_i + I_i)}{\sum_{i\in\mathcal{I}} r_i}. \tag{16}$$

Overall PER therefore pools edit counts and reference counts, rather than averaging sentence percentages. Day-wise PER uses the same calculation restricted to each recording day. WER applies the corresponding edit-distance calculation to word sequences and is reported separately from phoneme decoding.

Checkpoint Selection. For MOST-OPSD, validation PER selects the neural checkpoint, with validation CTC used as a secondary criterion in a tie. Day-wise analyses reuse one selected checkpoint per trained model across all evaluated days; they do not select a different checkpoint for each day. *CTC loss at the selected model* evaluates the checkpoint chosen by PER.

Text Decoding and System-level Evaluation. Single-acoustic-model evaluation passes one neural model's outputs to a 5-gram weighted finite-state transducer and OPT-6.7B rescoring (Willett et al., 2023). Fusion results additionally include candidate combination and the corresponding final text-selection procedure. We report these as separate endpoints because improvements in a complete text system need not arise solely from its acoustic model. For two datasets, validation and single-model results use mean and sample standard deviation, whereas fusion test results report the best submitted Hidden or Private WER for each system. Fusion results were obtained from multiple submissions; a best submitted system score is therefore distinguished from repeated validation performance. Appendix B.5 specifies the language models and fusion procedures.

Sequence Probability and Emitted-token Correctness. We complement edit rates with the reference-sequence CTC log score and an emitted-token Brier score (Gneiting & Raftery, 2007). The per-sentence CTC score is:

$$s_i^{\mathrm{CTC}} = -\frac{\log_2 P(y_i \mid X_i)}{r_i}, \tag{17}$$

measured in bits per reference phoneme. For emitted token j, let c_j denote its greedy-CTC emission-segment confidence and let $a_j \in \{0, 1\}$ indicate correctness under sequence edit alignment. For M emitted tokens, the Brier score is:

$$\mathrm{Brier}_{\mathrm{emit}} = \frac{1}{M}\sum_{j=1}^{M}(c_j - a_j)^2. \tag{18}$$

This is a binary score for the correctness of an emitted token, not a multiclass framewise Brier score. Deleted reference tokens are not emitted tokens and are instead captured by sequence-level metrics. The complementary measures thus evaluate reference-sequence probability and the reliability of actual emissions.

Reliability, Calibration, and Selective Risk. Reliability diagrams use shared fixed confidence bins and plot mean confidence against empirical correctness; cells containing fewer than three emitted tokens are left empty. ECE instead uses ten equal-frequency bins within each model and recording day. For bin $\mathcal{B}_b$, define its mean confidence and accuracy as $\mathrm{conf}(\mathcal{B}_b)$ and $\mathrm{acc}(\mathcal{B}_b)$, respectively. The model/day ECE is:

$$\mathrm{ECE} = \sum_{b=1}^{10} \frac{|\mathcal{B}_b|}{M} \left|\mathrm{acc}(\mathcal{B}_b) - \mathrm{conf}(\mathcal{B}_b)\right|. \tag{19}$$

ECE is multiplied by 100 when displayed as a percentage. Posterior-quality summaries aggregate model observations within recording days and then weight the 15 observed days equally. For selective risk, sentences are ranked by their mean emission-segment confidence, and retained-set PER is evaluated as coverage increases. AURC and E-AURC are reported over the displayed 10%-100% coverage interval. Their archived values and the associated curves appear in Appendix C.5.

Sequence Confusion and Reference-Conditioned Classification. Greedy sequence confusion aligns complete predicted and reference sequences before extracting displayed phoneme classes. All displayed classes retain their full reference denominators, including errors involving classes outside the displayed subset. Reference-conditioned token classification is a separate diagnostic: logits are averaged over the nonblank states assigned to each reference occurrence by CTC alignment, followed by an argmax over the full 41-class vocabulary. The complete classification display includes the 40 nonblank labels, with blank and unaligned mass accounted for separately. Reference-conditioned classification therefore measures label information at aligned occurrences and is not a replacement for unconstrained greedy PER. The 39-phone plots retain all 12,312 non-SIL reference occurrences.

Representation and Temporal Analyses. For PCA, token representations are L2-normalized and a 50-dimensional, non-whitened basis is fitted to the training token bank; the displayed validation points use the first two components of this fixed transform (Abdi & Williams, 2010). The t-SNE visualizations use the validation PCA50 coordinates, perplexity 30, PCA initialization, automatic learning rate, and 1,000 iterations. Each model or stage retains its own coordinate system, so the plots describe within-panel class geometry. Temporal CKA (Kornblith et al., 2019) uses 32 valid temporal positions per sentence and is summarized by recording day, with 1,000 within-sentence temporal permutations forming the reference distribution. Temporal-matching diagnostics hold the neural sentence, projection, and model fixed while comparing the associated speech, a different transcript's speech, and time-block-shuffled speech; matching costs are compared within each neural–speech stage. The eight-phone subset in Figure 3 was selected after analysis to illustrate recognition and representation changes, while the full-class analyses preserve complete occurrence support.

Statistical Units. Component-ablation boxes show the median and interquartile range (IQR), with whiskers extending to observations within 1.5 IQR of the quartiles; repeated reference boxes reuse the same observations. Their significance annotations use the archived two-sided Welch tests and p-values. Other comparison families retain their documented paired test. Gain and posterior intervals are descriptive 95% bootstrap intervals over the 15 observed recording days, after model combinations or repeated-model observations have been aggregated within days. They describe variation over those observed days, rather than uncertainty over a new participant population. Difficulty quartiles are defined separately from each baseline's sentence-level PER, with fixed sample identifiers resolving ties. Hyperparameter bands describe IQR across configuration summaries at 25,000 updates, not model-level standard deviations or confidence intervals for the final model.

A.3 Baseline Methods

Brain-to-Text '24 Baselines. We evaluate two neural baselines with publicly available implementations on the common validation set. The GRU baseline follows the recurrent phoneme-decoding approach of Willett et al. (2023) and uses five unidirectional recurrent layers with 1,024 hidden units. TMT follows Feghhi et al. (2026); it uses causal time-masked self-attention, retains its native log-feature input, and has five layers, a hidden dimension of 384, and six attention heads. We reproduce both baselines locally from their public code and compute their validation neural metrics from the resulting predictions. These are native architectures. MOST-OPSD and its components are

compared under the controlled configurations in Appendix C.2. For the two additional methods in the main-text comparison, DCoND-LIFT (Li et al., 2025) and BIT (Zhang et al., 2026), the reported entries come from the corresponding official reports and benchmark submissions.

Brain-to-Text '25 Baselines. We compare MOST-OPSD with the original released GRU baseline and Kord's five-layer GRU system, which ranks 10th in the benchmark. Both baseline results are obtained locally: we evaluate the GRU baseline under its original data split and reproduce Kord under the expanded split shared with MOST-OPSD. For the single-model comparison, Kord uses 5-gram decoding followed by OPT-6.7B rescoring. The validation and single-model baseline results are reported from these local evaluations, while fusion scores are system-level results taken from benchmark submissions. This separation keeps local baseline measurements distinct from benchmark-level fusion performance and preserves the protocol difference between the original and expanded data splits.

B Implementation Details

This section specifies how training-time speech priors enter the neural decoder and which computations remain at deployment. We first describe the neural and speech interfaces, then implement MOST and OPSD on a shared neural trajectory, and finally provide the optimization procedures, local privileged-prediction diagnostics, and language-assisted decoding systems. We use the data and metric definitions in Appendix A; the alignment, distillation, and complexity derivations are collected in Appendix E.

B.1 Neural Decoder and Speech Teachers

Neural Backbone and Intermediate States. The performance configuration for Brain-to-Text '24 uses a six-layer bidirectional GRU with $h = 1024$ hidden units per direction, giving a concatenated state dimension $2h = 2048$. Brain-to-Text '25 uses six bidirectional layers with $h = 768$ and state dimension 1536. Each model includes recording-day affine input transformations, a Softsign nonlinearity, temporal windowing, and a shared linear phoneme head $g_\psi : \mathbb{R}^{2h} \to \mathbb{R}^{41}$. The output classes are 39 phonemes, SIL, and the CTC blank, with blank index zero. We extract G2, G4, and G6 from the same forward pass, where Gl denotes the state sequence after recurrent layer l; H_i^{neu} denotes the final state $H_i^{(6)}$.

Preprocessing and Length Propagation. For Brain-to-Text '24, the input has $C = 256$ features, window width $k_{\mathrm{win}} = 32$, and stride $s = 4$. Padded positions are cleared before Gaussian smoothing and again afterward; training-time masking and input dropout precede the recording-day affine transformation, Softsign, window unfolding, and packed recurrent computation. White-noise and constant-offset augmentation follow the settings in Appendix A.1. For T15, $C = 512$, $k_{\mathrm{win}} = 14$, and $s = 4$; the model has 45 recording-day affine transformations and learnable initial recurrent states for each layer and direction. Its preprocessing additionally includes the documented random-walk augmentation and a Gaussian kernel of 100 bins with standard deviation 2 bins. For either dataset, the valid windowed length is $n_i = \lfloor (T_i - k_{\mathrm{win}})/s \rfloor + 1$ for $T_i \geq k_{\mathrm{win}}$, and all subsequent losses and masks use this length rather than the padded batch extent. Appendix E derives the length calculation and its role in excluding padded alignment positions.

Speech Features. We synthesize training transcripts with Piper (Hansen et al., 2023), using `en_US-hfc_female-medium` for Brain-to-Text '24 and `en_US-hfc_male-medium` for Brain-to-Text '25. The default teacher is frozen Wav2Vec 2.0 Base (Baevski et al., 2020); W6, W8, and W10 denote its sixth, eighth, and tenth Transformer-layer representations, each with dimension 768. These are three views of one speech encoder rather than three separately trained encoders. Features are detached when loaded, and their true lengths accompany the cached tensors. No gradient from neural training updates the speech encoder or TTS model.

B.2 Multi-Stage Representation Transfer

Independent Neural Projections. MOST pairs G2 with W6, G4 with W8, and G6 with W10. For each pairing, an independent trainable linear map P_k projects the neural states to the 768-dimensional speech space, as in Equation 3. The projections are initialized by copying the same deep-stage linear projection and then updated independently. Each projected trajectory receives its own speech-structured objective, while gradients from the three objectives accumulate in the shared recurrent encoder.

Alignment and Normalization. For each sentence and view, we construct squared Euclidean distances between valid projected neural states and valid speech features and evaluate the Soft-DTW recurrence (Cuturi & Blondel, 2017) in Equations 4–5. The recurrence uses $\gamma = 0.2$ and returns the sample-specific endpoint $d_i^{(k)} = A^{(i,k)}_{n_i, m_i^{(k)}}$; padded rows and columns do not contribute to either the forward cost or its gradient. The implementation then computes:

$$\begin{aligned} \widetilde{d}_i^{(k)} &= \frac{d_i^{(k)}}{n_i + m_i^{(k)}} \frac{c_{\mathrm{ref}}}{c_k}, \\ \mathcal{L}_{\mathrm{MOST}} &= \frac{1}{B} \sum_{i=1}^{B} \sum_{k=1}^{3} w_k \widetilde{d}_i^{(k)}. \end{aligned} \tag{20}$$

This is the length and scale normalization of Equations 6-7, applied before the minibatch reduction. The denominator is the sum of valid sequence lengths, not the length of a selected hard alignment path. The stage coefficients are used at their stated values without division by their sum. The monotonic-path formulation, gradients, and effect of fixed scale factors are derived in Appendix E.

B.3 Privileged Predictive Transfer

Privileged Speech Teacher Construction. The privileged adapter uses the final neural states as queries and detached deep speech features as keys and values in multi-head cross-attention (Vaswani et al., 2017). The attention latent dimension is 128, with four heads and an attention dropout of 0.1. Its output is mapped back to $2h$ dimensions and added to $H_i^{(6)}$, implementing the residual and privileged state in Equations 8–9. Since the number of queries is n_i, privileged and neural-only logits have the same frame count without interpolating the speech features onto the neural grid. Speech padding is excluded by the key-padding mask, and neural padding is cleared in the residual and outputs.

Shared Decision Space. The adapter output projection has zero-initialized weights and bias, so the initial residual is zero, and both branches initially present the same neural states to the same linear head. The privileged branch is therefore initialized as the neural-only predictor rather than as an independently initialized phoneme classifier. Its CTC loss uses the original, undetached privileged logits and updates the adapter, neural encoder, and shared head. Both branches receive the same reference phoneme sequence, with dataset-specific CTC reduction, as shown in Appendix A.2.

Detached-Target Forward KL. At each active OPSD update, we form temperature-scaled distributions from the current two branches:

$$\begin{aligned} p_{it}^{\tau} &= \mathrm{softmax}(\ell_{it}^{\mathrm{neu}}/\tau), \\ q_{it}^{\tau} &= \mathrm{softmax}(\mathrm{sg}(\ell_{it}^{\mathrm{priv}})/\tau), \\ \mathcal{L}_{\mathrm{KD}} &= \frac{\tau^2}{\sum_{i=1}^{B} n_i} \sum_{i=1}^{B} \sum_{t=1}^{n_i} D_{\mathrm{KL}}(q_{it}^{\tau} \,\|\, p_{it}^{\tau}). \end{aligned} \tag{21}$$

We use $\tau = 2$ and the full 41-class vocabulary, including blank, with the temperature compensation in Equation 12 (Hinton et al., 2015). The implementation evaluates log probabilities with `log_softmax`, sums class contributions, masks padded neural frames, and divides by the total number of valid frames in the minibatch. Thus, the distillation reduction is a frame-weighted mean.

The nonblank scale is 1: the blank/nonblank mass term and conditional nonblank term are not independently reweighted. Their exact decomposition and the temperature-scaled logit gradient are given in Appendix E.

The speech features remain fixed, but the privileged posterior is recomputed from the current neural states, adapter, and shared head. Stop-gradient holds that target constant only within the current backward pass; KL gradients flow through the student distribution to the neural encoder and shared head. Privileged CTC remains differentiable through the residual branch and supplies the adapter's task supervision. In this implementation, "on-policy" refers to the current-model-conditioned predictive target; it does not introduce an autoregressive rollout into CTC training.

B.4 Training and Inference Algorithms

Optimization Process. Both datasets begin with 20,000 updates of neural CTC and MOST, then activate privileged CTC and KL while continuing the same optimization process. Let u count completed updates and let $u_0 = 20{,}000$. The implemented schedule is:

$$(\beta_u, \mu_u) = \begin{cases} (0, 0), & u < u_0, \\ (0.25, 0.005), & u \geq u_0, \end{cases}$$
$$\mathcal{L}_u = \frac{\mathcal{L}_{\mathrm{CTC}}^{\mathrm{neu}} + \beta_u \mathcal{L}_{\mathrm{CTC}}^{\mathrm{priv}}}{1 + \beta_u} + \lambda \mathcal{L}_{\mathrm{MOST}} + \mu_u \mathcal{L}_{\mathrm{KD}}. \tag{22}$$

The privileged branch and its losses are omitted before activation. After activation, the objective is exactly Equation 14; the CTC normalization averages the two supervised terms by $1 + \beta$. This schedule is an implementation choice for the same two-module framework.

Dataset-Specific Optimization. For Brain-to-Text '24, Adam uses $(0.9, 0.999)$ momentum coefficients, epsilon 0.1, weight decay 10^{-5}, and gradient clipping norm 2 (Willett et al., 2023). The learning rate changes linearly from 0.03 at update zero to 0.003 at 40,000 updates, then linearly to 0.0003 at 100,000 updates. Validation starts at 20,000 updates and is repeated every 500 updates. Brain-to-Text '25 uses AdamW, a 1000-update learning-rate warm-up, and cosine decay with maximum learning rate 0.004, minimum learning rate 8×10^{-5}, and 120,000 decay steps for both the neural and recording-day parameter groups (Card et al., 2024). Its backbone weight decay is 10^{-4} and recording-day weight decay is zero; batches contain 64 sentences sampled in balance from four recording days, the clipping norm is 10, and validation runs every 2000 updates. For both datasets, neural-only validation PER selects the saved model, with CTC breaking ties.

The auxiliary transfer modules are used only during training (Algorithm 1), while inference retains the neural encoder and shared head (Algorithm 2).

B.5 Final Decoding Systems

Neural Validation. We evaluate validation PER and CTC loss using the neural-only branch with training perturbations and dropout disabled. The neural encoder and shared phoneme head produce logits over the CTC alphabet at valid neural frames. Greedy decoding takes the framewise argmax and applies CTC collapse to obtain the phoneme sequence used for PER. CTC loss is computed from the untempered logits and reference phonemes using the dataset-specific reduction. Checkpoints are selected by validation PER, with CTC loss as the tie-breaker.

Single-Acoustic-Model Text Decoding. Hidden-test WER is obtained by passing each independently trained acoustic model's neural outputs through a 5-gram WFST decoder and OPT-6.7B rescoring, following Willett et al. (2023). The WFST generates candidate word sequences from neural phoneme scores, and OPT-6.7B incorporates sentence context to select the final transcription. For MOST-OPSD on Brain-to-Text '25, decoding uses an acoustic scale of 0.45, a blank penalty of 90, and 100 hypotheses, with OPT rescoring weight $\alpha = 0.55$ and zero length penalty. Each acoustic model produces its own test submission, and single-model WER is summarized as mean $\pm$ sample SD across independent trainings. The decoding inputs comprise neural predictions and the language models.

Algorithm 1 Training MOST-OPSD

Require: Neural sequences X_i, day IDs r_i, phonemes y_i, frozen speech views $Z_i^{(k)}$, valid lengths, update budget U, and OPSD step u_0.
Ensure: Validation-selected checkpoint.

Initialize the neural encoder, day transforms, shared head, projections, adapter, and optimizer.
Zero-initialize the adapter output layer and freeze the estimated scales c_k, c_{ref}.
for $u = 1, \ldots, U$ **do**
    Sample and preprocess a minibatch; compute valid lengths n_i.
    Encode once to obtain $H_i^{(2)}, H_i^{(4)}, H_i^{(6)}$ and compute: $\ell_i^{\mathrm{neu}} = g_\psi(H_i^{(6)})$, $\mathcal{L}_{\mathrm{CTC}}^{\mathrm{neu}} = \mathrm{CTC}(\ell_i^{\mathrm{neu}}, y_i)$.
    For each stage-view pair, compute the masked Soft-DTW endpoint $d_i^{(k)}$ using Equation 4, and set: $\widetilde{d}_i^{(k)} = \frac{d_i^{(k)}}{n_i + m_i^{(k)}} \frac{c_{\mathrm{ref}}}{c_k}$, $\mathcal{L}_{\mathrm{MOST}} = \frac{1}{B} \sum_i \sum_k w_k \widetilde{d}_i^{(k)}$.
    if $u \geq u_0$ **then**
        Build the privileged state using detached W10:

$$H_i^{\mathrm{priv}} = H_i^{(6)} + W_O \Big[\mathrm{MHA}\Big(W_Q(H_i^{(6)}), W_K(\mathrm{sg}(Z_i^{\mathrm{W10}})), W_V(\mathrm{sg}(Z_i^{\mathrm{W10}})) \Big) \Big].$$

        Compute: $\ell_i^{\mathrm{priv}} = g_\psi(H_i^{\mathrm{priv}})$ and $\mathcal{L}_{\mathrm{CTC}}^{\mathrm{priv}}$.
        Compute: $p_{it}^\tau = \mathrm{softmax}(\ell_{it}^{\mathrm{neu}}/\tau)$, $q_{it}^\tau = \mathrm{softmax}(\mathrm{sg}(\ell_{it}^{\mathrm{priv}})/\tau)$,

$$\mathcal{L}_{\mathrm{KD}} = \frac{\tau^2}{\sum_i n_i} \sum_{i,t,v} q_{it}^\tau(v) \log \frac{q_{it}^\tau(v)}{p_{it}^\tau(v)}.$$

    else
        Set: $\mathcal{L}_{\mathrm{CTC}}^{\mathrm{priv}} = \mathcal{L}_{\mathrm{KD}} = 0$.
    end if
    Form:

$$\mathcal{L} = \frac{\mathcal{L}_{\mathrm{CTC}}^{\mathrm{neu}} + \beta \mathcal{L}_{\mathrm{CTC}}^{\mathrm{priv}}}{1 + \beta} + \lambda \mathcal{L}_{\mathrm{MOST}} + \mu \mathcal{L}_{\mathrm{KD}}.$$

    Backpropagate, clip gradients, and update the optimizer and schedule.
    if validation interval **then**
        Evaluate the neural-only branch and save the checkpoint with the lowest validation PER, using CTC loss as a tie-breaker.
    end if
end for
return the selected checkpoint.

Algorithm 2 Neural-Only Inference

Require: Neural sequence X_i, valid length n_i, day ID r_i, and selected checkpoint.
Ensure: Neural logits and decoded phonemes or text.

Load the neural encoder, day transform, and shared head in evaluation mode.
Remove the speech encoder, TTS model, MOST projections, and privileged adapter.
Apply inference preprocessing and compute: $\ell_i^{\mathrm{neu}} = g_\psi(\mathrm{Encoder}(X_i, r_i)) \in \mathbb{R}^{n_i \times 41}$.
Decode phonemes using: $\widehat{y}_i = \mathcal{B}\left(\left[\underset{v \in \mathcal{V}_{\mathrm{CTC}}}{\arg\max}\ \mathrm{softmax}(\ell_{it}^{\mathrm{neu}})(v) \right]_{t=1}^{n_i} \right)$.
Optionally pass ℓ_i^{neu} to the language-decoding pipeline in Appendix B.5.
return ℓ_i^{neu}, $\widehat{y}_i$, or decoded text.

LLaMA-Based Hypothesis Fusion. To combine complementary predictions across independently trained models, we collect ten candidate transcriptions for each sentence from the single-acoustic-model pipeline above. These candidates are supplied jointly to the LLaMA fusion model, which produces one consolidated transcription using their shared content and complementary phonetic evidence (Feghhi et al., 2026). The implementation uses Llama-3.1-8B-Instruct with 4-bit

weights and a fusion adapter, with deterministic generation, one beam, a maximum input sequence length of 768, and at most 400 generated tokens. The resulting transcription is the LLaMA fusion output, which combines text hypotheses while retaining the neural-only acoustic prediction path.

C Additional Experimental Results

This section presents additional results on the second dataset, followed by analyses of the primary dataset, including component contributions, speech teachers, prediction quality, representation learning, temporal correspondence, data efficiency, error structures, qualitative examples, and computational efficiency. Complete numerical results and implementation details accompany the corresponding analyses.

C.1 Additional Benchmark Results

To evaluate the applicability of MOST-OPSD across neural speech datasets, we conduct additional experiments on Brain-to-Text '25 [2], based on the recordings of Card et al. (2024). The expanded protocol comprises 9,336 training sentences from 45 recording days, 162 validation sentences from 41 days, and 1,450 test sentences from 41 days. We compare MOST-OPSD against the GRU baseline and a locally reproduced Kord model, which ranked 10th in the benchmark submission, with Kord and MOST-OPSD trained on the same expanded split (Table 3). MOST-OPSD employs a six-layer bidirectional GRU with 768 hidden units per direction and learns from Piper HFC male speech using frozen Wav2Vec 2.0 Base features. The implementation follows Appendix B, and the evaluation covers neural phoneme decoding, single-model text decoding, and complete-system fusion.

Table 3: Neural and text decoding performance on the *Brain-to-Text '25*. PER and WER are percentages; lower is better. Validation and single-model results report mean ± sample SD over five independent trainings; fusion results report benchmark Private WER. GRU uses the original split, while MOST-OPSD shares the expanded training split with Kord (#10). **Bold** marks the best result.

Method	Val. PER	Val. CTC loss	Private WER (single model)	Private WER (fusion system)
GRU (#baseline)	10.14 ± 0.09	22.40 ± 0.36	6.91 ± 0.15	–
Kord (#10)	8.40 ± 0.29	17.97 ± 0.61	5.03 ± 0.14	3.61
MOST-OPSD	$\mathbf{6.17 \pm 0.33}$	$\mathbf{10.79 \pm 0.34}$	$\mathbf{3.94 \pm 0.08}$	**2.46**

Neural Prediction and Single-Model Text Decoding. As shown in Table 3, MOST-OPSD achieves a validation PER of $6.17 \pm 0.33\%$, compared with $8.40 \pm 0.29\%$ for Kord (#10), an absolute reduction of 2.23 percentage points (26.5% relative). Validation CTC loss decreases from 17.97 ± 0.61 to 10.79 ± 0.34, indicating improved reference-sequence likelihood. These gains extend to single-model text decoding, where MOST-OPSD achieves a Private WER of $3.94 \pm 0.08\%$, compared with $5.03 \pm 0.14\%$ for Kord, corresponding to an absolute reduction of 1.09% and a relative reduction of 21.7%. The original GRU provides an additional reference, with $10.14 \pm 0.09\%$ validation PER and $6.91 \pm 0.15\%$ Private WER under its original data split.

Complete Fusion-System Results. The complete MOST-OPSD decoding system further reduces Private WER to 2.46%, compared to 3.61% for Kord, corresponding to an absolute reduction of 1.15% and a relative reduction of 31.9% (Table 3). The system employs Llama-3.1-8B-Instruct with 4-bit weights and the fusion adapter described in the Appendix B.5. Together, the neural, single-model, and fusion results show that improvements in phoneme prediction are accompanied by consistent gains in downstream text decoding in Brain-to-Text '25.

These results demonstrate that the lower greedy PER indicates improvements at the acoustic-model level, whereas the single-model and fusion results show corresponding gains in downstream text decoding. Together with the Brain-to-Text '24 results, these findings support the applicability of the proposed dual-level speech-prior transfer framework across different within-participant neural speech datasets.

[2] https://www.kaggle.com/competitions/brain-to-text-25/overview

C.2 Complete Ablation Study

This ablation study examines the contributions of representation transfer, predictive transfer, teacher selection, and temporal alignment to MOST-OPSD performance. The main PER comparisons are presented in Figure 2; here, we provide the corresponding validation CTC results and extend the analysis to alternative stage assignments and transfer-weighting strategies.

Complementarity of Representation and Prediction Transfer. We assess the contributions of MOST and OPSD using the four supervision configurations in Figure 2(a). Both transfer mechanisms improve phoneme prediction, and their combination achieves the lowest validation PER. The full method also achieves the lowest validation CTC loss of 0.457 ± 0.008 (Figure 5(a)), indicating improved reference-sequence likelihood. The advantage of joint supervision across both metrics supports the complementarity of representation-level speech supervision and privileged predictive supervision. MOST encourages the neural encoder to capture speech-related structure, while OPSD transfers phoneme-level predictive information to the neural branch through the shared decoding objective.

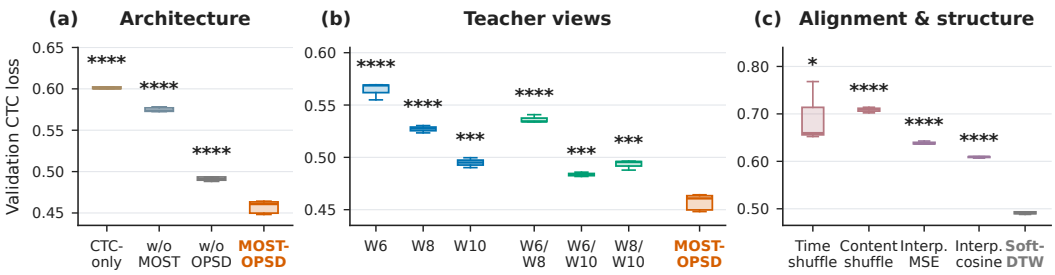

Figure 5: Supplementary ablation results for Figure 2. Validation CTC loss for **(a)** four supervision configurations, **(b)** three single- and two-teacher views, and **(c)** two speech-shuffling controls, interpolation with MSE or cosine loss compared with Soft-DTW. Boxes show medians and IQRs with 1.5-IQR whiskers; repeated references reuse the same observations. Significance symbols use two-sided t-tests against MOST-OPSD in (a) & (b) and Soft-DTW in (c).

Multiple Teacher Views. To examine the contribution of speech information at different abstraction levels, we compare individual W6, W8, and W10 views with their pairwise combinations, as shown in Figure 2(b). Among the single-view configurations, W8 achieves the lowest PER at 11.57%, compared with 13.37% for W6 and 11.90% for W10. Combining views yields PERs of 12.94%, 11.56%, and 11.77% for W6/W8, W6/W10, and W8/W10, respectively. The complete three-view configuration further reduces PER to 10.66%, with the corresponding CTC results shown in Figure 5(b). These results favor joint supervision from multiple speech layers: intermediate and deeper representations provide complementary information, and their combination yields a more effective training signal than any individual view or pair.

Temporal Alignment and Speech Controls. We next examine whether the benefits of speech supervision depend on the association between neural and speech sequences and on the flexibility of their temporal alignment. Figure 2(c) compares Soft-DTW with interpolated MSE and cosine objectives, together with speech-shuffling controls. Soft-DTW achieves $11.46 \pm 0.25\%$ PER, improving over interpolated MSE ($15.39 \pm 0.17\%$) and cosine loss ($15.08 \pm 0.15\%$). Time-shuffled and content-substituted speech yield substantially higher PERs of $19.24 \pm 2.16\%$ and $18.18 \pm 0.27\%$, respectively. The corresponding CTC comparisons in Figure 5(c) provide complementary evidence at the sequence-likelihood level. These results highlight two important properties of MOST: speech features must remain associated with the corresponding neural utterance, and flexible monotonic alignment is important for transferring information between sequences with different temporal dynamics. Appendix C.7 examines temporal correspondence in greater detail.

Extended Stage Assignments and Weights. We further examine whether the default G2/G4/G6–W6/W8/W10 assignment and fixed transfer weights can be improved by alternative stage pairings or learned weighting strategies. Table 4 reports the complete results. Among the alternative stage

assignments, G$(1,3,5)$–W$(2,6,10)$ obtains the lowest PER ($11.661 \pm 0.277\%$), whereas G$(2,4,6)$–W$(2,6,10)$ achieves the lowest CTC loss (0.493 ± 0.004). Randomizing teacher order increases PER to $11.911 \pm 0.078\%$, indicating that the correspondence between neural stages and speech representations influences the effectiveness of supervision.

Table 4: Additional ablations of stage-view assignment and transfer weighting. Ordered G/W tuples specify paired neural stages and speech layers; gated weights use the final neural stage, and bounded weights are constrained. Values are mean ± sample SD for validation PER (%) and minimum validation CTC loss. **Bold** marks the best result.

Configuration	Val. PER (%)	Val. CTC loss
Stage-view assignment		
Randomized teacher order	11.911 ± 0.078	0.522 ± 0.002
G$(2,4,6)$-W$(4,8,12)$	12.763 ± 0.259	0.496 ± 0.004
G$(2,4,6)$-W$(2,6,10)$	11.719 ± 0.126	0.493 ± 0.004
G$(1,3,5)$-W$(6,8,10)$	11.690 ± 0.111	0.518 ± 0.002
G$(1,3,5)$-W$(2,6,10)$	11.661 ± 0.277	0.513 ± 0.012
Transfer weighting		
Global learned weights, unbounded	14.982 ± 0.107	0.600 ± 0.002
Gated weights, unbounded	14.872 ± 0.199	0.596 ± 0.003
Global learned weights, bounded	11.596 ± 0.111	0.497 ± 0.004
Gated weights, bounded	11.646 ± 0.081	0.494 ± 0.003
MOST-OPSD	$\mathbf{10.657 \pm 0.101}$	$\mathbf{0.457 \pm 0.008}$

The weighting experiments reveal a stronger distinction between bounded and unbounded strategies. Unbounded global and gated weights produce PERs of $14.982 \pm 0.107\%$ and $14.872 \pm 0.199\%$, respectively. Constraining the weights substantially improves performance, reducing PER to $11.596 \pm 0.111\%$ with global weights and $11.646 \pm 0.081\%$ with gated weights. Their CTC losses show the same pattern, decreasing from approximately 0.60 to 0.49. The default MOST-OPSD configuration achieves the lowest PER ($10.657 \pm 0.101\%$) and CTC loss (0.457 ± 0.008) across the reported alternatives. Together with the stage-assignment results, these findings support the selected stage assignment and fixed weighting scheme, highlighting the importance of both the correspondence between representation levels and the relative strength of their supervision.

C.3 Speech Encoders and Synthetic Voices

This study evaluates the influence of pretrained speech encoders and synthetic voices on MOST-OPSD performance. The encoder comparison includes HuBERT (Hsu et al., 2021b), WavLM (Chen et al., 2022), and Wav2Vec 2.0 (Baevski et al., 2020), while the voice comparison evaluates Piper HFC Male and Female speech with Wav2Vec 2.0 fixed. Figure 6 presents validation PER and CTC loss for both experiments, allowing us to assess the effects of speech representation and synthesis choices on neural decoding.

Speech Encoders. Figures 6(a,b) compare the three pretrained speech encoders using validation PER and CTC loss. The observed differences across encoder configurations show that the choice of speech representation influences the training outcome of MOST-OPSD. Considering both metrics provides a complementary view of teacher quality, capturing changes in phoneme prediction accuracy and reference-sequence likelihood. These results motivate evaluating speech encoders as a distinct design choice within the transfer framework.

Synthetic Voices. Figures 6(c,d) examine the effect of synthetic voice selection with Wav2Vec 2.0 held fixed. The Male and Female conditions exhibit different validation performance distributions, indicating that the acoustic realization of the synthesized supervision can influence neural decoding even when the underlying pretrained encoder remains unchanged, although the difference is not significant ($p > 0.05$, paired t-tests, $n = 5$). The comparison shows that the selection of synthesized speech is a minor factor in constructing effective speech supervision.

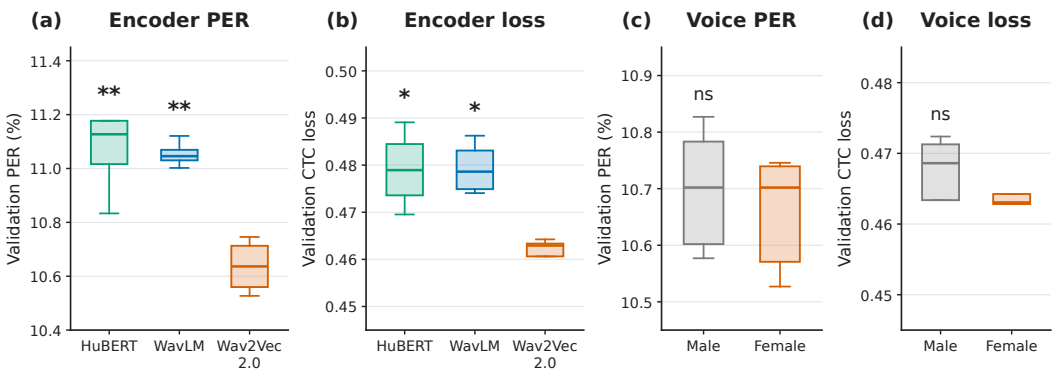

Figure 6: Speech encoders and synthetic voices with complete MOST-OPSD. **(a,b)** Val. PER and Val. CTC loss for HuBERT, WavLM, and Wav2Vec 2.0 with the *Piper HFC Female* voice; the recorded plotting protocol specifies five independent trainings per condition. **(c,d)** The same metrics for *Piper HFC Male* and *Female* speech with Wav2Vec 2.0 fixed, with five trainings per condition in the voice protocol. Boxes show medians and IQRs; marks indicate the two-sided paired t-tests.

Together, the encoder and voice experiments inform the selection of speech teachers and synthesis presets for MOST-OPSD, complementing the multi-view and temporal-alignment analyses in Appendix C.2.

C.4 Prediction Results

To characterize the prediction gains of MOST-OPSD, we evaluate performance across recording days and phoneme classes. The analysis combines day-wise PER (Figure 3(a)), CTC loss evaluated using the same checkpoints, and reference-conditioned phoneme classification on the full validation set.

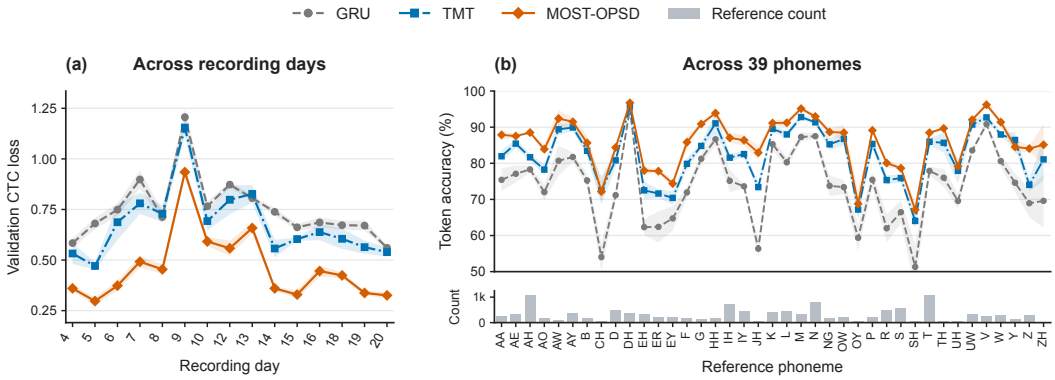

Figure 7: Validation loss across recording days and phoneme-wise classification. **(a)** Validation CTC loss on all 15 recording days, using the same frozen checkpoints as the existing day-wise PER comparison. Loss is the mean over sentences of natural-log CTC negative log-likelihood divided by reference length. **(b)** Frozen phoneme-wise classification for all 39 non-SIL reference phones. The aligned gray strip shows reference occurrence counts (12,312 total). Each run contributes one checkpoint across all days; daily minima are never selected independently. The shaded bands describe seed variability.

Greedy Decoding across Recording Days. Figure 3(a) shows that MOST-OPSD consistently improves greedy phoneme decoding across the 15 recording days. The corresponding CTC results in Figure 7(a) reveal a clearer distinction: TMT achieves only a modest loss reduction relative to GRU, whereas MOST-OPSD yields substantially lower CTC loss across recording days. This pattern may reflect improvements in TMT's greedy phoneme predictions that are accompanied by smaller gains

in reference-sequence likelihood. By combining representation alignment with predictive supervision, MOST-OPSD achieves improvements in both phoneme accuracy and reference-sequence likelihood across recording conditions.

Complete Phoneme Classification. We further examine how the prediction gains are distributed across phoneme classes using reference-conditioned classification. Figure 7(b) reports the classification accuracy of all 39 non-SIL phonemes, together with their occurrence counts. These class-wise results complement the sequence-level PER analysis by showing how reference-conditioned accuracy varies across phonemes with different reference frequencies. The aligned count strip contextualizes these results by distinguishing frequently represented phonemes from classes supported by fewer reference occurrences. From the figure, we can see that our method is better than or comparable to the baselines in terms of classification accuracy for almost all phonemes, and is slightly worse than TMT only for phoneme "Y", which indicates that our performance gain comes from a wide range of more accurate phoneme classifications.

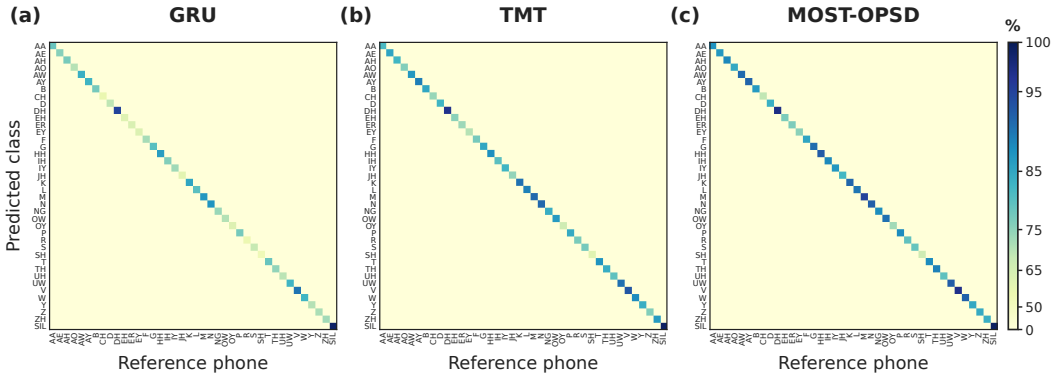

Figure 8: Full 40-class reference-conditioned token classification, including SIL. Rows are predicted classes and columns are reference phones. Token logits are averaged over reference-conditioned non-blank CTC states before a full-vocabulary argmax, and each column retains its complete reference denominator.

The complete confusion matrices in Figure 8 extend the class-wise analysis by identifying the predicted labels associated with each reference phoneme, including SIL. Their column-wise organization distinguishes correct classification from systematic substitutions and shows how errors are distributed across the phoneme inventory. Together, the day-wise and phoneme-wise analyses characterize the improvements of MOST-OPSD at two complementary levels: prediction quality across recording conditions and the structure of individual phoneme predictions.

C.5 Posterior Quality

To assess the quality of neural predictions beyond phoneme error rate, we examine reference-sequence probability, emitted-token confidence, selective risk, and calibration on the validation set. The evaluation covers 15 recording days, with 40 sentences per day, and uses frozen neural-only models. Metrics are computed within each model and day before aggregation, giving equal weight to each recording day. Let a_{kjd} denote the metric for model j of method k on day d, and let J_k denote the number of evaluated models for method k. The method-level summary is:

$$\overline{a}_k = \frac{1}{15} \sum_{d=1}^{15} \left(\frac{1}{J_k} \sum_{j=1}^{J_k} a_{kjd} \right). \tag{23}$$

Figure 9 presents the corresponding day-wise distributions and diagnostic curves, with numerical summaries and method comparisons reported in Tables 5-8.

Proper and Brier Scores. The sequence-level and token-level results in Figures 9(a,b) show improvements in the probabilities assigned by MOST-OPSD. Its reference-sequence CTC proper score

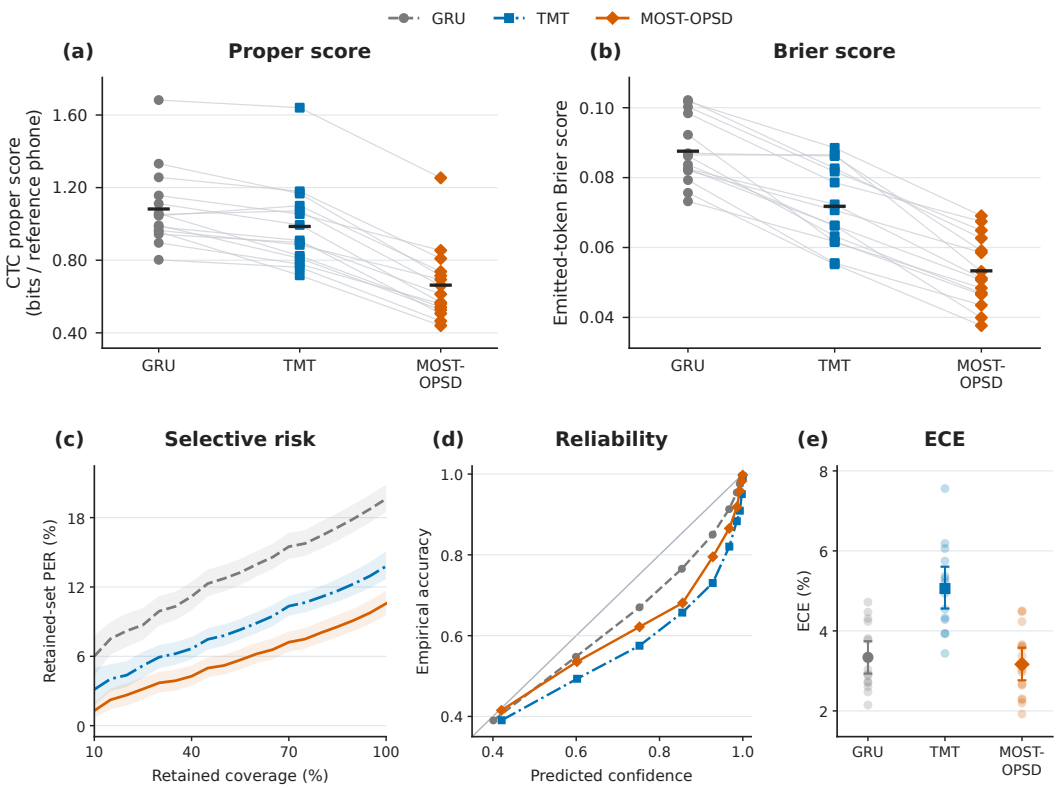

Figure 9: Posterior quality across 15 validation days. **(a,b)** Reference-sequence CTC proper score (bits/phoneme) and emitted-token Brier score; summaries and contrasts appear in Tables 5 and 6. **(c)** Confidence-ranked retained-set PER; AURC and E-AURC over 10%–100% coverage appear in Table 7. **(d)** Reliability diagrams using shared fixed bins. **(e)** ECE computed with ten equal-frequency bins within each model and day; summaries appear in Table 8. Metrics are computed per model/day before aggregation. Bands and error bars show descriptive 95% day-bootstrap intervals; Brier SDs in Figure 3(c) describe across-model variability.

reaches 0.662752 bits per reference phoneme, compared with 0.986043 for TMT and 1.082234 for GRU (Table 5). The corresponding emitted-token Brier score is 0.053323 ± 0.000841, lower than TMT's 0.071800 ± 0.002715 and GRU's 0.087565 ± 0.000888. The day-level contrasts in Table 6 are negative for both metrics against both baselines, with descriptive bootstrap intervals below zero. These findings indicate that MOST-OPSD assigns greater probability to reference sequences while producing more accurate confidence estimates for its emitted phonemes.

Table 5: Reference-sequence and emitted-token probability scores; lower is better. Metrics are computed within each model and recording day and summarized with equal day weights. The CTC proper score is reported in bits per reference phoneme. For Brier, values are the mean $\pm$ sample SD across trained models, computed after equal-day averaging within each model. These SDs also correspond to Figure 3(c); the intervals in Figure 9(a,b) instead describe day-bootstrap variability. **Bold** indicates the lowest mean in each column.

Method	CTC proper score	Emitted-token Brier
GRU	1.082234	0.087565 ± 0.000888
TMT	0.986043	0.071800 ± 0.002715
MOST-OPSD	$\mathbf{0.662752}$	$\mathbf{0.053323 \pm 0.000841}$

Selective Risk. Selective prediction further examines whether confidence identifies sentences with more reliable decoding. Sentences are ranked by mean emitted-segment confidence within each model and recording day, and retained-set PER is evaluated across the 10%–100% coverage range.

Table 6: Day-level probability-score contrasts corresponding to Figure 9(a,b). Differences are MOST-OPSD minus the indicated baseline, computed from within-day model averages and then averaged equally across the 15 recording days. Negative differences favor MOST-OPSD. Intervals are descriptive 95% day-bootstrap intervals obtained by retaining the within-day method comparisons during resampling. CTC proper-score differences are in bits per reference phoneme; Brier differences are dimensionless.

Metric	Baseline	Mean difference	95% day-bootstrap interval
CTC proper score	GRU	-0.419482	$[-0.474060, -0.369387]$
CTC proper score	TMT	-0.323291	$[-0.364094, -0.281160]$
Emitted-token Brier	GRU	-0.034242	$[-0.037013, -0.031346]$
Emitted-token Brier	TMT	-0.018477	$[-0.021778, -0.015505]$

MOST-OPSD achieves the lowest displayed risk-coverage curve in Figure 9(c), with an AURC of 0.052350, compared with 0.075908 for TMT and 0.118478 for GRU (Table 7). Its E-AURC is also the lowest at 0.014611. These results show that confidence-based sentence selection maintains lower decoding error with MOST-OPSD over the evaluated coverage range.

Table 7: Selective-risk area summaries for Figure 9(c). Sentences are ranked by mean emitted-segment confidence within each model and day; retained-set PER curves are then aggregated with equal day weights. AURC and E-AURC follow the archived numerical conventions over the displayed 10%–100% coverage range. Values are dimensionless point estimates. **Bold** indicates the lowest value in each column.

Method	AURC	E-AURC	Retained coverage
GRU	0.118478	0.017591	10%–100%
TMT	0.075908	0.016632	10%–100%
MOST-OPSD	**0.052350**	**0.014611**	10%–100%

Confidence Calibration. Calibration assesses the agreement between predicted confidence and empirical accuracy. The reliability diagrams in Figure 9(d) show high correctness for MOST-OPSD in the high-confidence region, alongside residual overconfidence at intermediate confidence levels. Using ten equal-frequency bins within each model and day, MOST-OPSD obtains an ECE of 3.16522%, compared with 3.33991% for GRU and 5.05825% for TMT (Table 8). TMT's lower proper and Brier scores relative to GRU are accompanied by a higher ECE, illustrating the distinct information captured by probability scoring and calibration.

Table 8: ECE summaries for Figure 9(e). ECE is computed using ten equal-frequency bins within each model and recording day, followed by within-day model averaging and equal-day aggregation. Means and descriptive 95% day-bootstrap interval endpoints are expressed in percent; lower is better. **Bold** indicates the lowest observed mean.

Method	Mean ECE (%)	95% day-bootstrap interval (%)
GRU	3.33991	$[2.93290, 3.74132]$
TMT	5.05825	$[4.55849, 5.60530]$
MOST-OPSD	**3.16522**	$[2.76451, 3.57935]$

Taken together, the sequence scores, emitted-token scores, selective-risk curves, and calibration results characterize the improvements of MOST-OPSD in posterior quality and confidence-guided prediction.

C.6 Representation Analyses

To better understand the representations learned through speech-prior supervision, this analysis examines neural-speech similarity and phoneme organization at the final and intermediate neural stages. CKA quantifies the association between neural and speech features, while PCA and t-SNE visualize phoneme-related structure using the same validation reference occurrences.

The CKA results in Figure 3(d) and Table 9 show that MOST-OPSD achieves an observed similarity of 0.181298, compared with 0.149517 for TMT and 0.092471 for GRU. To assess temporal correspondence, each method is also evaluated against a reference distribution generated from 1,000 within-sentence temporal permutations, using 32 valid positions per sentence. The observed-minus-permutation-mean gap is largest for MOST-OPSD (0.157812), followed by TMT (0.119938) and GRU (0.071636). These results indicate stronger temporal association between the neural features learned by MOST-OPSD and the analyzed speech representations.

Table 9: CKA summaries for Figure 3(d). The models are evaluated using 32 valid temporal positions per sentence and 1,000 within-sentence temporal permutations. The permutation mean and 95th percentile describe the reference distribution; the final column reports observed CKA minus the permutation mean. **Bold** indicates the largest observed CKA and corresponding descriptive gap.

Method	Observed	Permutation mean	Observed minus Permutation
GRU	0.092471	0.020836	0.071636
TMT	0.149517	0.029579	0.119938
MOST-OPSD	**0.181298**	0.023486	**0.157812**

Figure 10 extends the main-paper eight-phoneme visualization to all 39 non-SIL phonemes, comparing the final neural representations of GRU, TMT, and MOST-OPSD. The panels retain the same 12,312 reference occurrences and reveal phoneme-associated regions alongside substantial class overlap. PCA uses training-fitted projection bases, while t-SNE provides a complementary visualization of local neighborhoods in the validation representations. These full-inventory displays provide a broader view of phoneme organization underlying the decoding results.

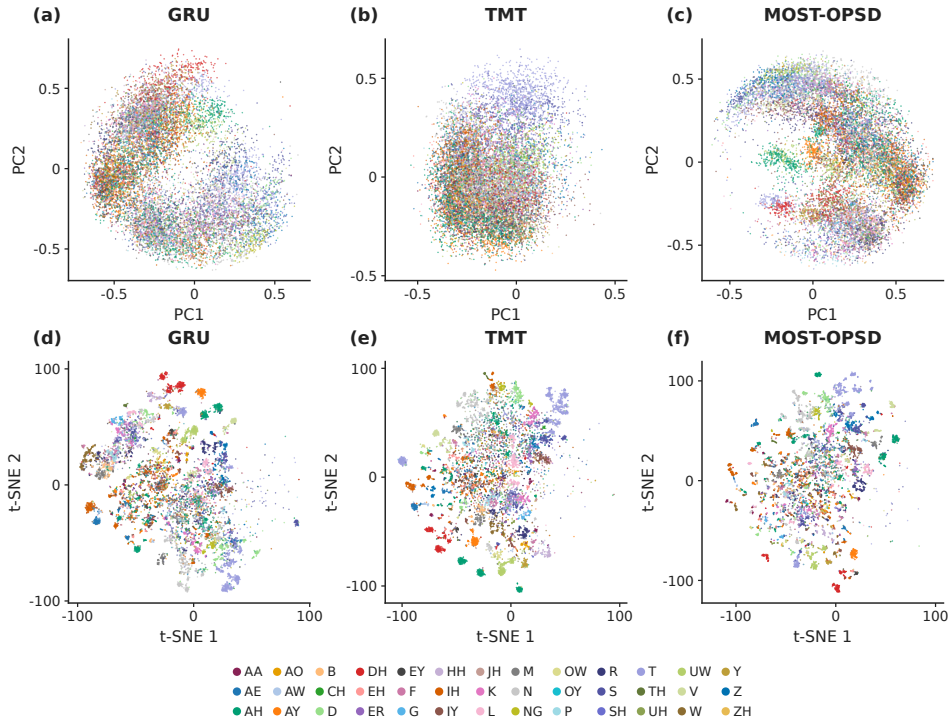

Figure 10: Full-inventory final neural representations. Columns compare GRU, TMT, and MOST-OPSD. **(a-c)** show training-fitted PCA, and **(d-f)** show t-SNE fitted to validation coordinates. Each panel retains the same reference occurrences across 39 non-SIL phones. Coordinate systems are model-specific, and the panels visualize within-model organization.

The intermediate representations in Figure 11 further illustrate how phoneme organization develops within MOST-OPSD. The densely mixed G2 representations become more spatially differentiated

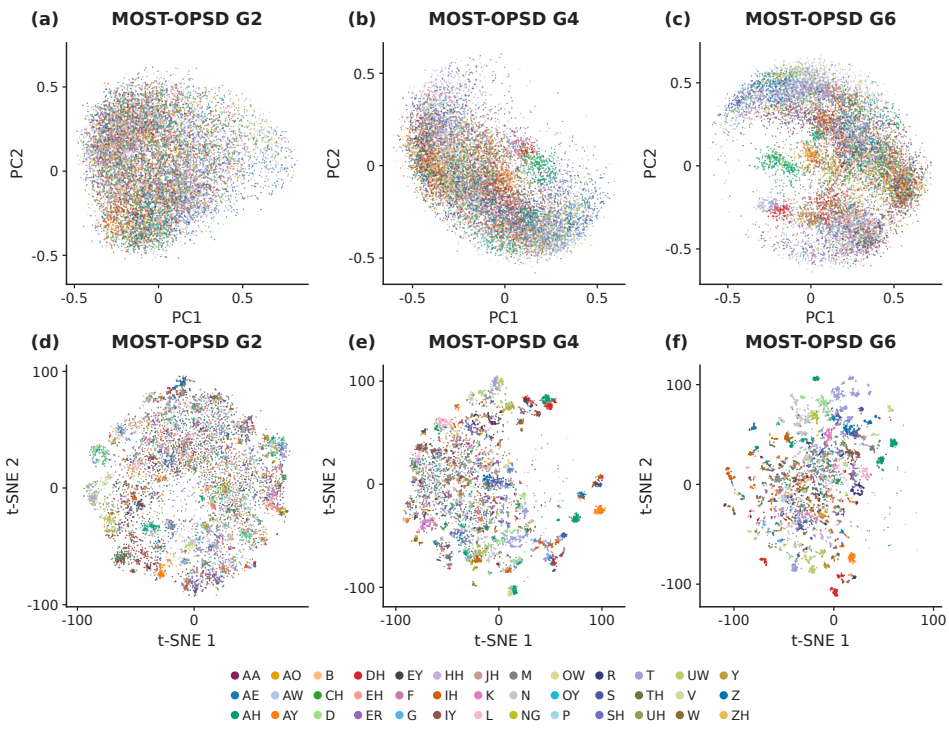

Figure 11: Representations at G2, G4, and G6 of the MOST-OPSD model. **(a-c)** show PC1/PC2 of stage-specific training-fitted PCA, and **(d-f)** show the corresponding validation t-SNE displays. All panels retain the same non-SIL reference occurrences and token assignments. Axes describe each layer's own coordinate system. The figure illustrates stage-dependent class organization.

at G4 and G6 in the PCA views, while the corresponding t-SNE panels reveal changes in local class organization. All stages use the same reference occurrences and token assignments, allowing the visualizations to trace changes within the trained model. Together with the CKA results, these observations suggest that multi-stage speech supervision is associated with structured neural representations and stronger neural-speech correspondence.

C.7 Temporal Correspondence

Does the learned neural-speech matching objective favor the associated speech over an alternative transcript or a reordering of the same speech features? We inspect one fixed sentence, holding the neural model and its learned projections fixed across all conditions. The stage pairs are G2-W6, G4-W8, and G6-W10; each comparison uses 63 valid neural positions and 83 valid speech-feature positions. Figure 12 provides the complete three-stage diagnostic. This is an offline intervention on the inputs to the matching calculation.

Associated and Different-Transcript Speech. For stage pair (g_k, w_k), let $U^{(k)} = P_k H^{(g_k)} \in \mathbb{R}^{63\times768}$ denote the fixed projected neural trajectory and let $Z^{(k)} \in \mathbb{R}^{83\times768}$ denote the speech features associated with its transcript. The correct condition uses $Z^{(k)}$ unchanged; the mismatch condition substitutes the corresponding different-transcript feature tensor, denoted $\overline{Z}^{(k)}$, while retaining $U^{(k)}$, P_k, and the scoring settings. Every supplied mismatch record has 83 valid feature rows at evaluation, matching the other conditions in that stage. This equality specifies the input shape used for scoring, not whether the donor originally had that length or underwent length adaptation; donor selection and any preceding adaptation must be recovered from the original diagnostic inputs rather than inferred from the final dimensions. The mismatch condition changes the speech prior, not the neural sentence or its reference phoneme sequence.

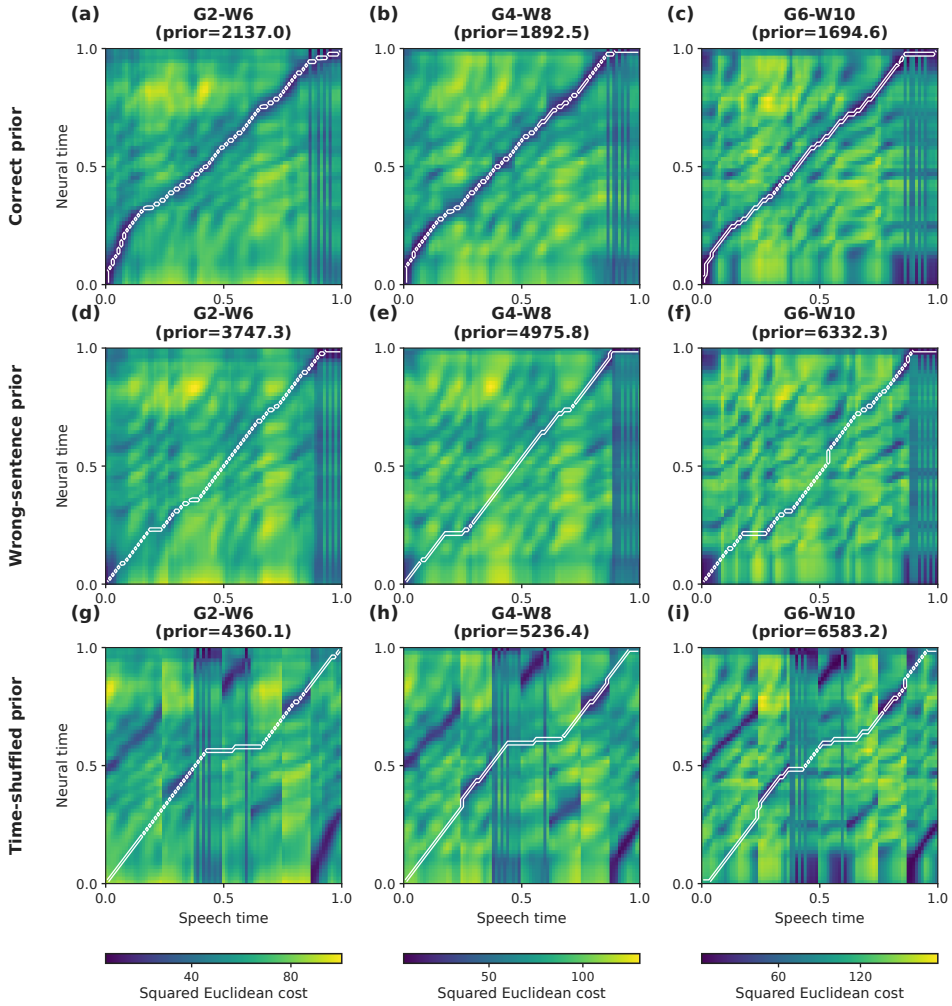

Figure 12: Temporal matching at three neural stages for one fixed sentence. Columns show G2-W6, G4-W8, and G6-W10; rows show correctly associated, different-transcript, and time-block-shuffled speech features. The model and learned projections are fixed; all panels use 63 neural and 83 speech positions, and the same eight-block permutation is used across stages. Heat maps show squared Euclidean distances with one shared linear scale per column. Titles report raw Soft-DTW endpoints at $\gamma = 0.2$, and white contours mark expected alignment occupancy 0.5. Axes show normalized native time for visualization only. Mathematical details are given in Appendix E.

Fixed Eight-Block Feature Permutation. The time-shuffled condition reorders the already encoded features of the correct prior, rather than substituting another transcript or shuffling and re-encoding the waveform. Write the block endpoints as $0 = b_0 < b_1 < \cdots < b_8 = 83$ and the fixed ordering of the eight blocks as $\pi = (\pi_1, \ldots, \pi_8)$, a permutation of $\{1, \ldots, 8\}$. The operation preserves the order and feature values within each block and concatenates the blocks in the prescribed order:

$$\begin{aligned} B_r^{(k)} &= Z^{(k)}_{b_{r-1}+1:b_r,:}, \qquad r = 1, \ldots, 8, \\ Z^{(k)}_{\text{shuffle}} &= \text{concat}_{\text{time}}\big(B^{(k)}_{\pi_1}, \ldots, B^{(k)}_{\pi_8}\big) = \Pi Z^{(k)}. \end{aligned} \tag{24}$$

Here Π is the induced 83×83 row-permutation matrix, applied with the same temporal indexing across W6, W8, and W10; no new permutation is drawn for a different stage. Only valid speech rows are permuted, leaving the valid length, feature dimension, and neural trajectory unchanged. The permutation changes only the order of the contextual feature vectors, preserving their multiset and the context already encoded within each vector.

Matching Computation. For condition $c \in \{\text{correct}, \text{mismatch}, \text{shuffle}\}$, define $Z_c^{(k)}$ as the corresponding tensor above and compute:

$$D_{tu}^{(k,c)} = \left\| U_t^{(k)} - Z_{c,u}^{(k)} \right\|_2^2, \qquad 1 \le t \le 63, \quad 1 \le u \le 83,$$
$$d_{k,c} = \text{SoftDTW}_\gamma\left(D^{(k,c)}\right), \qquad \gamma = 0.2. \tag{25}$$

The same squared-distance construction, valid endpoint, and smoothing parameter are used for all three conditions in a stage. The raw endpoint $d_{k,c}$ is computed before the length normalization, fixed scale factors, and stage weights used by MOST; it is neither the Val CTC loss nor the total training objective. The recurrence, valid-region masking, and derivative $\partial d_{k,c}/\partial D_{tu}^{(k,c)}$, interpreted as expected path occupancy, are derived in Appendix E. For the block permutation, $D^{(k,\text{shuffle})} = D^{(k,\text{correct})}\Pi^\top$: distances are rearranged along the speech axis, while the monotonic-path constraint acts on the resulting order. Thus, the shuffled-feature comparison examines order sensitivity without adding or deleting feature rows or changing their values.

Results within Each Stage. Among the three conditions, the correct prior yields the lowest raw Soft-DTW cost for each stage pair: 2137.0, 1892.5, and 1694.6 for G2-W6, G4-W8, and G6-W10, respectively. Within these stages, different-transcript speech raises the cost to 1.753, 2.629, and 3.737 times the corresponding correct-prior cost, while time-block permutation raises it to 2.040, 2.767, and 3.885 times. The correct-prior panels show smoothly progressing alignment contours, whereas the perturbed priors introduce longer horizontal or vertical segments, particularly under temporal shuffling. For this sentence, the relative cost penalties for both perturbations increase across the three stage pairs, indicating greater separation between the associated and perturbed speech features under the matching objective.

C.8 Edit Error Structures

To characterize the sources and distribution of decoding gains, we examine phoneme edit types, sentence-level outcomes, and gains stratified by baseline decoding difficulty. We compare the MOST-OPSD against the saved GRU/TMT predictions on the same validation set. Each model uses a fixed validation-selected checkpoint for all recording days. Comparisons use neural-only greedy predictions and archived sequence-level edit alignments, rather than language-model outputs or reference-conditioned token classification. Tables 10, 11, and 12 provide the complete results for Figure 3(f-h).

Day-level Aggregation. For baseline b, let N_b denote the number of recorded model combinations, each pairing a baseline model with a MOST-OPSD model on the same sentences. Let $g_{b,p,d}$ denote a panel-specific statistic for model combination p on recording day d. We average comparisons within each day and then weight the 15 validation days equally:

$$\overline{g}_{b,d} = \frac{1}{N_b}\sum_{p=1}^{N_b} g_{b,p,d}, \qquad \widehat{g}_b = \frac{1}{15}\sum_{d=1}^{15} \overline{g}_{b,d}. \tag{26}$$

Descriptive 95% day-bootstrap intervals are computed by resampling the day-level averages while preserving the Baseline-Ours pairing. These intervals are conditional on the saved model collection. Sentence outcomes are reported as point estimates.

Deletion, Insertion, and Substitution Reductions. Let $\mathcal{I}_d$ be the sentences on day d, r_i their reference lengths, and $E_{b,p,i}^{(t)}$ and $E_{\text{M},p,i}^{(t)}$ the baseline and MOST-OPSD edit counts of type $t \in \{D, I, S\}$. The per-comparison reduction is:

$$\Delta_{b,p,d}^{(t)} = 100 * \frac{\sum_{i\in\mathcal{I}_d}\left(E_{b,p,i}^{(t)} - E_{\text{M},p,i}^{(t)}\right)}{\sum_{i\in\mathcal{I}_d} r_i}. \tag{27}$$

We aggregate these reductions using Equation 26; positive values indicate fewer edits per 100 reference phonemes. As shown in Table 10, MOST-OPSD reduces all three error types relative to

both baselines. Substitutions account for the largest improvement, decreasing by 5.5382 percentage points relative to GRU and 1.7861 points relative to TMT. Deletion reductions provide the second-largest contribution, followed by insertion reductions. The predominance of substitution reductions indicates that improved phoneme selection is a major contributor to the decoding gains.

Table 10: Complete edit-type reductions for Figure 3(f). Entries are mean reduction [descriptive 95% day-bootstrap interval], in PER percentage points; positive values favor MOST-OPSD. Edit counts and reference lengths are pooled within each recording day, model comparisons are averaged within that day, and the 15 days receive equal weights. The intervals are obtained by resampling recording days after within-day aggregation.

Edit type	Relative to GRU	Relative to TMT
Deletion	$2.8043\ [2.4937, 3.1759]$	$0.8553\ [0.5794, 1.1369]$
Insertion	$0.7128\ [0.5487, 0.8835]$	$0.5872\ [0.3862, 0.8197]$
Substitution	$5.5382\ [5.1138, 5.9884]$	$1.7861\ [1.4230, 2.1431]$

Sentence-level Improvement. For each model combination, we classify each sentence by whether MOST-OPSD makes fewer, equal, or more total edits $D + I + S$ than the baseline, then average the corresponding sentence fractions within day and equally across days. The mean improved fractions are 78.7750% relative to GRU and 53.4250% relative to TMT, while the regressed fractions are 7.2833% and 19.9583%, respectively (Table 11). Ties mean equal edit counts, not necessarily identical predicted sequences or error-free decoding. These results show broad gains: fewer edits are observed for a majority of sentence comparisons, while a small subset regresses relative to TMT.

Table 11: Complete sentence-outcome proportions for Figure 3(g). Improved, Tied, and Regressed indicate fewer, equal, and more total phoneme edits by MOST-OPSD, respectively. Values are percentages of sentence comparisons, averaged over model combinations within each recording day and then equally over the 15 days.

Outcome	Relative to GRU (%)	Relative to TMT (%)
Improved	78.7750	53.4250
Tied	13.9417	26.6167
Regressed	7.2833	19.9583

Baseline-Difficulty Stratification. We retain the archived quartile assignments based on each baseline's sentence-level PER, with fixed sample identifiers breaking ties; Q1 denotes the easiest and Q4 the hardest baseline-defined group. The baseline and MOST-OPSD are evaluated on the same sentence support within each comparison and stratum, with model comparisons combined within days before equal-day averaging and day bootstrap. For GRU, mean PER reductions increase from 2.1383 points in Q1 to 16.3278 in Q4; for TMT, the Q2–Q4 reductions increase from 1.8774 to 4.3712 and 7.5932 points (Table 12). The TMT-defined Q1 is an exception: the mean reduction is -1.2691 points, indicating higher MOST-OPSD error on this easiest baseline-conditioned group. Thus, the largest conditional gains occur in difficult groups, but the aggregate benefit does not imply improvement in every difficulty stratum.

These decoding gains are driven primarily by fewer substitution errors and extend across a majority of sentence comparisons. Their magnitude generally increases with baseline-defined difficulty. Together, these results characterize the error structure and coverage of MOST-OPSD, complementing the component-level analyses in Appendix C.2.

C.9 Qualitative Case Studies

To examine how the two complementary transfer modules appear in individual predictions, we analyze neural-speech trajectories and phoneme posterior dynamics. The trajectory cases examine the temporal organization targeted by MOST representation transfer, while the posterior cases examine the task-aligned phoneme decisions targeted by OPSD predictive transfer. Together, these analyses connect the framework's two design objectives to concrete sequence-level improvements.

Table 12: Complete baseline-difficulty results for Figure 3(h). Entries are mean PER reduction [descriptive 95% day-bootstrap interval], in percentage points. Quartiles are defined separately from each baseline's sentence-level PER, with fixed sample identifiers resolving ties. Model-combination effects are aggregated within recording days before equal-day averaging and bootstrap resampling. Positive values favor MOST-OPSD.

Difficulty group	Relative to GRU	Relative to TMT
Q1 (easiest)	2.1383 $[1.4285, 2.7769]$	-1.2691 $[-1.8498, -0.7480]$
Q2	7.1244 $[6.3614, 7.8223]$	1.8774 $[1.2357, 2.4606]$
Q3	10.7732 $[10.1298, 11.4315]$	4.3712 $[3.8238, 4.9157]$
Q4 (hardest)	16.3278 $[15.1703, 17.5100]$	7.5932 $[6.8354, 8.4005]$

MOST: Strengthening Neural-Speech Temporal Correspondence. We first investigate whether speech-informed representation learning produces neural trajectories that more closely follow the temporal structure of speech. Figure 13 compares three sentences across GRU, TMT, and MOST-OPSD, using the same speech reference within each row and a shared PCA of temporal-similarity profiles. MOST-OPSD achieves CKA values of 0.783, 0.776, and 0.768, compared with 0.629, 0.586, and 0.629 for GRU and 0.615, 0.623, and 0.636 for TMT. The trajectory plots reveal closer correspondence in the progression and turning patterns of neural and speech representations, complementing the higher agreement measured in the full temporal-similarity space. These examples illustrate the representational advantage sought by MOST: organizing neural activity around speech-related temporal structure to support downstream phoneme decoding.

OPSD: Strengthening Task-Aligned Phoneme Decisions. We next examine how the predictive transfer targeted by OPSD is reflected in the neural-only decoder's recovery of phonemes missed by both baselines. Figure 14 tracks the same reference-phone occurrence across models for three sentences, relating target, competing-phone, and blank posteriors to actual greedy emissions. MOST-OPSD recovers the target phones /NG/, /AH/, and /IH/ with pronounced posterior peaks and corresponding greedy emissions, while GRU and TMT miss these occurrences. The local recoveries accompany lower full-sequence edit counts: MOST-OPSD yields 5, 2, and 1 errors, compared with 9, 6, and 6 for GRU and 9, 5, and 7 for TMT. These examples illustrate the predictive advantage sought by OPSD: strengthening target-phone evidence at consequential decision points and carrying speech-informed supervision into neural-only inference.

The two case studies illustrate the complementary benefits of the MOST-OPSD design at different levels of neural decoding. MOST targets the temporal organization of intermediate representations, while OPSD targets the phoneme probabilities produced from the final neural states. The observed combination of closer speech correspondence and improved phoneme recovery provides a sequence-level account of how these two transfer objectives support effective neural-only decoding.

C.10 Computational Efficiency

We compare the parameter counts, training throughput, inference latency, and peak training memory of the neural decoding methods to characterize the computational cost of multi-stage speech supervision and predictive transfer. Table 13 reports the measured efficiency of BiGRU, TMT, MOST, OPSD, and the complete MOST-OPSD model.

MOST-OPSD contains 158.24M parameters during training and 152.73M during inference, with a training throughput of 3.551 ± 0.015 iterations/s. Its peak training memory is 4.02 GB, compared with 5.15 GB for BiGRU and 0.84 GB for TMT. The complete method has substantially lower training throughput than OPSD alone (10.270 ± 0.004 iterations/s), reflecting the additional computation associated with multi-stage representation alignment. MOST and MOST-OPSD exhibit similar training throughput and peak memory.

MOST-OPSD combines substantial decoding improvements with an inference latency of 13.40 ms, only 2.71 ms above the BiGRU baseline. MOST, OPSD, and MOST-OPSD retain the same 152.73M inference parameters and nearly identical latencies, demonstrating that the two transfer modules strengthen decoding while preserving the deployed neural prediction path. This millisecond-scale

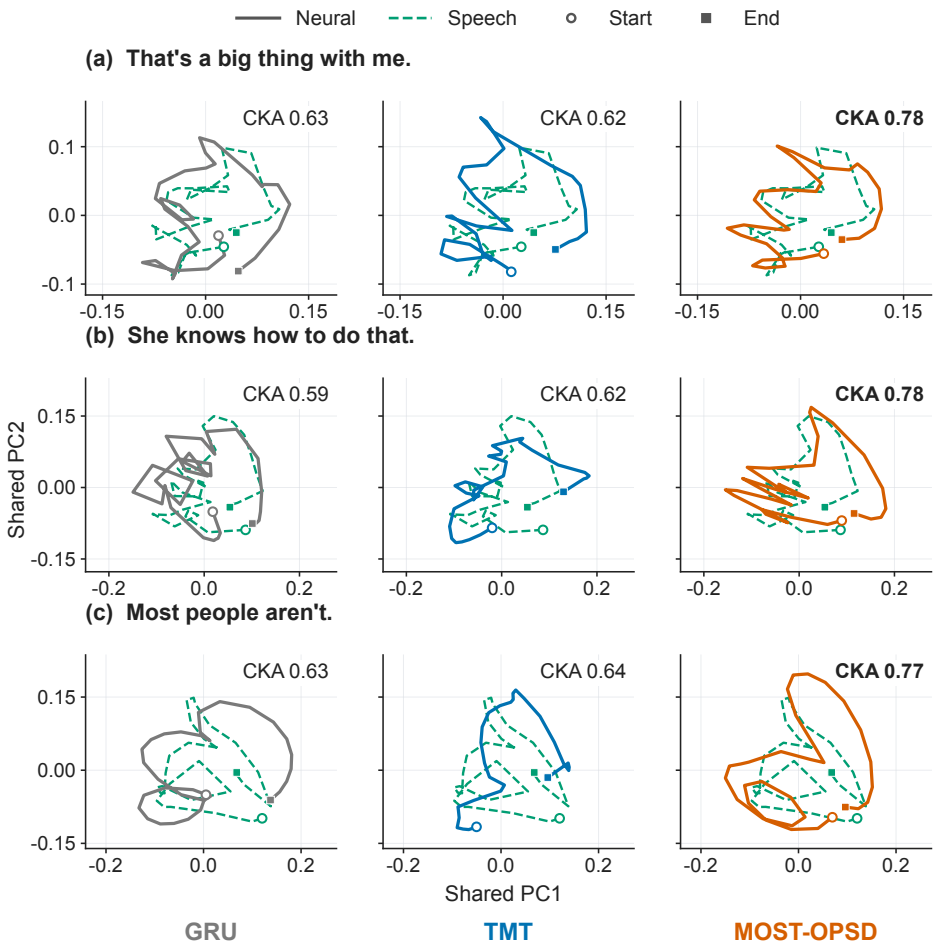

Figure 13: Neural-speech temporal correspondence in three illustrative cases. Columns compare GRU, TMT, and MOST-OPSD. Solid curves show neural trajectories, and dashed green curves show the same speech trajectory within each row. Open circles and filled squares mark sequence starts and ends. Temporal-similarity profiles from 32 normalized-time bins are projected using a shared PCA fitted to speech profiles. Axes share limits within each row and preserve equal aspect ratios. CKA measures agreement in the full temporal-similarity space. MOST-OPSD has higher correspondence than both baselines in these cases.

Table 13: Computational efficiency of neural decoding methods. Parameter counts are in millions (M); training throughput and inference latency are reported as mean ± SD.

Method	Train params (M)	Infer params (M)	Train iterations/s	Latency (ms)	Peak train memory (GB)
BiGRU	133.85	133.85	10.519 ± 0.018	10.69 ± 3.28	5.15
TMT	9.38	9.38	58.662 ± 0.145	2.39 ± 0.12	0.84
MOST	157.45	152.73	3.594 ± 0.024	13.42 ± 3.95	4.00
OPSD	153.52	152.73	10.270 ± 0.004	13.40 ± 3.95	4.62
MOST-OPSD	158.24	152.73	3.551 ± 0.015	13.40 ± 3.95	4.02

neural computation supports the responsiveness needed for interactive clinical communication, with auxiliary speech processing confined to training. Future work will further optimize training throughput and inference efficiency through implementation improvements and model compression.

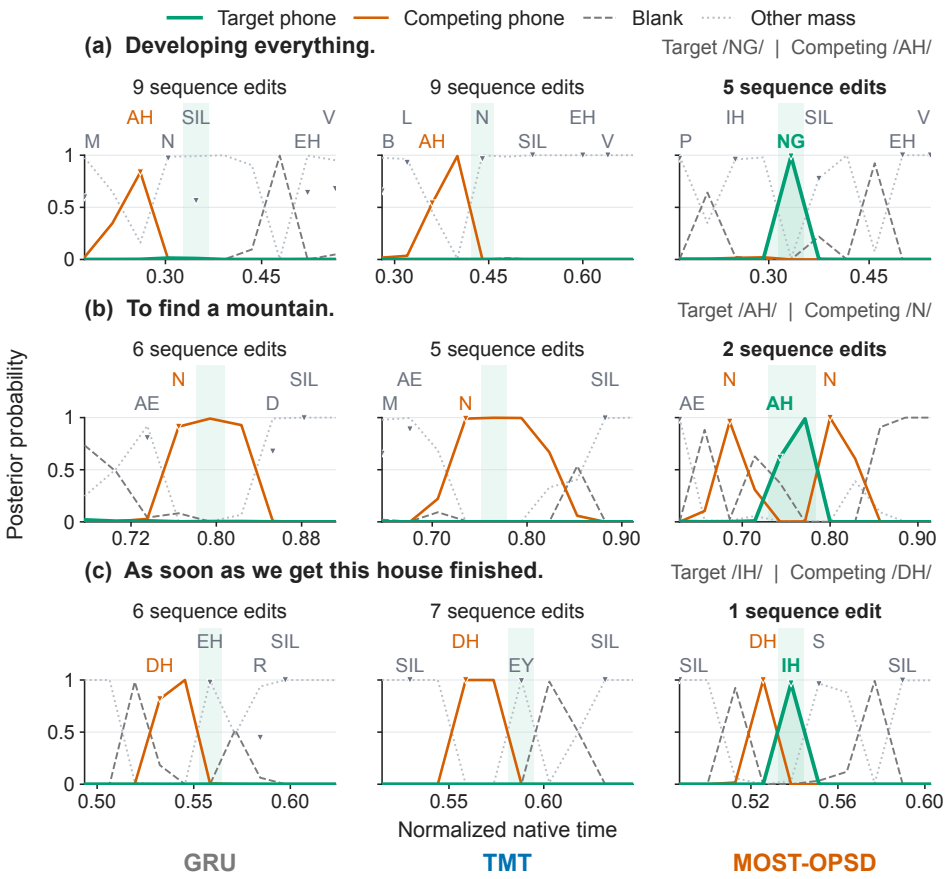

Figure 14: Phoneme recovery in three illustrative CTC posterior cases. Each row compares the same sentence and reference-phone occurrence across GRU, TMT, and MOST-OPSD. Curves show neural-only posteriors: target phone in green, competing phone in orange, blank in dashed gray, and remaining probability mass in dotted light gray. Triangles and phone labels mark greedy emissions. Shading indicates the target interval inferred by reference-conditioned CTC alignment. Each panel retains its model-specific native-time window; headings report full-sequence edit counts.

D Hyperparameter Settings

This appendix consolidates the settings for the two dataset experiments, the comparison baselines, and then reports the complete fixed-budget sensitivity summaries. Implementations are described in Appendix B; the mathematical effects of alignment and distillation weights are derived in Appendix E.

D.1 Model and Optimization Settings

Experiment Configurations. Brain-to-Text '24 uses a six-layer bidirectional GRU with 1024 hidden units per direction, whereas Brain-to-Text '25 uses six bidirectional layers with 768 units per direction. Tables 14-16 specify their inputs, speech-transfer modules, and optimization settings at the recorded precision. Both models use recording-day affine transformations, Softsign, and a shared 41-class phoneme head; T15 additionally specifies learnable initial recurrent states for each layer and direction. These configurations are not parameter-matched to the native baselines and must not be replaced by the 1024-unit cost configuration below.

Transfer Coefficients. The default teacher is frozen Wav2Vec 2.0 Base. Both datasets pair G2-W6, G4-W8, and G6-W10, with independent projections to 768 dimensions; the stage weights in

Table 14: Neural architecture and preprocessing for the two dataset experiments of our method.

Setting	Brain-to-Text '24	Brain-to-Text '25
Input features	256	512
GRU layers	6	6
Directionality	Bidirectional	Bidirectional
Hidden units per direction	1024	768
Window width / stride	$32/4$	$14/4$
Output classes / blank index	$41/0$	$41/0$
GRU dropout	0.30	0.25
Input dropout	0.15	0.20
Gaussian smoothing SD	2	2
Gaussian kernel length	–	100
White-noise SD	0.40	0.80
Constant-offset SD	0.05	0.20
Random-walk SD	–	0.10
Time-mask fraction	0.075	0.05
Number of time masks	20	10
Training batch size	32	64
Balanced recording days per batch	–	4

Table 15 are used directly. The values of λ, β, and μ apply to the total objective in Section 3.4, including division of the two CTC terms by $1 + \beta$.

Table 15: Speech-transfer settings for the two dataset experiments.

Setting	Brain-to-Text '24	Brain-to-Text '25
Stage weights (w_6, w_8, w_{10})	$(0.40, 0.46, 1.00)$	$(0.40, 0.54, 1.00)$
MOST coefficient λ	0.0003	0.0004
Soft-DTW smoothing γ	0.2	0.2
Privileged CTC weight β	0.25	0.25
KL coefficient μ	0.005	0.005
Temperature τ	2	2
Nonblank KL scale	1	1
Attention dimension / heads	$128/4$	$128/4$
Attention dropout	0.1	0.1

Optimization. Brain-to-Text '24 uses Adam with piecewise-linear learning rates through $(0, 0.03)$, $(40{,}000, 0.003)$, and $(100{,}000, 0.0003)$, where each pair specifies an update index and rate. Brain-to-Text '25 uses AdamW with a 1,000-update learning-rate warm-up and the recorded cosine-decay settings for both backbone and recording-day parameter groups. In both performance experiments, the first 20,000 updates train neural CTC and MOST, after which privileged CTC and KL are activated within the same optimization process. Neural-only validation PER selects the performance checkpoint, with CTC breaking ties; training algorithms are given in Appendix B.4.

Brain-to-Text '24 Baselines. Table 17 transcribes the saved GRU and TMT configurations used in Appendix A.3. The baseline is a five-layer bidirectional GRU, whereas TMT uses causal time-masked attention and its native log-feature input. Their batch sizes, perturbations, and learning-rate settings are retained. In particular, TMT's equal configured start/end rates do not imply a constant rate, since its saved scheduler is multistep; the milestones and decay factor are separate required settings.

Brain-to-Text '25 Baselines. The original released GRU baseline retains its original data protocol, while the local Kord reproduction and MOST-OPSD use the expanded training split (Table 18). The supplied Kord description specifies a five-layer GRU, but does not supply a complete numeric

Table 16: Optimization settings for MOST-OPSD performance runs.

Setting	Brain-to-Text '24	Brain-to-Text '25
Optimizer betas	$(0.9, 0.999)$	$(0.9, 0.999)$
Optimizer epsilon	0.1	0.1
Weight decay	10^{-5}	10^{-4}
Learning-rate schedule	Piecewise linear	Warm-up + cosine
Maximum / minimum LR	$0.03/0.0003$	$0.004/0.00008$
Maximum training updates	100,000	120,000
OPSD activation, update	20,000	20,000
Gradient-clipping norm	2	10

Table 17: All supplied numeric and scheduler fields for the BTT24 neural baselines.

Setting	GRU	TMT
Layers / depth	5	5
Recurrent units / model dimension	1024	384
Recorded input-feature count	256	–
Window width / stride	$32/4$	–
Attention heads / head dimension	–	$6/64$
MLP dimension ratio	–	4
Batch size	64	64
Dropout	0.40	0.35
Input dropout	0	0.20
Gaussian smoothing width	2	2
White-noise SD	0.80	0.20
Constant-offset SD	0.20	0.05
Time-mask fraction	0	0.075
Number of time masks	0	20
Configured LR start / end	$0.02/0.02$	$0.001/0.001$
Recorded L2 decay	10^{-5}	10^{-5}
Optimizer betas	$(0.9, 0.999)$	$(0.9, 0.999)$
Learning-rate scheduler	None	Multistep

optimization configuration. The Kord single-model evaluation uses the local reproduction, whereas the fusion comparison uses the archived official result.

Table 18: Available Brain-to-Text '25 baseline configuration information.

Reference	Data protocol	Supplied model
GRU (#baseline)	Original release split	Official released
Kord (#10)	Expanded split: 9,336 training sentences	Five-layer GRU reproduction

D.2 Hyperparameter Sensitivity

Fixed-Budget Protocol. We examine w_6, w_8, λ, γ, β, and μ on the same Brain-to-Text '24 validation set at exactly 25,000 updates (Figure 15). Val PER is read from the fixed-update checkpoint. The MOST panels summarize an existing joint configuration search, whereas the OPSD panels summarize a $\beta \times \mu$ grid with temperature $\tau = 2$.

Configurations. For a metric M, let $\overline{M}_j$ be its mean over the recorded training repetitions within configuration j, each evaluated at update 25,000. For displayed parameter q and location x, let $\mathcal{C}_{q,x}$ contain the configurations assigned to that value or its registered bin. The displayed center and

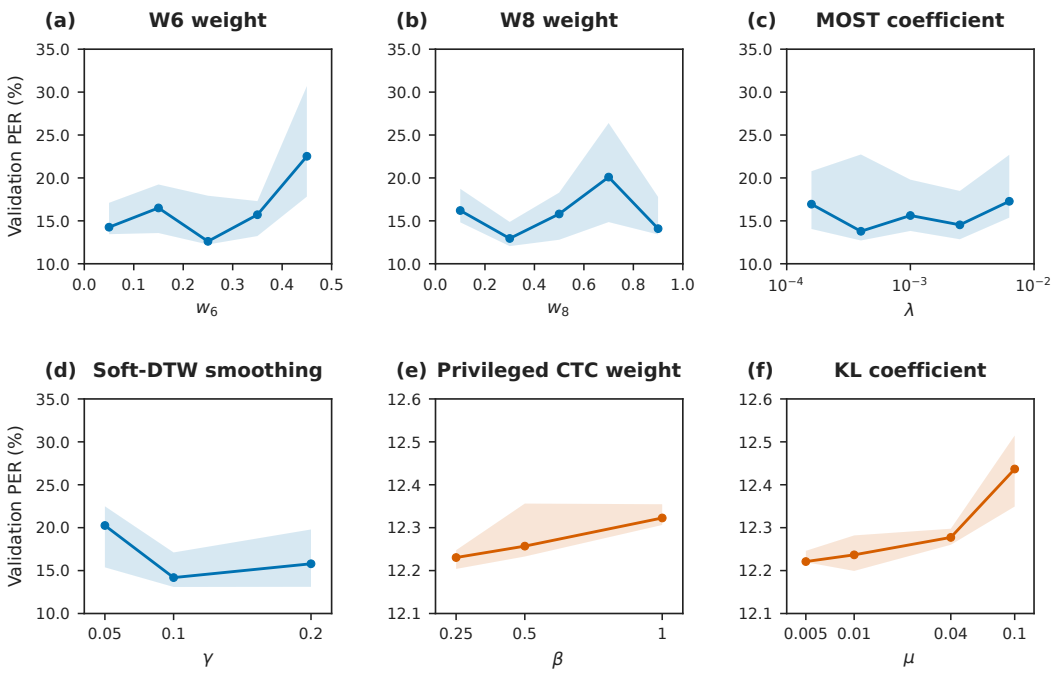

Figure 15: Fixed-budget Val PER sensitivity at 25,000 updates on the validation set. **(a-d)** Conditional summaries for $w_6, w_8, \lambda, \gamma$ of the MOST module; weight and coefficient locations are registered bin centers. **(e,f)** Marginal summaries of the same $\beta \times \mu$ grid of the OPSD module, with temperature 2. Lines show medians, and shaded bands span the 25th-75th percentiles across configuration means.

limits:

$$\begin{aligned} \overline{M}_j &= \frac{1}{R_j} \sum_{r=1}^{R_j} M_{j,r}(25{,}000), \\ \widehat{M}_q(x) &= \text{median}_{j \in \mathcal{C}_{q,x}} \overline{M}_j, \\ [L_q(x), U_q(x)] &= [Q_{0.25}(\{\overline{M}_j\}_{j \in \mathcal{C}_{q,x}}), Q_{0.75}(\{\overline{M}_j\}_{j \in \mathcal{C}_{q,x}})]. \end{aligned} \tag{28}$$

Thus, each configuration contributes one summary rather than one observation for every training repetition; the shaded band spans the configuration quartiles, with IQR width $U_q(x) - L_q(x)$. It is not a sample SD, a confidence interval, or a recording-day bootstrap interval. For w_6, w_8, λ, x is a recorded bin center rather than a claim that every configuration used exactly that number; γ, β, μ use the listed discrete settings.

Representation-Supervision Parameters. The weight summaries are nonmonotonic: the lowest displayed PER medians occur at w_6 bin center 0.25 (12.612%) and w_8 bin center 0.30 (12.946%), while the $w_6 = 0.45$ bin has a much higher median (22.523%) and a broad configuration spread. For λ, the 0.0004 bin has the lowest displayed median PER and CTC (13.784%, 0.610), whereas the largest coefficient bin reaches a median CTC of 1.041. Among the three smoothing settings, $\gamma = 0.10$ has lower fixed-budget medians (14.184%, 0.610) than 0.05 or 0.20. These conditional patterns are consistent with balancing speech-structured supervision against phoneme decoding. The final $\gamma = 0.2$ and dataset-specific weights remain those recorded in Table 15; the fixed-budget curves do not replace them.

Predictive-Transfer Parameters. The grid contains 12 configurations: three values of β (0.25, 0.50, 1.00) crossed with four values of μ (0.005, 0.010, 0.040, 0.100). A β point summarizes the four configuration means across μ, while a μ point summarizes the three means across β. Within the OPSD grid, $\beta = 0.25$ has the lowest median PER and CTC, 12.230% and 0.496, after summarizing

over μ. Increasing μ from 0.005 to 0.100 raises the PER median from 12.221% to 12.437% and the CTC median from 0.484 to 0.639, after summarizing over β. The results favor the weaker tested predictive-supervision region at this budget.

E Mathematical Details

This appendix derives the objectives implemented in Appendix B. We first establish valid-length alignment and fixed-scale normalization, then decompose full-vocabulary distillation and trace its gradients through the shared parameters, and finally separate training computations from the deployed neural path. These results characterize the implemented supervision.

Notation and Assumptions. Let B be the minibatch size, K the number of speech views, $r_i = |y_i| > 0$ the reference length, and $V = |\mathcal{V}_{\mathrm{CTC}}| = 41$ the output vocabulary size, with blank index zero. The encoder parameters θ include recording-day transformations; ψ denotes the shared head, ϕ the privileged adapter, and P_k the independent MOST projections. Lengths and masks are fixed metadata; every scored speech view has a positive valid length, speech features are detached constants, and the calibration constants are finite and strictly positive. We differentiate with fixed $\gamma > 0$, $\tau > 0$, and nonnegative loss weights; dropout realizations are held fixed when taking a training gradient, while the permutation result explicitly assumes evaluation mode. References contain nonblank task labels, and the CTC derivative assumes a nonempty valid path set and finite logits. Gradients follow the per-step computational graph with its detached target, before clipping, weight decay, or an optimizer update.

E.1 Length-Aware Soft-DTW

Valid Window Count and Padding. Assume valid neural input length $T_i \geq k_{\mathrm{win}}$, integer stride $s \geq 1$, no padding added by the window-unfolding operation, and no subsequent change in temporal length. Window j, starting at $1 + (j-1)s$, is valid precisely when $k_{\mathrm{win}} + (j-1)s \leq T_i$. Counting these starts gives:

$$n_i = \left\lfloor \frac{T_i - k_{\mathrm{win}}}{s} \right\rfloor + 1, \qquad M^{\mathrm{neu}}_{it} = \mathbf{1}\{1 \leq t \leq n_i\}. \tag{29}$$

The speech mask similarly uses $1 \leq u \leq m_i^{(k)}$. Thus the alignment for sample i and view k is a function of the valid feature rectangle alone. This does not follow merely from setting padded distances to zero: scoring a padded endpoint would introduce additional paths and alter the objective. At the encoder-input level, padding independence additionally requires valid states not to depend on padded values; the documented padding clearing around smoothing and packed bidirectional recurrence address this requirement (Appendix B.1).

Path Partition and Dynamic Program. Suppress the sample and view indices and set $N = n_i$, $M = m_i^{(k)}$, $u_t = P_k(H_{i,t}^{(l_k)})$, and $D_{tu} = \|u_t - Z_u\|_2^2$. Let $\mathcal{A}_{N,M}$ contain binary occupancy matrices of paths from $(1,1)$ to (N,M) with steps $(1,0)$, $(0,1)$, and $(1,1)$. With $C_a = \langle a, D \rangle$, define:

$$\mathcal{Z}_\gamma(D) = \sum_{a \in \mathcal{A}_{N,M}} e^{-C_a/\gamma}, \qquad d_\gamma(D) = -\gamma \log \mathcal{Z}_\gamma(D). \tag{30}$$

To derive the recurrence (Cuturi & Blondel, 2017), let F_{tu} be the partition sum of partial paths ending at (t,u). Grouping paths by their last predecessor yields:

$$\begin{aligned} &F_{tu} = e^{-D_{tu}/\gamma}(F_{t-1,u} + F_{t,u-1} + F_{t-1,u-1}), \\ &F_{00} = 1, \qquad F_{t0} = F_{0u} = 0 \quad (t,u > 0), \\ &A_{tu} = -\gamma \log F_{tu} = D_{tu} - \gamma \log\Big(e^{-A_{t-1,u}/\gamma} + e^{-A_{t,u-1}/\gamma} + e^{-A_{t-1,u-1}/\gamma}\Big). \end{aligned} \tag{31}$$

The initialization becomes $A_{00} = 0$ and positive-index boundary values $+\infty$, and $d_\gamma(D) = A_{NM}$, matching Equations 4–5. Monotonicity prevents a path to this endpoint from visiting a future padded row or column, so differentiation gives zero with respect to all excluded distance entries and feature rows.

Hard-DTW Limit and Entropy Term. Write $C_{\min} = \min_a C_a$ and $Q = |\mathcal{A}_{N,M}|$. Factoring out the minimum-cost exponential gives:

$$d_\gamma = C_{\min} - \gamma \log \sum_{a \in \mathcal{A}_{N,M}} e^{-(C_a - C_{\min})/\gamma}, \qquad C_{\min} - \gamma \log Q \le d_\gamma \le C_{\min}. \tag{32}$$

Proof. The sum is at least one, since a minimum-cost path exists, and at most Q, since every summand is at most one. Taking its logarithm proves the bounds; for fixed finite lengths, they imply $d_\gamma \to C_{\min}$ as $\gamma \downarrow 0$. For Gibbs weights $\rho_a = e^{-C_a/\gamma}/\mathcal{Z}_\gamma$, substituting their logarithms into entropy gives the exact identity:

$$d_\gamma = \sum_a \rho_a C_a - \gamma \mathcal{H}(\rho), \qquad \mathcal{H}(\rho) = -\sum_a \rho_a \log \rho_a. \tag{33}$$

Raw Soft-DTW can therefore be negative even with squared distances; it is not a nonnegative, self-corrected divergence. The implemented outer normalization retains this soft-minimum objective rather than removing its entropy term.

Expected Occupancy and Feature Gradients. Differentiating the finite partition sum gives:

$$\frac{\partial d_\gamma}{\partial D_{tu}} = -\frac{\gamma}{\mathcal{Z}_\gamma} \sum_a e^{-C_a/\gamma} \left(-\frac{a_{tu}}{\gamma} \right) = \sum_a \rho_a a_{tu} \equiv E_{tu}. \tag{34}$$

Each E_{tu} is an expected path occupancy in $[0, 1]$, not a row-normalized attention weight. Indeed, a path visits between $\max(N, M)$ and $N + M - 1$ cells, so:

$$\sum_{t,u} E_{tu} = \mathbb{E}_\rho[|a|], \qquad \max(N, M) \le \mathbb{E}_\rho[|a|] \le N + M - 1. \tag{35}$$

The chain rule for the squared distances then yields:

$$\frac{\partial d_\gamma}{\partial u_t} = 2 \sum_{u=1}^{M} E_{tu}(u_t - Z_u), \qquad \frac{\partial d_\gamma}{\partial H_{i,t}^{(l_k)}} = (J_{H_{i,t}^{(l_k)}} P_k)^\top \frac{\partial d_\gamma}{\partial u_t}. \tag{36}$$

For an affine projection $P_k(h) = W_k^P h + b_k^P$, its weight gradient is $\sum_t (\partial d_\gamma / \partial u_t) h_t^\top$ and its bias gradient is $\sum_t \partial d_\gamma / \partial u_t$. No speech-encoder gradient is propagated, because Z is detached. These derivatives combine multiple feasible temporal correspondences, explaining how asynchronous trajectories receive continuous supervision and why the occupancy contours in Figure 12 are not measured phoneme boundaries.

Fixed Length and Calibration Scales. For the designated training calibration set $\mathcal{C}$, with $d_{i,\mathrm{cal}}^{(k)}$ computed at calibration time, the recorded scales are:

$$c_k = \mathrm{median}_{i \in \mathcal{C}} \left| \frac{d_{i,\mathrm{cal}}^{(k)}}{n_i + m_i^{(k)}} \right|, \qquad c_{\mathrm{ref}} = \mathrm{median}_{i \in \mathcal{C}} \, |d_{i,\mathrm{cal}}^{(K)}|. \tag{37}$$

The constants remain fixed after calibration, and the actual objective is:

$$\begin{aligned} \widetilde{d}_i^{(k)} &= \frac{c_{\mathrm{ref}}}{(n_i + m_i^{(k)}) c_k} \, d_i^{(k)}, \\ \mathcal{L}_{\mathrm{MOST}} &= \frac{1}{B} \sum_{i=1}^{B} \sum_{k=1}^{K} w_k \widetilde{d}_i^{(k)}, \\ \nabla_\eta \mathcal{L}_{\mathrm{MOST}} &= \frac{1}{B} \sum_{i,k} \frac{w_k c_{\mathrm{ref}}}{(n_i + m_i^{(k)}) c_k} \nabla_\eta d_i^{(k)}. \end{aligned} \tag{38}$$

Here η is any relevant encoder or projection parameter; no derivative passes through lengths, fixed weights, or calibration medians. The denominator $n_i + m_i^{(k)}$ is a length proxy, not the visited-cell count or expected path length in Equation 35. The absolute value is used to estimate scales. If a calibration median is zero, the ratio is undefined: a reproduction must specify its actual safeguard rather than infer an undocumented epsilon. Finally, for $a > 0$, the partition formula implies $d_{a\gamma}(aD) = a\, d_\gamma(D)$; scaling distances at fixed γ generally changes the Gibbs path weights, whereas multiplying the final loss by a fixed positive constant does not. This separates smoothing from outer scale calibration when interpreting Appendix D.2.

E.2 Posterior Decomposition and Gradients

Full-Vocabulary Decomposition. Consider one valid frame and write $p = p^\tau$ and $q = q^\tau$ for its student and detached privileged distributions. Finite logits and finite positive τ give strictly positive probabilities in exact arithmetic. Define $p_+ = 1 - p_0$, $q_+ = 1 - q_0$, $\overline{p}_v = p_v/p_+$ and $\overline{q}_v = q_v/q_+$ for $v > 0$; the nonblank set includes SIL as well as the 39 phonemes. Then:

$$\begin{aligned} D_{\mathrm{KL}}(q\|p) &= q_0 \log \frac{q_0}{p_0} + q_+ \log \frac{q_+}{p_+} + q_+ D_{\mathrm{KL}}(\overline{q}\|\overline{p}) \\ &= D_{\mathrm{KL}}\big((q_0, q_+)\|(p_0, p_+)\big) + q_+ D_{\mathrm{KL}}(\overline{q}\|\overline{p}). \end{aligned} \tag{39}$$

Proof. Substitute $q_v = q_+ \overline{q}_v$ and $p_v = p_+ \overline{p}_v$ in the nonblank terms:

$$\begin{aligned} \sum_{v>0} q_v \log \frac{q_v}{p_v} &= q_+ \sum_{v>0} \overline{q}_v \left(\log \frac{q_+}{p_+} + \log \frac{\overline{q}_v}{\overline{p}_v} \right) \\ &= q_+ \log \frac{q_+}{p_+} + q_+ D_{\mathrm{KL}}(\overline{q}\|\overline{p}), \end{aligned} \tag{40}$$

since $\sum_{v>0} \overline{q}_v = 1$; adding the blank term proves the result. Thus full-vocabulary KL transfers both blank/nonblank mass and conditional label preferences, with the latter weighted by the teacher's nonblank mass. Dropping the first term, omitting q_+, or independently rescaling the last term changes the objective in general; the implementation's nonblank scale of one preserves the exact identity.

Log-Domain Computation. For $a_v = \log p_v$ and $b_v = \log q_v$ obtained by `log_softmax`, compute:

$$\begin{aligned} \log p_+ &= \mathrm{logsumexp}_{v>0}\, a_v, \quad \log \overline{p}_v = a_v - \log p_+, \\ \log q_+ &= \mathrm{logsumexp}_{v>0}\, b_v, \quad \log \overline{q}_v = b_v - \log q_+. \end{aligned} \tag{41}$$

This avoids computing a small nonblank mass by subtracting a near-one blank probability. The formulas are exact for positive distributions; if probabilities are explicitly truncated to zero, conditional distributions require the corresponding limiting/support conventions rather than division by zero. The full log-domain KL need not explicitly form the conditional distributions to obtain the same scalar loss.

Temperature-Scaled Gradient. Let $N_{\mathrm{val}} = \sum_i n_i > 0$, $z_{it} = \ell_{it}^{\mathrm{neu}}$, and $z_{it}^{\mathrm{priv}} = \ell_{it}^{\mathrm{priv}}$. For fixed temperature, the loss is:

$$\begin{aligned} p_{it}^\tau = \mathrm{softmax}(z_{it}/\tau), \qquad q_{it}^\tau = \mathrm{softmax}(\mathrm{sg}(z_{it}^{\mathrm{priv}})/\tau), \\ \mathcal{L}_{\mathrm{KD}} = \frac{\tau^2}{N_{\mathrm{val}}} \sum_{i=1}^{B} \sum_{t=1}^{n_i} D_{\mathrm{KL}}(q_{it}^\tau \| p_{it}^\tau). \end{aligned} \tag{42}$$

Derivation. The detached target is constant in this backward pass, so only $-\sum_c q_c^\tau \log p_c^\tau$ contributes a student-logit derivative. Using:

$$\frac{\partial \log p_c^\tau}{\partial z_v} = \frac{\mathbf{1}\{c = v\} - p_v^\tau}{\tau} \quad \Longrightarrow \quad \frac{\partial \mathcal{L}_{\mathrm{KD}}}{\partial z_{it,v}} = \frac{\tau}{N_{\mathrm{val}}} (p_{it,v}^\tau - q_{it,v}^\tau) \quad (t \le n_i), \tag{43}$$

where $\sum_c q_c^\tau = 1$ completes the calculation. Gradients at padded frames and along the detached target-logit input are zero. This is a valid-frame mean, equivalently a sentence-KL average weighted by n_i/N_{val}, not an equal-weight mean of sentence means. The coefficient μ in the total objective multiplies this gradient but does not change its reduction.

Temperature Compensation. For fixed finite logits, the large-τ expansion is:

$$p_v^\tau = \frac{1}{V} + \frac{z_v - \overline{z}}{V\tau} + O(\tau^{-2}), \qquad \frac{\partial \mathcal{L}_{\mathrm{KD}}}{\partial z_v} = \frac{(z_v - \overline{z}) - (z_v^{\mathrm{priv}} - \overline{z}^{\mathrm{priv}})}{N_{\mathrm{val}} V} + O(\tau^{-1}). \tag{44}$$

Here bars denote class means and a single valid frame is understood; the expansion follows by applying $e^{z_v/\tau} = 1 + z_v/\tau + O(\tau^{-2})$ to the softmax numerator and denominator. Without the τ^2 multiplier, the same large-temperature gradient would decay as $O(\tau^{-2})$, explaining the standard compensation (Hinton et al., 2015). At finite temperature, the probabilities still change with τ; the multiplier does not make training temperature invariant. Temperature is a fixed hyperparameter here, not a learned variable whose derivative is being optimized.

CTC Gradients. Both CTC branches use temperature one. Let $\xi_{it,v}^b$ be the posterior probability that a CTC path emitting the reference occupies label v at frame t, for branch $b \in \{\mathrm{neu}, \mathrm{priv}\}$. For the path sum in Equation 13, differentiating its log gives:

$$\begin{aligned} \xi_{it,v}^b &= \frac{\sum_{\pi \in \Pi_i(y_i)} \mathbf{1}\{\pi_t = v\} \prod_{u=1}^{n_i} p_{iu}^b(\pi_u)}{\sum_{\pi \in \Pi_i(y_i)} \prod_{u=1}^{n_i} p_{iu}^b(\pi_u)}, \\ \mathcal{L}_{\mathrm{CTC}}^b &= \frac{1}{B} \sum_i \frac{\mathcal{J}_{i,\mathrm{CTC}}^b}{\nu_i}, \qquad \delta_{it,v}^b := \frac{\partial \mathcal{L}_{\mathrm{CTC}}^b}{\partial \ell_{it,v}^b} = \frac{p_{it}^b(v) - \xi_{it,v}^b}{B\nu_i}. \end{aligned} \tag{45}$$

Here $\nu_i = r_i$. To obtain the last equality, average the log-softmax derivative $\mathbf{1}\{\pi_t = v\} - p_{it}^b(v)$ under the normalized path weights, then negate it and apply the sample reduction. This requires a feasible path set; for ordinary CTC, adjacent repeated reference labels require separating blanks, so sufficient length is $n_i \geq r_i + \sum_{j=2}^{r_i} \mathbf{1}\{y_{ij} = y_{i,j-1}\}$.

Shared Parameters. Write the final state and residual as h_{it} and R_{it}, and use the shared affine head $g_\psi(h) = Wh + b$. Let $d_{it} = \tau(p_{it}^\tau - q_{it}^\tau)/N_{\mathrm{val}}$, and define the upstream logit gradients of the total objective by:

$$G_{it}^{\mathrm{neu}} = \frac{\delta_{it}^{\mathrm{neu}}}{1+\beta} + \mu d_{it}, \qquad G_{it}^{\mathrm{priv}} = \frac{\beta \delta_{it}^{\mathrm{priv}}}{1+\beta}. \tag{46}$$

Let $J_\eta \ell^b$ denote the Jacobian of all stacked valid branch logits with respect to parameter block η, and stack G^b in the same order. The chain rule for Equation 14 gives:

$$\begin{aligned} \nabla_\theta \mathcal{L} &= (J_\theta \ell^{\mathrm{neu}})^\top G^{\mathrm{neu}} + (J_\theta \ell^{\mathrm{priv}})^\top G^{\mathrm{priv}} + \lambda \nabla_\theta \mathcal{L}_{\mathrm{MOST}}, \\ \nabla_\psi \mathcal{L} &= (J_\psi \ell^{\mathrm{neu}})^\top G^{\mathrm{neu}} + (J_\psi \ell^{\mathrm{priv}})^\top G^{\mathrm{priv}}, \\ \nabla_\phi \mathcal{L} &= (J_\phi \ell^{\mathrm{priv}})^\top G^{\mathrm{priv}}, \qquad \nabla_{P_k} \mathcal{L} = \lambda \nabla_{P_k} \mathcal{L}_{\mathrm{MOST}}. \end{aligned} \tag{47}$$

In particular, the shared-head gradients are:

$$\begin{aligned} \nabla_W \mathcal{L} &= \sum_{i,t} \left[G_{it}^{\mathrm{neu}} h_{it}^\top + G_{it}^{\mathrm{priv}} (h_{it} + R_{it})^\top \right], \\ \nabla_b \mathcal{L} &= \sum_{i,t} (G_{it}^{\mathrm{neu}} + G_{it}^{\mathrm{priv}}), \qquad \ell_{it}^{\mathrm{priv}} - \ell_{it}^{\mathrm{neu}} = W R_{it}. \end{aligned} \tag{48}$$

The privileged Jacobian includes both the direct state and its role as an attention query: $J_\theta(h+R) = (I + J_h R) J_\theta h$ for stacked states. By contrast, there is no KL target-side Jacobian in Equation 47, even though encoder and head parameters are shared. They still receive the student KL gradient, and the adapter receives task supervision through undetached privileged CTC. A later parameter update can change both predictions when the target is recomputed; stop-gradient freezes the target only for the current backward pass, not for future updates. Before OPSD activation, its two weights are zero, and its branch is omitted, reducing these expressions to neural CTC and MOST.

Zero-Initialized Residual. Write the residual as $R_t = W_O F_{\omega,t}(H, Z) + b_O$. Initialize $W_O = 0$ and $b_O = 0$, and let ω collect the internal adapter parameters. Then $R = 0$, the two logits agree, $p^\tau = q^\tau$, and the KL value and student gradient are zero at that initialization. For upstream residual gradient $g_t = \partial \mathcal{L} / \partial R_t$,

$$\begin{aligned} \nabla_{W_O} \mathcal{L} &= \sum_t g_t F_{\omega,t}^\top, & \nabla_{b_O} \mathcal{L} &= \sum_t g_t, \\ \nabla_\omega \mathcal{L} &= \sum_t (J_\omega F_{\omega,t})^\top W_O^\top g_t = 0 & &\text{when } W_O = 0. \end{aligned} \tag{49}$$

Thus, privileged CTC can first update the output map; its loss gradient reaches internal attention parameters after the output map becomes nonzero, rather than requiring a nonzero KL gradient at initialization. The output gradient need not be nonzero on every example, and this statement does not exclude parameter changes from optimizer state or regularization. At the zero-residual initialization, the normalized sum of the two identical CTC terms equals neural CTC for the shared neural parameters.

Neural Time Grid and Joint Key/Value Permutation. There is one privileged output per neural query, so both branches have n_i valid frames without interpolating speech features onto the neural grid. For a single attention head, let $Q \in \mathbb{R}^{n \times d}$, $K, V_a \in \mathbb{R}^{m \times d}$, and let M be its additive key mask, with at least one valid key per query. For a permutation matrix Π on speech positions,

$$\begin{aligned} \mathrm{softmax}_{\mathrm{row}}\left(\frac{Q(\Pi K)^\top}{\sqrt{d}} + M\Pi^\top\right)(\Pi V_a) &= \mathrm{softmax}_{\mathrm{row}}\left(\left[\frac{QK^\top}{\sqrt{d}} + M\right]\Pi^\top\right)\Pi V_a \\ &= \mathrm{softmax}_{\mathrm{row}}\left(\frac{QK^\top}{\sqrt{d}} + M\right)\Pi^\top \Pi V_a \\ &= \mathrm{softmax}_{\mathrm{row}}\left(\frac{QK^\top}{\sqrt{d}} + M\right) V_a. \end{aligned} \tag{50}$$

Proof. A common column permutation reorders each row's exponentials without changing its softmax denominator. The resulting probability columns are permuted by $\Pi^\top$, which cancels the value-row permutation; the argument applies independently to every head and survives their concatenation and output projection. Mask products in this notation denote column reindexing; implementations reorder mask entries rather than numerically multiply zeros by $-\infty$. The identity holds mathematically, up to floating-point differences in computation, and is not generally a samplewise identity with independent dropout masks or unmatched positional terms. It concerns already contextualized features. This remains compatible with Soft-DTW's order-sensitive monotonic paths in Appendix C.7.

E.3 Training and Deployment Complexity

Scope and Dimensions. We count dense arithmetic for one sample with valid neural length n, teacher lengths m_k, common speech width d_s, stage widths d_k, final neural width $d_h = 2h$, total attention width a, and H_a attention heads. The recurrent encoder has L bidirectional layers with h units per direction and first-layer input width $d_{\mathrm{in}} = k_{\mathrm{win}} C$. Let m_* denote the deep attention view length and r the reference length. Training statements include forward and reverse differentiation of the active operations, which have the same leading dense-arithmetic orders here; minibatch work sums over samples when evaluated on their valid regions.

Representation Transfer. Independent projections, squared-distance construction, and dynamic programming contribute, respectively,

$$\begin{aligned} \mathcal{T}_{\mathrm{proj}} &= O\left(\sum_k n d_k d_s\right), & \mathcal{T}_{\mathrm{dist}} &= O\left(\sum_k n m_k d_s\right), \\ \mathcal{T}_{\mathrm{DP}} &= O\left(\sum_k n m_k\right), & \mathcal{S}_{\mathrm{DP}} &= O\left(\sum_k n m_k\right). \end{aligned} \tag{51}$$

The recurrence performs constant work per cell; its reverse traversal computes occupancy in the same order, and propagating occupancy through squared distances costs $O(nm_k d_s)$ per view. The space expression covers explicitly stored distance, dynamic-programming, and occupancy arrays up to constant factors, not every encoder activation or optimizer tensor. Processing views in chunks or recomputing intermediates changes peak memory and time; a rolling-row forward recurrence does not by itself give the same memory bound for full backpropagation. For dense padded batches, actual work may instead follow batch maxima; masks establish the correct loss without necessarily eliminating padded computation.

Privileged Attention and Sequence Losses. Summing over heads of width a/H_a, score construction and attention-weighted values cost $O(nm_* a)$. Query/output projections contribute $O(nd_h a)$ and key/value projections contribute $O(m_* d_s a)$, giving:

$$\begin{aligned} \mathcal{T}_{\text{att}} &= O(nd_h a + m_* d_s a + nm_* a), \\ \mathcal{S}_{\text{att}} &= O(H_a nm_* + (n + m_*)a). \end{aligned} \tag{52}$$

These are ordinary explicitly materialized attention costs; additional dense latent projections inside an attention implementation add $O((n + m_*)a^2)$ arithmetic, and its actual parameters belong to ϕ. The extra privileged phoneme head costs $O(nd_h V)$ even though it shares parameters with the neural head. Standard CTC forward/backward traverses $2r + 1$ expanded reference states per neural frame, giving $O(nr)$ per branch; forming vocabulary probabilities and full-vocabulary KL adds $O(nV)$ operations. Head sharing therefore removes duplicate parameters, not the privileged branch's training-time head computation.

Retained Neural Computation. The first-layer input multiplications cost $O(nd_{\text{in}}h)$. Recurrent multiplications and later layers cost $O(nLh^2)$, absorbing gate and direction counts into constants. Let $\mathcal{T}_{\text{pre}}$ denote the retained input preprocessing. The deployed forward path has complexity:

$$\mathcal{T}_{\text{deploy}} = O(\mathcal{T}_{\text{pre}} + nd_{\text{in}}h + nLh^2 + nd_h V). \tag{53}$$

Greedy collapse adds $O(nV + n)$ if the framewise argmax is included; CTC dynamic programming against a reference is not required at deployment. With all transfer objectives active, the per-update neural-training arithmetic is of order:

$$\mathcal{T}_{\text{train}} = O(\mathcal{T}_{\text{deploy}} + \mathcal{T}_{\text{proj}} + \mathcal{T}_{\text{dist}} + \mathcal{T}_{\text{DP}} + \mathcal{T}_{\text{att}} + nd_h V + 2nr + nV + |\Theta_{\text{train}}|)\,, \tag{54}$$

where the last term covers dense parameter-gradient bookkeeping, clipping, and the optimizer update up to constants; absent modules remove their terms during the initial schedule. Frozen-teacher feature generation is a separate preparation cost, not made free by excluding it from a cached training update; storing its feature cache scales with $\sum_{i,k} m_i^{(k)} d_s$ scalar values.

Parameter Separation and Relation to Measurements. For the independent parameter blocks defined above,

$$|\Theta_{\text{train}}| = |\theta| + |\psi| + \sum_k |P_k| + |\phi|, \qquad |\Theta_{\text{deploy}}| = |\theta| + |\psi|. \tag{55}$$

Shared head parameters are counted once, and frozen speech or external language-model parameters are outside these trainable/retained neural counts. Deployment memory has no MOST alignment tables or privileged attention maps; total training memory additionally includes backbone activations, parameter gradients, optimizer states, and implementation-specific workspaces. Equal retained architecture implies the same neural computational structure. The bidirectional predictor still needs its sentence context, and the WFST, language rescoring, and fusion stages in Appendix B.5 remain outside these neural-only complexity and latency statements.