ALIGN: Adversarial Learning for Generalizable Speech Neuroprosthesis

Zhanqi Zhang[*1] **Shun Li**[*5,6,7] **Bernardo L. Sabatini**[5,6,7] **Mikio Aoi**[†2,4] **Gal Mishne**[†1,2,3]

[1]Computer Science and Engineering, UC San Diego [2]Halıcıoğlu Data Science Institute, UC San Diego
[3]Electrical and Computer Engineering, UC San Diego [4]Neurobiology, UC San Diego
[5]Kempner Institute, Harvard University [6]Harvard Medical School [7]Howard Hughes Medical Institute

Abstract

Intracortical brain-computer interfaces (BCIs) can decode speech from neural activity with high accuracy when trained on data pooled across recording sessions. In realistic deployment, however, models must generalize to new sessions without labeled data, and performance often degrades due to cross-session nonstationarities (e.g., electrode shifts, neural turnover, and changes in user strategy). In this paper, we propose **ALIGN**, a session-invariant learning framework based on multi-domain adversarial neural networks for semi-supervised cross-session adaptation. ALIGN trains a feature encoder jointly with a phoneme classifier and a domain classifier operating on the latent representation. Through adversarial optimization, the encoder is encouraged to preserve task-relevant information while suppressing session-specific cues. We evaluate ALIGN on intracortical speech decoding and find that it generalizes consistently better to previously unseen sessions, improving both phoneme error rate and word error rate relative to baselines. These results indicate that adversarial domain alignment is an effective approach for mitigating session-level distribution shift and enabling robust longitudinal BCI decoding.

1 INTRODUCTION

Intracortical brain–computer interfaces (BCIs) have recently achieved strong brain-to-text performance (Willett et al. [2023]), but a central challenge for real-world use is cross-session generalization. Neural recordings are nonstationary: the mapping from neural activity to intended output drifts across sessions due to electrode impedance changes, electrode drift, and neural turnover (Figure 1, Chestek et al. [2011], Perge et al. [2013], Downey et al. [2018]). As a result, decoders trained on one session can degrade on later sessions, requiring frequent recalibration and limiting long-term usability. For participants, each additional day of calibration is a substantial burden, adding clinical workload and reducing time available for everyday communication. Moreover, it is well known in neuroscience that biological systems naturally execute motor programs like vocalization/movement at diverse speeds with high temporal flexibility (Banerjee et al. [2024], Wang et al. [2018], Stroud et al. [2018]). Therefore, developing methods to mitigate such trial- and session-level variability is critical.

In machine learning, domain adaptation aims to mitigate session-dependent distribution shift by leveraging labeled source data together with unlabeled target data, typically by learning representations that are invariant across domains. Common approaches include adversarial objectives (Ganin et al. [2016], Zhao et al. [2018]) and importance reweighting (Borgwardt et al. [2006]). In neuroscience, related ideas have been used to align neural representations across days using adversarial training (Farshchian et al. [2019], Ma et al. [2023]), improve robustness to channel/unit turnover via masking (Feghhi et al. [2026]) or permutation-invariant population models (Le et al. [2026]), and explicitly match latent manifolds across sessions (Perge et al. [2013], Karpowicz et al. [2025]). Other work applies test-time adaptation (TTA) using pseudo-labels to update models online at deployment (Feghhi et al. [2026], Fan et al. [2023]).

However, extending these approaches to brain-to-text decoders is substantially harder. While most of these methods are evaluated on continuous decoding tasks with dense supervision, phonemes/words are discrete categorical outputs, with no ground truth label for each neural time step because supervision is only available at the sequence level (i.e., the prompted sentence), and there is no overt behavior that can be time-aligned to neural activity.

As a result, training must rely on sequence-level objectives for alignment rather than per-frame targets, and decoder out-

[*]Equal contribution. [†] Equal contribution.

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:8006–8025.

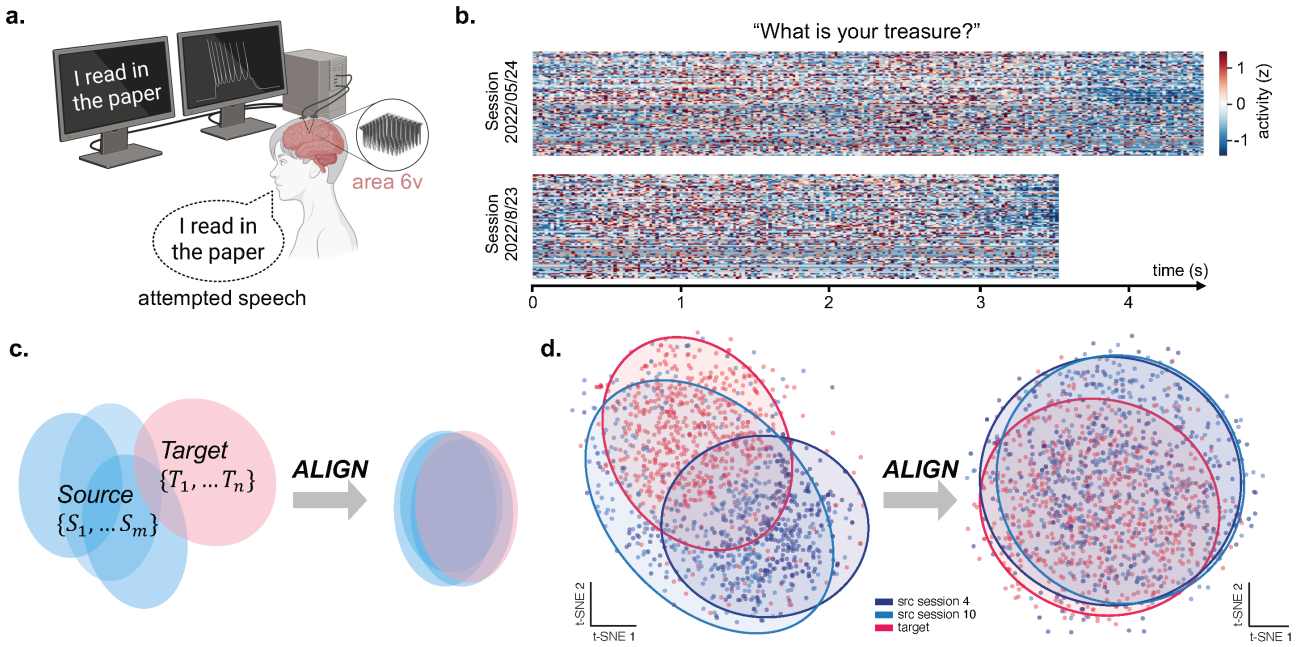

Figure 1: **Dataset and domain shift across sessions. a.** Attempted speech dataset. Intracortical neural activity was recorded from a participant with amyotrophic lateral sclerosis (ALS) while they attempted to speak sentences presented on a screen. Although vocalizations were produced, the speech was not intelligible. **b.** Example neural activity corresponding to the same attempted sentence recorded in two different sessions, illustrating session-dependent neural dynamics variabilities. **c.** Illustration of distribution shift in source sessions and target sessions caused by nonstationarities in neural dynamics and the session-invariant distributions with adaptation. **d.** t-SNE visualization of the latent embeddings used for phoneme decoding (input to the phoneme classifier), showing source sessions (blue) and target sessions (red) before and after ALIGN.

puts are typically composed with a language model whose prior is strong but imperfect and therefore cannot reliably recover from all classes of phonetic errors. Moreover, in brain-to-text settings, test-time adaptation (TTA) via self-training can be brittle under large distribution shifts: once the phoneme error rate crosses a threshold, language-model-based pseudo-labels can become systematically wrong and errors can compound, especially when the test session is far from the last labeled training session (Fan et al. [2023]). These challenges motivate the need for models that learn session-invariant neural representations for intracortical brain-to-text decoding and generalize robustly to unseen data.

In this paper, we propose a new approach **ALIGN**, (**A**dversarial **L**earn**I**ng for **G**eneralizable Speech **N**europrostheses), for brain-to-text decoding robust to session drifts, building on domain adaptation (Figure 1). Our main contributions are:

- A multi-source adversarial session-invariance objective for intracortical brain-to-text decoding that aligns session representations to promote learning of session-invariant features.
- An intermediate-layer adversarial regularization strategy that promotes day-invariant yet phoneme-discriminative features for decoding.
- Temporal Stretch Augmentation (TSA) to improve robustness to temporal nonstationarities.

2 RELATED WORK

Brain–computer interfaces (BCIs) aim to restore communication and motor function by translating neural activity into intended actions or language (Metzger et al. [2023], Moses et al. [2021], Willett et al. [2023], Card et al. [2024]). This is particularly important for people with severe paralysis (e.g. due to amyotrophic lateral sclerosis or brainstem stroke), for whom neuroprostheses could restore communication by converting neural signals into words. However, a major challenge for chronic BCI deployment is that neural decoding performance can degrade as recording conditions and neural representations drift over time (Sussillo et al. [2016], Driscoll et al. [2022], Pun et al. [2024]). Models that remain stable across days can therefore reduce the need for frequent decoder recalibration (Wilson et al. [2025]).

Brain-to-text decoding is a BCI setting that aims to infer a person's intended speech directly from recorded neural activity, without direct reference to the musculoskeletal components of vocal production. Willett et al. [2023] demonstrated strong performance with a Gated Recurrent

Unit (GRU) based model to decode phonemes from neural activity, followed by beam search guided by an n-gram language model (LM) to convert phoneme to words; Card et al. [2024] extended this to another participant. More recent work Feghhi et al. [2026] explores replacing the GRU with transformer decoders (e.g. relative positional embeddings, temporal patches, and masking strategies) or using pre-trained end-to-end frameworks (Zhang et al. [2026]).

In contrast to prior work that primarily addressed online in-session decoding and offline in-session decoding, we directly study cross-session generalization for brain-to-text decoding, for which the closest related efforts are the test-time adaptation (TTA) approaches such as those in Feghhi et al. [2026], Fan et al. [2023]. However, TTA does not learn a generalizable embedding during training and instead relies upon post-hoc adaptation. In addition, TTA methods that rely on pseudo-labels are sensitive to pseudo-label quality: if the pseudo-label accuracy is low, TTA recalibration accuracy would also degrade (Fan et al. [2023]). Under large session shifts, where the initial phoneme error rates are high, pseudo-labels may become systematically inaccurate, leading to runaway error compounding. These shortcomings highlight the need for methods that address distribution drifts.

Domain adaptation methods aim to mitigate distribution shift by combining labeled source data with unlabeled target data to learn representations that transfer across domains. Classic approaches include adversarial objectives that encourage domain-invariant features (Ganin et al. [2016], Zhao et al. [2018]), as well as discrepancy and re-weighting methods such as Maximum Mean Discrepancy (MMD)-based alignment (Borgwardt et al. [2006]). Specifically, the multi-source domain adaptation network (MDAN) framework (Zhao et al. [2018]) extends adversarial domain adaptation to multiple source domains by coupling a task predictor with a domain classifier and aggregating per-source discrepancies to learn features that generalize to an unlabeled target domain. This multi-source perspective is especially relevant in our setting, where session identity induces substantial shifts in neural feature distributions.

However, directly porting standard domain adaptation recipes to brain-to-text is nontrivial: many alignment objectives assume per-sample labels (or low-noise pseudo-labels) in the target domain and are mostly developed for continuous-output decoding, whereas speech decoding involves discrete symbols with only sentence-level supervision. Prior BCI work has pursued cross-session robustness through adversarial alignment with autoencoder-style objectives (Farshchian et al. [2019], Ma et al. [2023]), manifold alignment of latent dynamics (Karpowicz et al. [2025]), re-weighting techniques (Cowan and Linderman [2025]), and other approaches that align neural activity between a single source session and a single held-out target session. Beyond explicit alignment, permutation-invariant set-based encoders address changes in the recorded neural population (Le et al. [2026]). These methods are designed for continuous outputs (e.g., cursor position or EMG), thus not directly compatible with sentence-level CTC-based decoding.

Taken together, these limitations motivate a training-time approach that explicitly targets session-induced distribution shift while respecting the discrete, sequence-level supervision inherent to speech. In this work, we draw inspiration from multi-source domain adaptation frameworks like MDAN (Zhao et al. [2018]) to treat sessions as distinct source domains and to encourage session-invariant neural representations. Next, we introduce ALIGN, which is designed to provide such a foundation and enable more reliable test-time adaptation under large session shifts.

3 METHOD

The key intuition underlying our approach is to formulate cross-session brain-to-text decoding as a domain adaptation problem. Given labeled neural data from multiple source sessions and unlabeled data from subsequent target sessions, our objective is to learn a session-invariant representation that preserves speech-discriminative information while reducing session-specific encoding. The goal is to improve test-time generalization without the need for retraining or fine-tuning the decoder, which is burdensome to the participant. To this end, we introduce **ALIGN**, an adversarial multi-source single-target alignment framework built upon state-of-the-art neural decoders. ALIGN augments the base neural decoder with a domain classifier operating on latent features. Conceptually, by adversarially suppressing session-specific structure in the latent space, ALIGN retains speech-relevant information while mitigating sensitivity to session variability. We detail each component below. Code is available at github.com/ZhanqiZhang66/align.

3.1 NEURAL DECODER

Similar to established methods (Feghhi et al. [2026], Willett et al. [2023]), our neural decoder consists of two components: (1) a feature encoder f and (2) a phoneme classifier p. We schematically illustrate the ALIGN architecture in Figure 2. The feature encoder takes neural signals $\mathbf{X} \in \mathbb{R}^{c \times T}$ of c channels and length T, recorded during attempted speech and produces a sequence of latent features $z_t \in \mathbb{R}^D$, a D-dimensional latent embedding at each time t, that captures the relevant information for decoding phonemes. The phoneme classifier maps the latent sequence $\{z_t\}_{t=1}^{T}$ through a linear softmax layer to estimate a distribution over phonemes at each time step.

Let y be the ground-truth phoneme sequence corresponding to the spoken sentence in a single trial $x \in \mathbf{X}$. We train the encoder and phoneme classifier with the connectionist

temporal classification (CTC) loss (Graves et al. [2006]):

$$L_{\text{CTC}}(\theta) = -\log P_\theta(y \mid x) = -\log \sum_{\pi \mapsto y} \prod_{t=1}^{T} P_\theta(\pi_t \mid x),$$

where $P_\theta(\pi_t|x)$ are the probabilities of frame-level phoneme label π_t at time t given by the softmax over the encoder's output $z_t = f(x_t; \theta_f)$, and $\theta = (\theta_f, \theta_p)$, where θ_f and θ_p denote the parameters of the encoder and phoneme classifier, respectively. The CTC loss marginalizes over all valid frame-level alignments between neural activity and the target phoneme sequence. This is essential to enabling decoding in the BCI setting where precise temporal alignment between neural signals and phoneme onsets is unavailable, since there is no overt behavior.

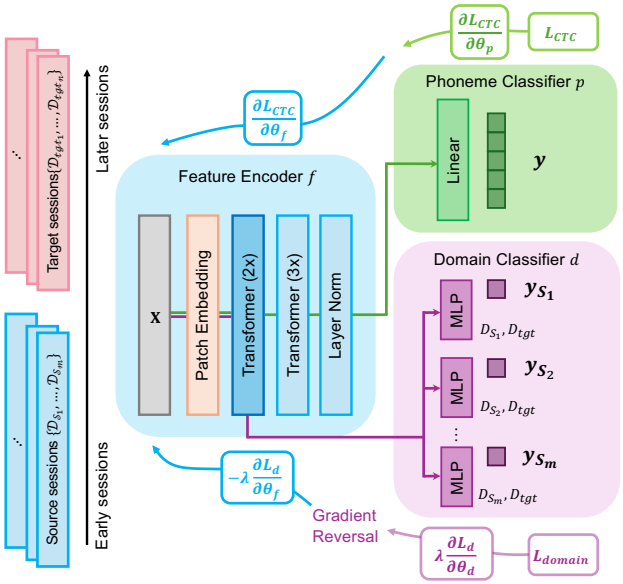

Figure 2: **ALIGN model architecture.** Our ALIGN model consists of three major modules. The feature encoder f and phoneme classifier p backbone shown here is instantiated with a Transformer-based decoder (Feghhi et al. [2026]). The "Transformer (2× / 3×)" blocks denote stacked transformer layers (i.e., repeated blocks), which together form the 5-layer transformer model. We added a domain classifier d, which takes the transformer encoder embedding as input and the multi-head classifiers are trained to predict whether the input is coming from source or target sessions. A gradient reversal layer is used to make the encoder learn session-invariant features.

3.2 ADVERSARIAL SESSION INVARIANCE

Neural recordings collected across sessions exhibit distribution shifts, even though the phoneme prediction decoding task remains unchanged. We therefore treat earlier labeled sessions as source domains and subsequent sessions as target domains, framing cross-session decoding as a domain adaptation problem. To promote generalization to target sessions and unseen test sessions, ALIGN introduces an adversarial domain classifier on the feature encoder's latent features.

Formally, the adversarial domain classifier $d(z_d; \theta_d)$ is constructed as a multi-head binary classifier with m heads, one for each source session, where θ_d are the parameters of the domain classifier and z_d denotes the intermediate encoder representation passed to the domain classifier. Each head i is trained to distinguish embeddings from source session $\mathcal{D}_{S_i}$ versus embeddings from all target sessions. Samples from $\mathcal{D}_{S_i}$ are routed only to head i, while samples from the target sessions are routed to all heads. Each head outputs a binary prediction $y_{S_i} \in \{0, 1\}$ (source vs. target) using a three-layer multilayer perceptron (Figure 2). We train each domain classifier head with binary cross-entropy classification loss and take the mean across all m classifier heads. The overall domain loss is computed as the mean across all m heads

$$\mathcal{L}_{\text{domain}}(\theta_d) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}^{(i)}_{\text{session}}(\theta_d). \tag{1}$$

While we optimize θ_d to *minimize* L_{domain}, we simultaneously optimize the encoder parameters θ_f to *maximize* this loss with a gradient reversal layer (GRL, Ganin and Lempitsky [2015]) during each training step. At test time, the domain classifier is inactive, and phoneme predictions are generated by the neural decoder, and then passed to the LM.

3.3 ALTERNATIVE SCHEDULING

The total loss is given by

$$\mathcal{L} = \mathcal{L}_{\text{CTC}} + \alpha\, \lambda\, \mathcal{L}_{\text{domain}}.$$

where λ determines the relative importance of the domain objective in the total loss, controlling the global strength of domain alignment throughout training.

During the forward pass, the GRL acts as the identity mapping between f and d. During backpropagation, it multiplies the gradient flowing from the domain loss into the feature encoder by $-\alpha\lambda$. Therefore, the gradient update on the encoder parameters become:

$$\frac{\partial \mathcal{L}}{\partial \theta_f} = \frac{\partial \mathcal{L}_{\text{CTC}}}{\partial \theta_f} - \alpha\lambda \frac{\partial \mathcal{L}_{\text{domain}}}{\partial \theta_f}.$$

The phoneme and domain classifiers receive gradients:

$$\frac{\partial \mathcal{L}}{\partial \theta_p} = \frac{\partial \mathcal{L}_{\text{CTC}}}{\partial \theta_p}, \qquad \frac{\partial \mathcal{L}}{\partial \theta_d} = \lambda \frac{\partial \mathcal{L}_{\text{domain}}}{\partial \theta_d}.$$

We use α to modulate the magnitude of the reversed gradient passed at each step, governing how strongly adversarial

signals influence the feature encoder. Let s be the training step, and s_{total} be the total number of training steps then:

$$\kappa = \min\left(\frac{s}{s_{\text{total}}}, 1\right),$$

denotes normalized training progress. We applied a sinusoidal scheduling $\alpha(\kappa, \omega)$

$$\alpha(\kappa, \omega) = \frac{1}{2} + \frac{1}{2}\sin\left(\omega\,\pi\,\kappa - \frac{\pi}{2}\right).$$

Thus, $\alpha(\kappa, \omega)$ is a sinusoidal function whose frequency is controlled by the ω parameter. This scheduling strategy allows the model to alternatively prioritize learning phoneme-discriminative structure and progressively enforcing domain invariance. Together, λ sets the scale of domain supervision, while $\alpha(\kappa, \omega)$ dynamically regulates its effect on representation learning over the course of training.

This two-player setup eventually converges to an equilibrium where z_t contains minimal session information and the encoder cannot further improve by altering z_t without hurting CTC performance. See Appendix K the full set of hyperparameters.

3.4 TEMPORAL STRETCH AUGMENTATION

To address variability in the duration of equivalent sequences across both repetitions and sessions observed in the data (Figure 1b), we developed Temporal Stretch Augmentation (TSA) . For each neural feature sequence $\mathbf{X} \in \mathbb{R}^{c \times T}$, we simulate trial-to-trial variability in the temporal dimension by generating a stretched version of the sample. Specifically, for each sample, a stretch factor r is drawn from a predefined range $[r_{\min}, r_{\max}]$, and the sequence is rescaled to length $T' = \lfloor rT \rfloor$. The resampled sequence $\widetilde{\mathbf{X}} \in \mathbb{R}^{c \times T'}$ is obtained by linear interpolation along the temporal axis while preserving feature dimensionality. See Appendix D for more details. TSA simulates natural variations in speech rate and neural timing without altering phoneme labels.

3.5 TEST-TIME ADAPTATION

We used DietCORP proposed in Feghhi et al. [2026] for test-time adaptation (TTA) with two initializations: (i) TTA from the first *target* session, where we first adapt on the validation sessions stream and use the resulting weights as initialization for test sessions; and (ii) TTA from the first *test* session, where adaptation begins from the neural decoder checkpoint trained on all source sessions, at the first test trial.

Specifically, for each trial, we first feed the neural decoder's CTC logits into the 3-gram language model to obtain an LM-refined pseudo-transcription, which is then converted to a sequence of phoneme pseudo-labels. We then adapt the decoder online by minimizing the CTC loss to these pseudo-labels. For each trial, we generate $n = 64$ augmented copies using additive white noise and baseline shifts and perform one gradient step, carrying the updated weights forward to subsequent trials, following the original DietCORP adaptation procedure in Feghhi et al. [2026].

4 EXPERIMENTS

4.1 DATASETS

We evaluated ALIGN on two intracortical speech BCI datasets from participants with Amyotrophic Lateral Sclerosis (ALS), T12 (Willett et al. [2023]), and T15 (Card et al. [2024]). Both datasets consist of multi-session recordings from four 64-channel micro-electrode arrays (256 electrodes total) implanted in the cortical speech motor network (ventral and premotor cortex for T12; ventral premotor cortex, area 6v, area 55b, and primary motor cortex for T15). Neural activity is represented as binned features every 20 ms (256 features for T12; 512 for T15). T12 contains 10,850 sentences collected over 24 sessions ($\sim$4 months), and T15 contains 10,948 sentences collected over 45 sessions ($\sim$20 months). Each trial includes neural time-series features paired with sentence-level transcriptions and phoneme sequence labels. Full dataset details are provided in Willett et al. [2023], Card et al. [2024].

The original T12 and T15 train/validation/held-out dataset partitions are split per session (Figure 6, left). For example, in T12, the validation partition (referred to as "test" in Willett et al. [2023]) contains the last block of each day with 40 sentences; the held-out partition (referred to as "competitionHoldOut" in Willett et al. [2023]) contains the first two blocks with 80 sentences; and "train" contains the rest. In T15, approximately 90% of each session's data was used for training and validation, and 10% was used for held-out.

For cross-session generalization experiments, we restructured the train/validation/test partition as follows: m source sessions (domains) $\{\mathcal{D}_{\text{src}_1}, \ldots, \mathcal{D}_{\text{src}_m}\}$, n subsequent unlabeled target sessions $\{\mathcal{D}_{\text{tgt}_1}, \ldots, \mathcal{D}_{\text{tgt}_n}\}$, and ℓ subsequent held-out test sessions $\{\mathcal{D}_{\text{test}_1}, \ldots, \mathcal{D}_{\text{test}_\ell}\}$, where each session corresponds to a separate recording day (Figure 6, right). There are gap days between recording sessions (i.e., sessions are not consecutive days). For both T12 and T15, we combine the train and validation data from source days as our training set, and the train and validation data from target days as our validation set. We use the unlabeled (i.e. no phoneme labels) held-out data from all target days as the target set, and we use the combined original training and validation data from test days as our test set.

We evaluate our models on both T12 and T15 using several session-level partitions to assess cross-session generaliza-

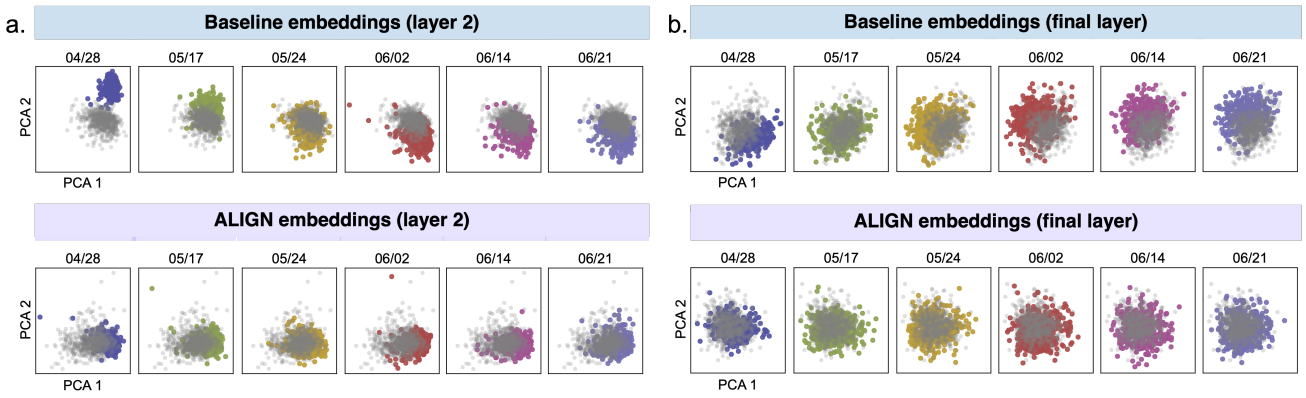

Figure 3: **Embedding visualization before and after ALIGN.** (**a**) PCA projections of the intermediate latent embeddings produced by the feature encoder in the transformer baseline decoder for T12 (top) and in ALIGN (bottom). Twelve source sessions are shown in color, along with three target sessions shown in gray. (**b**) Corresponding embeddings of final latent embeddings in both model.

tion. For T12, we consider three partitions: (a) 12 source, 8 target, 3 test (referred to as 12–8–3), i.e. 12 sessions are used for training, 8 sessions for validation (target-domain tuning), and the remaining 3 sessions for testing; (b) 15 source, 5 target, 3 test (referred to as 15–5–3); and (c) 12 source, 4 target, 7 test (referred to as 12–4–7). The first two partitions match those of Feghhi et al. [2026]. For T15, we consider 5 source, 13 target, 4 test. If a source session contains only training data (i.e., no validation or test partition), we use this data as the source data.

In addition, all models use standard preprocessing, including per-session z-scoring of neural features, applied consistently across all baselines and ALIGN.

4.2 EVALUATION METRICS

Phoneme error rate (PER) and word error rate (WER) are used to evaluate model decoding accuracy. PER is the number of phoneme errors (i.e., insertions, deletions, and substitutions) from all sentences over the total number of ground truth phonemes, computed directly from the output of the neural decoder after duplicated phonemes were removed. PER serves as the raw measurement of the decoding performance and validation PER was used for model selection. By feeding phoneme probabilities from the neural decoder into an n-gram language model, we can obtain the sequence of words forming a likely natural language sentence and compute the WER with respect to the ground-truth transcript of the attempted speech.

4.3 BASELINES

GRU-based Neural Decoder The GRU decoder is a recurrent neural network that maps time-varying neural activity to phoneme sequences (Willett et al. [2023], Card et al. [2024]). It first applies a day-specific transformation to normalize session-dependent variability in neural signals. The transformed features are then processed by a stack of gated recurrent units (GRUs), which capture temporal dependencies in the neural activity. The final recurrent representation is passed through a linear layer to produce frame-level phoneme logits, and the model is trained using CTC loss to align neural activity with phoneme sequences without requiring explicit frame-level labels. Data augmentation includes additive white noise, baseline shift, random temporal cuts, and Gaussian smoothing to improve robustness across sessions. We adopt the same augmentation scheme during training. The GRU decoder serves as baseline for both T12 and T15.

Transformer-based Neural Decoder We used the Transformer decoder of Feghhi et al. [2026] as an additional baseline for T12 and T15. While the official checkpoint trained on T15 was not available, we adapted the same Transformer-based decoder of Feghhi et al. [2026] to T15. The input neural time series of 256 channels (512 neural features for T15) was padded and Gaussian-smoothed, then partitioned into non-overlapping temporal patches of length 5 (overlapping temporal patches of length 14 with stride 4 for T15). Each patch ($\mathbb{R}^{256\times5}$ for T12; $\mathbb{R}^{512\times14}$ for T15) was flattened and mapped to a $m = 384$-dimensional embedding with LayerNorm, a linear projection, and a second LayerNorm over m, yielding a patch sequence that was processed by a 5-layer Transformer with causal self-attention. During training, patch-level time masking was applied (disabled for T15). A linear projection produces phoneme logits per output timestep for CTC-based phoneme decoding.

4.4 MULTI-SESSION GENERALIZATION

For all experiments, we performed model selection and hyperparameter tuning based on validation PER.

T12 evaluation: For participant T12, ALIGN achieved better PER across target sessions compared to the transformer baseline (Figure 5 in Appendix B), yielding an average session improvement of approximately 9% validation PER on the 12-8-3 split. PCA projections of source and target representations of the baseline model at earlier transformer layers exhibited more noticeable session drift (Figure 3a, top) than deeper layers (Figure 3b, top). We therefore applied the adversarial invariance loss to the intermediate layers (Figure 2), which yielded the most effective drift-robust decoding performance. Applying ALIGN resulted in more spatially overlapped embeddings between each source session and the target sessions, indicating that the learned representations were less session-dependent as shown in Figure 3 (bottom rows) for T12 partition 12-8-3.

We also compared ALIGN against the baseline models for unseen test sessions using word error rate (WER). Because the transformer-based decoder in Feghhi et al. [2026] achieves lower WER than the GRU baseline, we focus our T12 comparisons on the transformer model. We evaluated both models with and without test-time adaptation (TTA) for 5 random seeds. We evaluated three different train–test partitions and reported results with and without TTA (Figure 4). Across partitions, ALIGN consistently improved WER relative to the transformer baseline in both settings. Without TTA, ALIGN reduced test-session WER compared to both baselines (GRU: $65.93 \pm 0.28\%$; Transformer: $60.01 \pm 1.44\%$; ALIGN: $46.50 \pm 0.46\%$ on the first held-out test session in the 12–4–7 partition, with consistently lower WER on subsequent test sessions), suggesting that the adversarial objective promotes better generalization to held-out test sessions.

Test-time adaption: With TTA, ALIGN further improved test-session WER and, importantly, yielded greater stability across sessions than the transformer baseline with TTA (Figure 4). When TTA started from the first target day, the transformer baseline remained consistently worse than ALIGN across partitions and test sessions (Figure 4, left). For two of the three partitions, the transformer baseline's WER rose sharply on later test sessions, while WER of ALIGN showed around 65% absolute improvement at the last test session. This pattern suggests that, under accumulating session drift, language-model pseudo-labels and self-training can become unreliable and may fail to correct, or even amplify, decoder errors, whereas ALIGN provides a more robust starting point for adaptation.

The benefit of ALIGN is even clearer when TTA started from the first *test* session rather than the first target session. Because test sessions took place after target sessions, there was a comparatively larger distribution shift between the last labeled training day and the first day used for TTA, making self-training more challenging. Consistent with this, the transformer baseline with TTA initially performed moderately on the first test session $55.68 \pm 0.83\%$ but rapidly degraded (Figure 4, right), reaching higher than 90% WER from the second test session onward. On the other hand, WER of ALIGN with TTA remained lower, tightly bounded between 29% and 33% across days with a low overall WER of $32.23 \pm 0.35\%$, without exhibiting a late-session collapse. These results suggest that ALIGN is more resilient to cross-session variability and can be effectively paired with modern TTA to maintain performance under larger session shifts.

Across the three T12 partitions, performance consistently degraded as the temporal gap between the final training session and the held-out test sessions increased. In the challenging 12–4–7 partition, only four target sessions are available for ALIGN, and the seven test sessions spanned a larger temporal distance from the last training day. The reduced number of intermediate target days limited the model's ability to gradually track session drift, resulting in larger performance drops (i.e. higher WER) for the baseline models. ALIGN mitigated this degradation by learning a more session-invariant latent representation, yielding consistently lower WER.

T15 evaluation: We evaluated performance on T15 using the most challenging setup: 5 training sessions, 13 target sessions, and 3 held-out test sessions. This configuration contained the fewest training days and the largest temporal gap between the final training session and the latest test session, spanning up to 86 days. The limited labeled training data and substantial cross-session drift over time made this partition particularly demanding. ALIGN improved over the GRU baseline in test WER. When TTA was initialized from the first test session, the GRU baseline degraded on later sessions, whereas ALIGN remained lower than baseline across time (Table 1).

Table 1: **T15 Test Word Error Rate (WER).** TTA was performed from the first test session. 5 seeds were run for each condition. Test WER is shown as mean ± std.

Days from last train	GRU Baseline	ALIGN (GRU)
71	51.28 ± 0.95	$\mathbf{45.50 \pm 0.00}$
84	86.71 ± 2.29	$\mathbf{65.65 \pm 0.19}$
86	99.11 ± 0.11	$\mathbf{71.79 \pm 0.68}$

We also adapted the Transformer-based decoder used in T12 to T15. ALIGN model based on transformer also significantly improves PER from the adapted transformer baseline (Table 3 in Appendix I). This result demonstrates that the performance gain in decoding achieve through ALIGN is consistent across different base structures of the model and different datasets.

4.5 ALIGN MODEL ABLATIONS

We tested the importance of each of ALIGN's components by removing (1) adversarial loss, (2) temporal stretch aug-

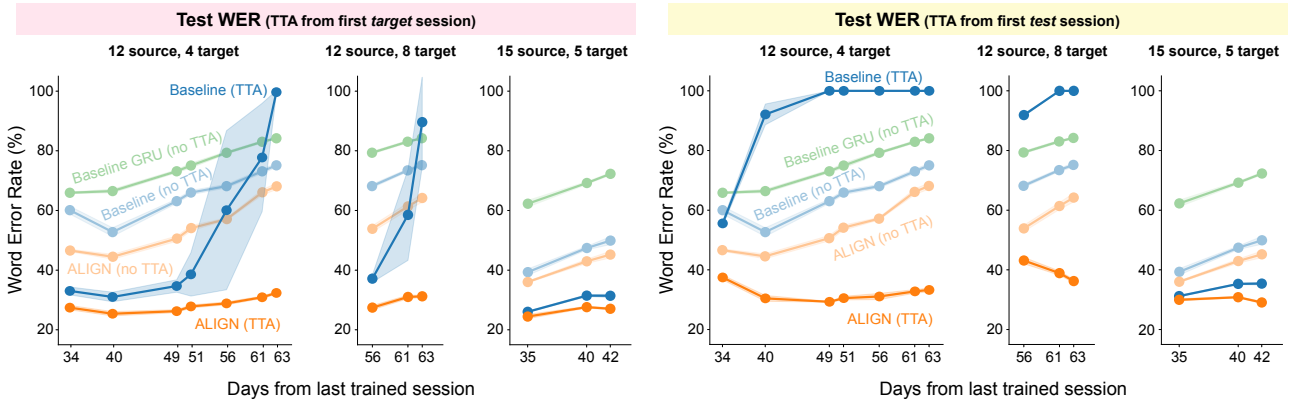

Figure 4: **ALIGN with and without TTA.** Test WER of GRU baseline (green) without TTA, transformer baseline (blue) and ALIGN model (orange) with and without TTA (dark and light shades), where TTA is trained from the first target session (left) or from the first test session (right). All models are tested on three different train-test partition for 5 seeds.

mentation, and (3) alternative scheduling while keeping the hyperparameter unchanged, and presented these results in the T12 12–4–7 setting in Table 2.

Removing the adversarial loss altogether drastically degraded validation PER. This result supports the core hypothesis of ALIGN: explicitly discouraging session-identifiable structure in the learned representation is necessary for robust cross-session generalization. Without the adversarial loss, the decoder can achieve low training error by exploiting session-specific features (e.g., electrode- or day-dependent statistics), but these cues do not transfer to new sessions, resulting in higher PER.

Removing alternative scheduling moderately decreased performance, as shown by increased PER with lower variance. This suggests alternative scheduling may provide a modest benefit by balancing phoneme-discriminative learning and session-invariant alignment, helping the model to avoid a stable but slightly suboptimal solution.

Lastly, removing temporal stretch augmentation degraded validation PER. This indicates that TSA provides a complementary robustness beyond adversarial alignment. By expanding the temporal variability observed during training, TSA reduces reliance on narrow timing statistics of the source sessions and encourages representations that are stable under realistic temporal distortions.

To further examine the effect of TSA, we detached the adversarial loss and trained the transformer baseline decoder in Feghhi et al. [2026] on (i) data without TSA and (ii) with TSA applied. To evaluate whether model trained with TSA can be more robust to temporal variability (Figure 1b), we computed Wasserstein distance (WD) between source–target and validation–target embedding pairs on temporally-stretched trials. The TSA-trained decoder consistently produced embeddings that remained closer to the target distribution (lower WD across stretch factors) than the same model

Table 2: **Ablation study on validation phoneme error rate.** Models are trained with 12 source and 4 target sessions. Validation PER are shown in mean ± std across 4 seeds. p-values are obtained from pair t-tests against ALIGN.

Ablation	Validation PER (%)	p-value
ALIGN	$\mathbf{26.08 \pm 0.33}$	–
w/o adversarial loss	31.95 ± 0.42	6.85×10^{-6}
w/o TSA	27.70 ± 0.67	0.0392
w/o alt scheduling	26.38 ± 0.09	0.1086

trained without TSA (Appendix D, Figure 8). Therefore, temporal stretch augmentation likely acts as an implicit regularizer that encourages temporal invariance in the learned features, making the representation less sensitive to trial-wise speed fluctuations and reducing the mismatch between training and evaluation conditions.

4.6 ALIGN MODEL ANALYSIS

Discriminator: In ALIGN, we use K independent binary domain discriminators, one for each source-target session pair, instead of a single multi-class domain classifier. To validate this choice, we compare our K-binary-head discriminator against two alternatives: a multi-class domain classifier and a shared conditional binary discriminator $D(x, i)$, where a single MLP is conditioned on the session index through a learned embedding. As shown in Table 4 in Appendix J, our design achieves the lowest validation PER.

Adversarial layer: In Section 4.4, we mentioned that ALIGN applies the adversarial invariance loss to the intermediate layer (Layer 2), since PCA projections of source vs target representations of the baseline model displays more drift in those layers (Figure 3). To test this quantitatively, we compared validation PER of ALIGN models with different

choices of adversarial layer depth. Applying the adversarial loss at an intermediate layer yields the best target-day validation PER, supporting our design choice (Table 5 in Appendix J).

Phoneme discrimination: To examine whether ALIGN affects phoneme prediction, we performed a fine-grained analysis of phoneme-level errors on the target-day validation set using confusion matrices and per-category phoneme error rate (PER). As shown in Figure 10 and detailed in Appendix H, the adversarial objective does not suppress phoneme-relevant information. The confusion matrices retain a strong diagonal structure, per-category PER remains balanced across phoneme classes, and stop consonants achieve PER comparable to the overall model performance, with only a localized increase for affricates. In addition, the most frequent errors remain phonetically local, rather than showing broad collapse across phoneme categories. Together, these results suggest that ALIGN improves cross-session generalization while preserving the phonetic structure needed for accurate decoding.

Source session amount: To test whether ALIGN requires a large number of source sessions for effective alignment, we additionally evaluated T12 using only 5 source sessions (5-0-16). Even in this limited-source setting, ALIGN substantially improved validation PER over the baseline, reducing PER from $67.21 \pm 0.68\%$ to $51.67 \pm 1.44\%$ (across 3 seeds). This suggests that ALIGN remains effective in a 5-source regime and does not require a strict minimum number of source sessions. Rather, decoding performance depends jointly on the number of source sessions, the temporal gap, and the availability of intermediate target sessions.

Computation cost: Finally, ALIGN introduces negligible additional cost. During training, it adds M lightweight discriminator heads (one per source session), each implemented as a linear layer with $d+1$ parameters. For example, in the T12 setting ($M = 15$, $d = 256$), this corresponds to only $3{,}855$ additional parameters, which is negligible compared to the transformer encoder ($\sim$3.9M parameters, <0.1% of the total model size). The added computation scales linearly with the number of frames and heads and is small relative to transformer self-attention and feed-forward layers. At inference time, the discriminator is removed, resulting in zero additional parameters, FLOPs, or latency.

5 DISCUSSION

Our work addresses a practical challenge in intracortical speech BCIs: models trained on pooled data often degrade when deployed to a new recording session without labels. We propose a session-invariant learning framework that combines adversarial alignment across multiple source sessions with alignment applied at an intermediate decoder layer. This placement reduces session-discriminative structure while preserving information relevant for CTC-based phoneme decoding. Importantly, our approach operates in a few-shot, unsupervised setting on target sessions, requiring no phoneme labels during alignment.

Moreover, we introduced temporal stretch augmentation (TSA), which generates greater temporal diversity in neural speech sequences. Biologically, animals execute motor programs over a wide range of speeds (Banerjee et al. [2024], Wang et al. [2018], Stroud et al. [2018]), and biophysical properties of neurons can support downstream circuits to readout time-warped activity patterns as time-invariant representations (Gütig and Sompolinsky [2009]). Motivated by this, TSA increases temporal variability during training by stochastically time-stretching trials. Empirically, TSA complements the adversarial loss of ALIGN and further improves performance in terms of phoneme- and word-level decoding accuracy in unseen data.

We think that our methodological setting is not unique to BCI. It arises in sequence decoding problems where supervision is provided only at the sequence level and nuisance variation exists at finer timescale. This creates a mismatch not addressed by standard sample-level domain adaptation methods. ALIGN may provide a general way to couple adversarial domain alignment with sequence-level decoding objectives, solving other tasks such as text-decoding using wearable EMG or visual sign-language recognition.

We suggest several directions for future work. Decoding performance depends jointly on the phoneme decoding model and the language model, and tighter integration or a language-model-aware training objective could further improve accuracy. Leveraging self-supervised pretraining on large corpora of unlabeled neural activity may also reduce reliance on labeled data and enhance robustness across sessions. Alignment appears to improve cross-session robustness while largely preserving phoneme discrimination, with effectiveness influenced by factors such as source-session diversity, temporal gap, and the choice of alignment objective. Future work can further expand this analysis to better understand the conditions under which alignment is most beneficial. In addition, deeper analysis of what information is removed/preserved by alignment, such as identifying which neural dimensions are most session-specific and which phoneme errors are most impacted, could improve interpretability and guide more principled regularization.

Overall, our findings suggest that adversarial multi-session alignment, when applied at the right representational level and paired with temporally targeted augmentation, can mitigate session-level distribution shift and advance intracortical speech decoding closer to reliable long-term deployment, and reduce the need for frequent decoder recalibration.

Author Contributions

ZZ: Conceptualization, methodology, experiments, analysis, writing. SL: Methodology, experiments, analysis, writing. BS: Resources, supervision. MA: Conceptualization, supervision, writing. GM: Conceptualization, supervision, writing.

Acknowledgements

We thank collaborators and colleagues, especially Jingya Huang, Dr. Vikash Gilja for discussions and feedback. ZZ was supported in part by the Halıcıoğlu Data Science Institute Ph.D. Fellowship. This work was partially supported by NSF award EFRI 2223822 and a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence.

References

Arkarup Banerjee, Feng Chen, Shaul Druckmann, and Michael A Long. Temporal scaling of motor cortical dynamics reveals hierarchical control of vocal production. *Nature Neuroscience*, 27(3):527–535, 2024.

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

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

Cynthia A Chestek, Vikash Gilja, Paul Nuyujukian, Justin D Foster, Joline M Fan, Matthew T Kaufman, Mark M Churchland, Zuley Rivera-Alvidrez, John P Cunningham, Stephen I Ryu, et al. Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex. *Journal of neural engineering*, 8 (4):045005, 2011.

Noah Cowan and Scott Linderman. Sliced-wasserstein importance weighting for robust brain-computer interface speech decoding. In *Data on the Brain & Mind Findings*, 2025.

John E Downey, Nathaniel Schwed, Steven M Chase, Andrew B Schwartz, and Jennifer L Collinger. Intracortical recording stability in human brain–computer interface users. *Journal of neural engineering*, 15(4):046016, 2018.

Laura N. Driscoll, Lea Duncker, and Christopher D Harvey. Representational drift: Emerging theories for continual learning and experimental future directions. *Current Opinion in Neurobiology*, 76:102609, 2022. doi: 10.1016/j.conb.2022.102609.

Chaofei Fan, Nick Hahn, Foram Kamdar, Donald Avansino, Guy Wilson, Leigh Hochberg, Krishna V Shenoy, Jaimie Henderson, and Francis Willett. Plug-and-play stability for intracortical brain-computer interfaces: a one-year demonstration of seamless brain-to-text communication. *Advances in neural information processing systems*, 36: 42258–42270, 2023.

Ali Farshchian, Juan A. Gallego, Joseph P. Cohen, Yoshua Bengio, Lee E. Miller, and Sara A. Solla. Adversarial domain adaptation for stable brain-machine interfaces. *International Conference on Learning Representations*, 7, 2019.

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.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189. PMLR, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

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*, pages 369–376, 2006.

R. Gütig and H. Sompolinsky. Time-warp–invariant neuronal processing. *PLOS Biology*, 7(7):e1000141, 2009. doi: 10.1371/journal.pbio.1000141.

Brianna M. Karpowicz, Yahia H. Ali, Lahiru N. Wimalasena, Andrew R. Sedler, Mohammad Reza Keshtkaran, Kevin Bodkin, Xuan Ma, Daniel B. Rubin, Ziv M. Williams, Sydney S. Cash, Leigh R. Hochberg, Lee E. Miller, and Chethan Pandarinath. Stabilizing brain-computer interfaces through alignment of latent dynamics. *Nature Communications*, 16(1):4662, 2025.

Trung Le, Hao Fang, Jingyuan Li, Tung Nguyen, Lu Mi, Amy L Orsborn, Uygar Sümbül, and Eli Shlizerman. Spint: Spatial permutation-invariant neural transformer

for consistent intracortical motor decoding. *Advances in Neural Information Processing Systems*, 38:25815–25835, 2026.

Xuan Ma, Fabio Rizzoglio, Kevin L Bodkin, Eric Perreault, Lee E Miller, and Ann Kennedy. Using adversarial networks to extend brain computer interface decoding accuracy over time. *elife*, 12:e84296, 2023.

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

Mehryar Mohri, Fernando Pereira, and Michael Riley. Speech recognition with weighted finite-state transducers. In *Springer Handbook of Speech Processing*, pages 559–584. Springer, 2008.

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

János A Perge, Mark L Homer, Wasim Q Malik, Sydney Cash, Emad Eskandar, Gerhard Friehs, John P Donoghue, and Leigh R Hochberg. Intra-day signal instabilities affect decoding performance in an intracortical neural interface system. *Journal of neural engineering*, 10(3):036004, 2013.

Tsam Kiu Pun, Mona Khoshnevis, Tommy Hosman, Guy H. Wilson, Anastasia Kapitonava, Foram Kamdar, Jaimie M. Henderson, John D. Simeral, Carlos E. Vargas-Irwin, Matthew T. Harrison, and Leigh R. Hochberg. Measuring instability in chronic human intracortical neural recordings towards stable, long-term brain-computer interfaces. *Communications Biology*, 7(1):1363, 2024. doi: 10.1038/s42003-024-06784-4.

Jake P. Stroud, Mason A. Porter, Guillaume Hennequin, and Tim P. Vogels. Motor primitives in space and time via targeted gain modulation in cortical networks. *Nature Neuroscience*, 21(12):1774–1783, 2018. doi: 10.1038/s41593-018-0276-0.

David Sussillo, Sergey D. Stavisky, Jonathan C. Kao, Stephen I. Ryu, and Krishna V. Shenoy. Making brain–machine interfaces robust to future neural variability. *Nature Communications*, 7:13749, 2016. doi: 10.1038/ncomms13749.

Jing Wang, Devika Narain, Eghbal A. Hosseini, and Mehrdad Jazayeri. Flexible timing by temporal scaling of cortical responses. *Nature Neuroscience*, 21(1):102–110, 2018. doi: 10.1038/s41593-017-0028-6.

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

Guy H. Wilson et al. Long-term unsupervised recalibration of cursor-based intracortical brain–computer interfaces using a hidden markov model. *Nature Biomedical Engineering*, 2025. doi: 10.1038/s41551-025-01536-z.

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

Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31, 2018.

ALIGN: Adversarial Learning for Generalizable Speech Neuroprosthesis (Supplementary Material)

Zhanqi Zhang[*1] **Shun Li**[*5,6,7] **Bernardo L. Sabatini**[5,6,7] **Mikio Aoi**[†2,4] **Gal Mishne**[†1,2,3]

[1]Computer Science and Engineering, UC San Diego [2]Halıcıoğlu Data Science Institute, UC San Diego

[3]Electrical and Computer Engineering, UC San Diego [4]Neurobiology, UC San Diego

[5]Kempner Institute, Harvard University [6]Harvard Medical School [7]Howard Hughes Medical Institute

A BRAIN-TO-TEXT BENCHMARKING

We evaluated on the *Brain-to-Text Benchmark '24* (T12) Willett et al. [2023], and the *Brain-to-Text Benchmark '25* (T15) Card et al. [2024]. We apply the same preprocessing pipeline as the baseline neural decoders, following the procedures described in the original benchmark releases.

B PHONEME ERROR RATE

We selected models based on each dataset's validation phoneme error rate (PER). ALIGN consistently achieved the lowest PER across all validation and test sessions in the cross-session setting (Figure 5).

For the T12 12–8–3 partition, on the validation set (8 sessions), the Transformer baseline PER increased from $24.92 \pm 0.17\%$ (5 days from last training day) to $49.21 \pm 0.33\%$ (51 days), with a session-average of $38.74 \pm 0.43\%$. ALIGN ranges from $22.07 \pm 0.50\%$ to $37.14 \pm 0.88\%$, achieving a lower session-average of $30.46 \pm 0.52\%$, corresponding to an absolute improvement of over 8% points in PER. On the held-out test sessions (56–63 days away from last training), the Transformer baseline obtains a session-average PER of $52.61 \pm 0.24\%$, while ALIGN reduces this to $43.39 \pm 0.62\%$, yielding an improvement of approximately 9% absolute PER (Figure 5).

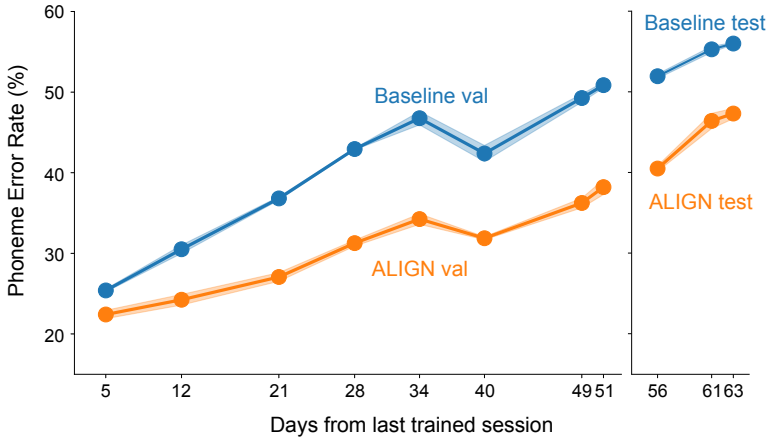

Figure 5: **Per-day phoneme error rate (PER).** Mean and std of PER of the transformer baseline (blue) and ALIGN (orange) on T12 12–8–3 dataset. The first eight sessions correspond to validation of target sessions, and the last three correspond to held-out test sessions.

These results demonstrate that ALIGN improves phoneme decoding accuracy and reduces cross-session performance degradation as temporal distance from the last training session increases. The reported PER reflects raw neural decoder

performance without test-time adaptation (TTA), directly measuring cross-session generalization of the learned representation. The consistently lower PER values therefore indicate improved cross-session robustness. Combined with prior findings that Transformer decoders outperform GRU baselines Feghhi et al. [2026], ALIGN achieves the best performance among the evaluated methods in the cross-session brain-to-text decoding setup.

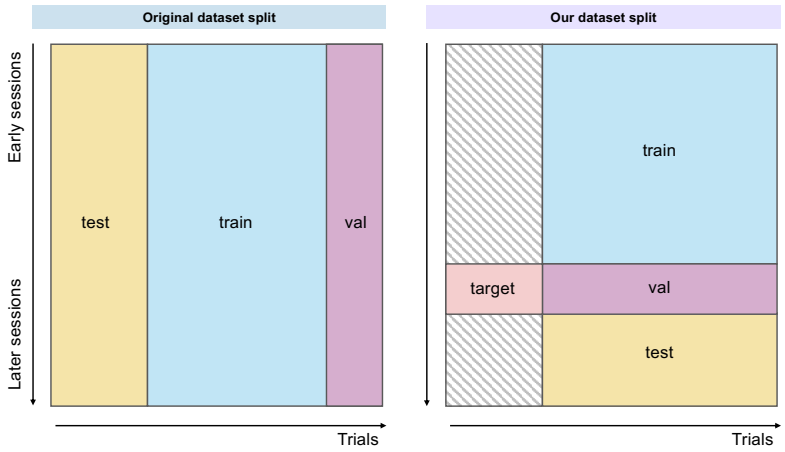

Figure 6: **Dataset Partition.** Left: In the official *Brain-to-Text Benchmark '24* partition, each recording day is internally divided into train, validation, and competitionHoldOut (test) blocks (except for a few sessions that lack a test block), so most sessions contribute data to every partition. Right: We reorganize data by session into source (train), unlabeled target (for unsupervised alignment), labeled validation (for model selection via validation PER), and fully unseen test sessions to evaluate cross-session generalization.

C ALIGN ON TARGET AND TEST DAYS

ALIGN leverages the distribution of the target sessions for domain alignment, but does not use any supervised CTC objective on target data. Specifically, target session neural data are used only to reduce session-related distribution shifts between source and target domains via the alignment loss. No target transcripts are used, and the CTC decoding objective is optimized exclusively on source sessions.

Consequently, the reported Phoneme Error Rate and Word Error Rate on the target validation sessions can also be interpreted as *adaptation results*, as the model is aligned to the target session distribution, but without supervised decoding on those sessions. In contrast, the unseen test sessions represent the strictest generalization and is closest to a real-world deployment setting, as neither alignment nor supervision is performed on those sessions. The model is adapted using target sessions only, and then directly evaluated on unseen test sessions.

In Figure 7, we report the full trajectory of Word Error Rate (WER) across both target sessions and unseen test sessions, spanning up to 63 days from the last trained source day in T12.

D TEMPORAL STRETCH AUGMENTATION

We used temporal stretch augmentation (TSA) as a complementary approach to the adversarial structure of ALIGN to further address data variability, specifically temporal variability at the trial level. TSA is applied per trial to the neural feature sequence (time x features). The sequence is resampled along the time axis using linear interpolation with a stretch factor.

The stretch factor used in this paper is always greater than 1 due to the CTC objective of ALIGN. CTC aligns a sequence of encoder frames (length T) to a sequence of labels (length L) using blanks and repeated labels. A valid alignment exists only if the input length T is at least as long as the target length L (i.e. $T \geq L$) to allow the encoder have enough time steps to "place" all labels (with optional blanks and repeats). Therefore, stretch factors less than 1 would violate such a constraint.

For each trial, a stretch factor is sampled from a specified range (e.g., $1.5 - 5.0\times$). During training, the data loader includes (1) the original trial and (2) a single time-stretched copy generated using a randomly sampled factor from this range. In summary, the model is trained on a mix of original and temporally warped trials to mimic trial-to-trial temporal variability.

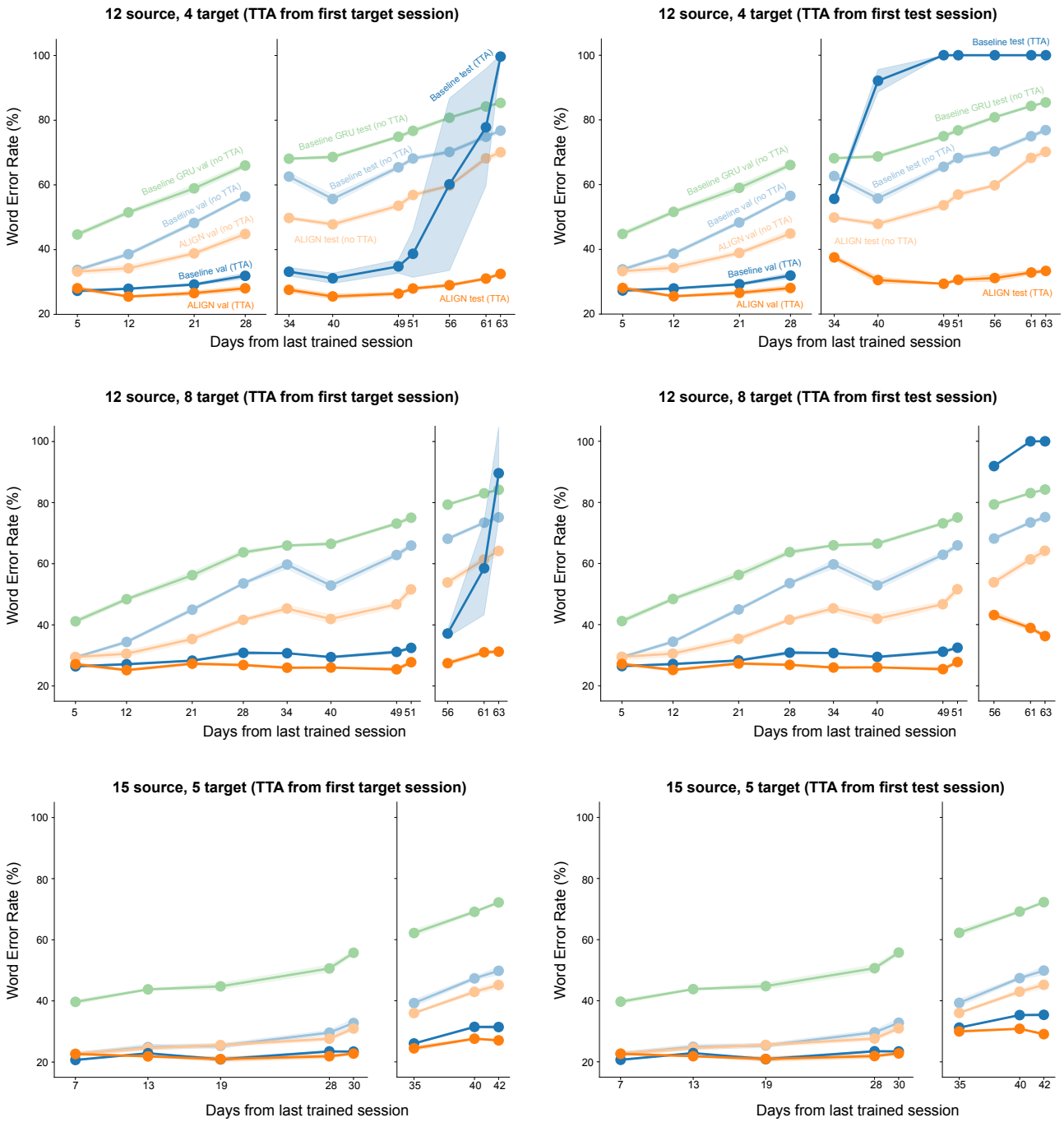

Figure 7: **ALIGN on target and test sessions.** Validation and test WER (%) of GRU baseline (green) without TTA, transformer baseline (blue) and ALIGN model (orange) with and without TTA (dark and light shades), where TTA is trained from the first target session (left) or from the first test session (right). All models are tested on three different train-test partition for 5 seeds.

E EMBEDDINGS OF ALIGN

Session-Distinct Embedding Dimension Analysis. To quantify session-specific variability in the learned representation, we measured the 1D Wasserstein distance between source and target distributions along each embedding dimension in the final latent space. Specifically, for each dimension $j = 1, \ldots, D$, we compute

$$W_j = W\big(z_j^{\text{src}}, z_j^{\text{tgt}}\big),$$

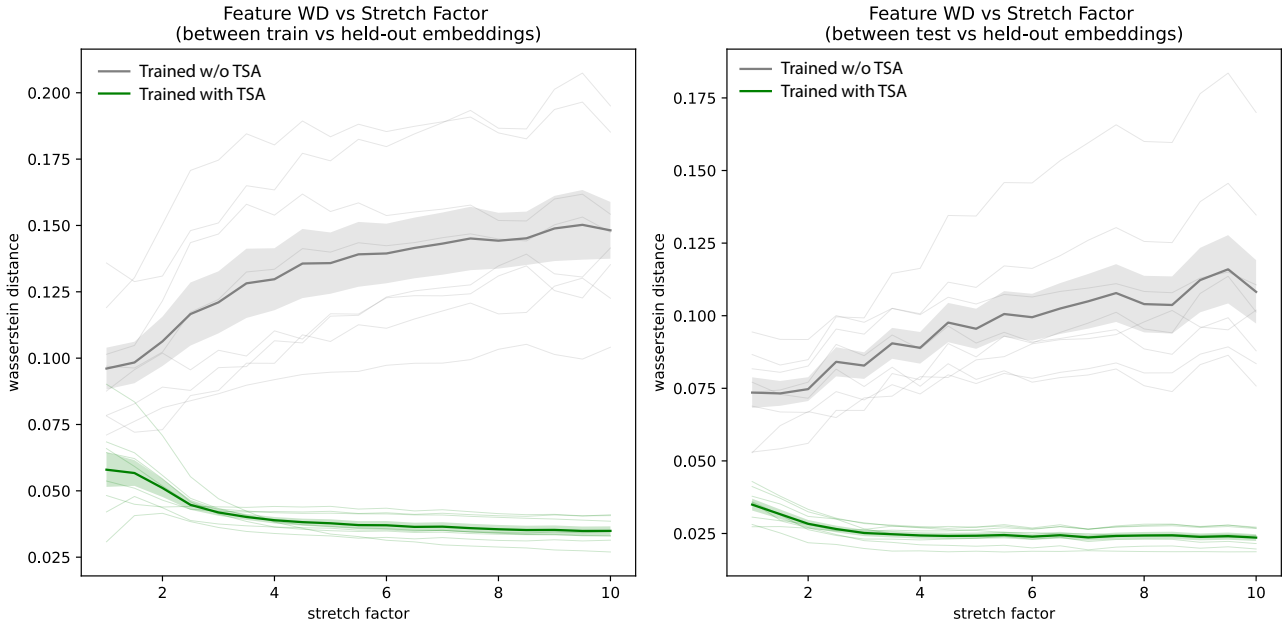

Figure 8: **Embedding analysis of TSA-trained decoder.** Feature-wise Wasserstein distance (WD) between augmented embeddings and a held-out set as a function of the stretch factor. Left: WD computed between source embeddings and target embeddings (average across the first 10 embedding dimensions); Right: WD computed between validation embeddings and target embeddings (average across the first 10 embedding dimensions). Thick lines show the mean across days and shaded bands denote $\pm$SEM; faint lines show individual day traces. Gray indicates the decoder trained on data without TSA, and green indicates decoder trained on data with TSA.

where z_j^{src} and z_j^{tgt} denote the marginal distributions of the j-th embedding coordinate for source and target sessions, respectively. We first rank dimensions in the transformer baseline from largest to smallest W_j, thereby identifying the most session-distinct components of the representation. We then compute the same distances after applying ALIGN. As shown in Figure 9, ALIGN aligned the activation distributions across all dimensions, indicating that it suppresses session-specific variability and promotes tighter alignment between source and target distributions in the final embedding space.

F LANGUAGE MODEL

The n-gram language model (LM) is trained on the OpenWebText2 corpus and constructed using a CMU-dictionary–based phoneme lexicon. As in Willett et al. [2023], Feghhi et al. [2026], Card et al. [2024], we use the 125,000-word 3-gram language model implemented in Kaldi and integrated through a Weighted Finite State Transducer (WFST) beam search decoder (Mohri et al. [2008]). The LM is loaded into a custom Python runtime and used for phoneme-to-word decoding.

During decoding, the neural decoder produces a sequence of frame-level phoneme posterior probabilities. Beam search is applied over these probabilities to generate multiple candidate hypotheses, where each candidate beam b corresponds to a complete phoneme sequence obtained after collapsing repeated labels and removing blanks.

For each candidate beam b, we compute a joint score by combining the encoder likelihood and the n-gram language model likelihood using an `acoustic_scale`:

$$\text{score}(b) = \texttt{acoustic\_scale} \log P_{\text{enc}}(b) + \log P_{\text{ngram}}(b). \tag{2}$$

Here, $P_{\text{enc}}(b)$ denotes the probability of the phoneme sequence under the phoneme classifier (CTC), and $P_{\text{ngram}}(b)$ is the probability assigned by the n-gram language model to the same phoneme sequence. The beam search is performed on the composed WFST graph, and the hypothesis with the highest combined score is selected as the final decoded transcription.

To discourage excessive blank emissions from the encoder, blank probabilities are penalized by dividing them by a fixed constant during decoding. All language model hyper-parameters matched those reported in Willett et al. [2023], Feghhi et al. [2026] for T12 and Card et al. [2024] for T15. Specifically, for the 3-gram language model on T12, we used a beam

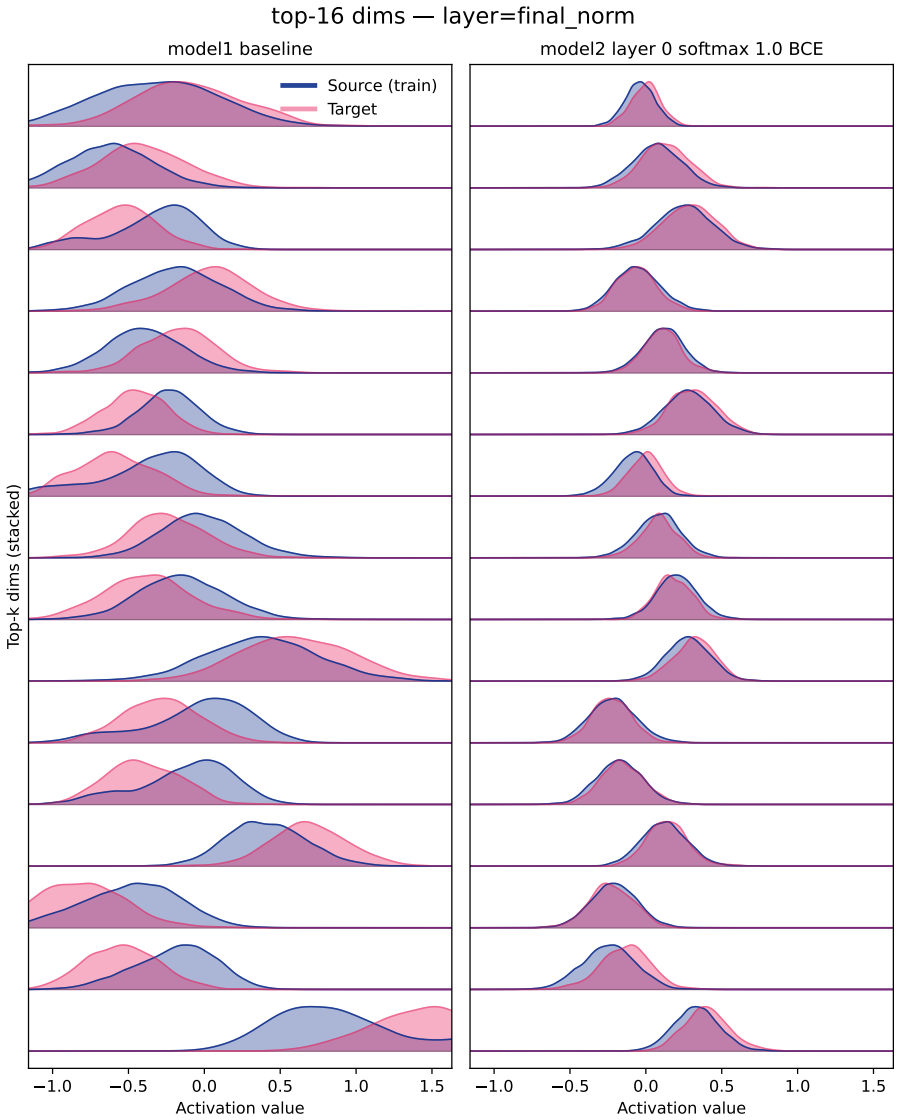

Figure 9: **ALIGN reduces the session distributional gap.** Top-k most session-distinct embedding dimensions before (left) and after ALIGN (right).

width of 18, set `acoustic_scale` $= 0.8$, and applied a blank penalty of $\log(2)$. For T15, we used a beam width of 17, set `acoustic_scale` $= 0.5$, and applied a blank penalty of $\log(9)$.

G ADVERSARIAL CLASSIFIER AND TTA IMPLEMENTATION DETAILS

Each classifier head is a multi-layer perceptron with two hidden layers of size $d_h = 256$, each followed by ReLU activation and dropout. For test time adaptation, because DietCORP was not evaluated on T15, we implement the same DietCORP adaptation logic while adopting data augmentation hyperparameters reported in Card et al. [2024]. The full hyperparameters for the models are provided below.

H PHONEME DISCRIMINATION ANALYSIS

We perform a fine-grained analysis of phoneme-level errors on the target-day validation set. The confusion matrices at convergence show a strong diagonal structure for both the baseline model and ALIGN, indicating that adversarial alignment

does not collapse phoneme discrimination. ALIGN reduces the overall phoneme error rate (PER) from 23.5% to 21.1%. When errors are grouped by phoneme category, ALIGN improves or maintains performance across phoneme classes, with stop consonants achieving PER comparable to the overall model performance. Affricates show relatively higher PER, suggesting some sensitivity to fine temporal cues; however, this effect remains localized and does not dominate overall performance. Finally, the most frequent directed confusions occur between acoustically or articulatorily similar phonemes, such as voicing-related stop consonants and nearby vowels. This suggests that the remaining errors are phonetically meaningful rather than reflecting broad information suppression.

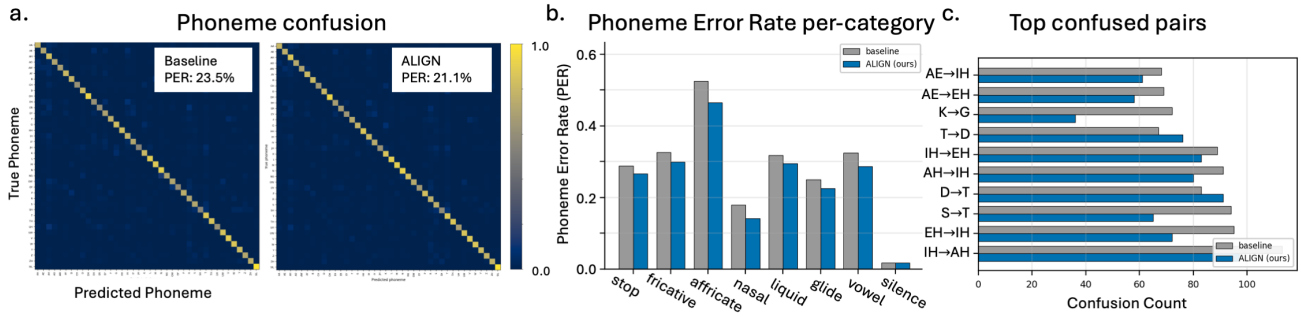

Figure 10: **Phoneme discrimination analysis of ALIGN. a.** Normalized phoneme confusion matrices for the baseline model and ALIGN at convergence. **b.** Phoneme error rate (PER) grouped by phoneme category. **c.** Most frequent directed phoneme confusions for the baseline model and ALIGN.

I T15 TRANSFORMER-BASED DECODER

Table 3: **T15 validation PER for transformer-based decoder.** Models are trained with 5 source and 8 target sessions. Validation PER is shown as mean ± std across 5 seeds. p-values are obtained from pair t-test against transformer baseline.

Method	Validation PER (%)	p-value
Transformer baseline	58.99 ± 0.52	–
ALIGN (transformer)	$\mathbf{46.36 \pm 0.62}$	2.51×10^{-8}

J ALIGN MODEL ANALYSIS

Table 4: **Ablation of domain discriminator design.** Models are trained on T12 with 12 source sessions and 4 target sessions. 4 seeds were ran for each condition. Validation PER are shown in mean ± std.

Discriminator design	Validation PER (%)
Multi-class domain classifier	32.89 ± 0.54
1 binary head & session input	29.59 ± 0.32
K binary heads (ours)	$\mathbf{26.30 \pm 0.52}$

Table 5: **Ablation study on adversarial layer placement.** Models are trained on T12 with 12 source sessions and 4 target sessions. 3 seeds were ran for each condition. Validation PER are shown in mean ± std.

Adversarial layer	Validation PER (%)
Early layer (Layer 0)	30.02 ± 0.24
Intermediate layer (Layer 2, ours)	$\mathbf{26.30 \pm 0.52}$
Final layer (Layer 4)	31.32 ± 0.43

K HYPERPARAMETERS

Table 6: Transformer Decoder Training and Domain Adaptation Hyperparameters

Parameter	Value
Optimizer	AdamW
Learning rate	0.001
Weight decay	1×10^{-5}
Batch size	64
Epochs	500
Scheduler	Multistep (milestone=300)
γ	0.1
Depth	5
Embedding dim	384
Heads	6
Dropout	0.35
Input dropout	0.2
Patch size	(5, 256)
Rep. layer index	2
λ_{DANN}	0.6
λ_{src}	1
λ_{tgt}	1
Disc. hidden dim	256
Disc. LR multiplier	0.6
Alpha type	Alternative ($\omega = 16$)
Binary loss	True
Target loss	True
Stretch range	(1.5, 5)
Max mask pct	0.075
Num masks	20
White noise SD	0.2
Baseline Shift	0.05
Gaussian smooth width	2
Test-time Adaptation learning rate	3e-3

Table 7: GRU Decoder Training and Domain Adaptation Hyperparameters for T12 and T15

Parameter	T12	T15
Optimizer	Adam	AdamW
Learning rate scheduler	None	Cosine
Learning rate warmup batches	–	1000
Training	73 epochs	120000 batches
Batch size	64	64
β_0, β_1	(0.9, 0.999)	(0.9, 0.999)
ϵ	0.1	0.1
LR (start / end)	0.02 / 0.02	0.005 / 0.0001
Weight decay (main)	1×10^{-5}	0.001
Grad norm clip	–	10
Neural dim	256	512
# GRU layers	5	5
GRU hidden units	1024	768
Bidirectional	False	False
GRU dropout	0.4	0.4
Input layer dropout	0	0.2
Patch size / stride	32 / 4	14 / 4
White noise std	0.8	1.0
Baseline shift (constant offset std)	0.2	0.2
Random cut (timesteps)	–	3
Smoothing kernel std	2.0	2.0
ALIGN		
Rep. layer index	–	2
Discriminator hidden dim	–	256
Discriminator LR multiplier	–	0.8
Discriminator weight decay	–	1×10^{-5}
Alpha type	–	Alternative ($\omega = 24$)
Alpha warmup steps	–	5
Domain dropout	–	0.0
TTA learning rate	–	5×10^{-4}
Stretch range	–	(1.5, 5.0)

Table 8: Transformer Decoder Training and Domain Adaptation Hyperparameters for T15

Parameter	T15 Transformer	T15 Transformer + ALIGN
Optimizer	AdamW	AdamW
Learning rate scheduler	Cosine	Cosine
Learning rate warmup batches	1000	1000
Training	120000 batches	120000 batches
Batch size	64	64
β_0, β_1	(0.9, 0.999)	(0.9, 0.999)
ϵ	0.1	0.1
LR (start / end)	0.005 / 0.0001	0.005 / 0.0001
Weight decay (main)	0.001	0.001
Grad norm clip	10	10
Neural dim	512	512
Embedding dim m	384	384
# Transformer layers	5	5
# attention heads	6	6
Attention head dim	512	512
MLP dim ratio	4	4
Transformer dropout	0.35	0.35
Attention dropout	0.4	0.4
Input layer dropout	0.2	0.2
Patch size / stride	14 / 4	14 / 4
Gaussian smoothing width / size	2.0 / 21	2.0 / 21
Patch-level masking	Disabled	Disabled
White noise std	0.2	0.2
Baseline shift (constant offset std)	0.05	0.05
Random cut (timesteps)	3	3
Source / target sessions	–	5 / 8
ALIGN		
Rep. layer index	–	3
Discriminator hidden dim	–	256
Discriminator LR multiplier	–	0.5
Discriminator weight decay	–	1×10^{-5}
Domain loss weight λ_{DANN}	–	0.8
Target-domain weight λ_{tgt}	–	0.8
Alpha type	–	Alternative ($\omega = 16$)
Alpha warmup steps	–	100
Domain dropout	–	0.0
Channel dropout prob.	–	0.0
Stretch range	–	(1.5, 5.0)
Include original samples	–	True
Include stretched samples	–	True
Include prolonged samples	–	False