Improving Neural Decoding Performance for Language BCIs by Explicitly Modeling Context-Induced Noise

Yao Jia[1,2,3*] **Xianhan Tan**[1,2,3*] **Binli Luo**[4 *] **Yueming Wang**[1] **Yu Qi**[1,2,3,5†]

[1]MOE Frontier Science Center for Brain Science and Brain-Machine Integration, Zhejiang University
[2]College of Computer Science and Technology, Zhejiang University
[3]Affiliated Mental Health Center & Hangzhou Seventh People's Hospital, Zhejiang University
[4]College of Computer Science and Technology, Central South University
[5]State Key Lab of Brain-Machine Intelligence, Zhejiang University

Abstract

Language brain-computer interfaces (BCIs), including speech and handwriting BCIs, hold significant promise for restoring communication in individuals with paralysis. However, the performance of current language BCIs remains limited due to the non-stationarity of neural activity. Existing neural decoding approaches typically treat neural non-stationarity as stochastic noise. For language BCIs, however, the noise is more complex due to the influence of semantic context in language sequences. For example, the articulation of the same phoneme can vary subtly depending on its surrounding content. Such context-induced variations introduce additional noise into neural representations and can confuse neural decoders, yet this issue has been largely overlooked in previous studies. Here, we propose explicitly modeling these context-induced noises to enhance neural decoding performance in language BCIs. Our key insight is that, unlike stochastic noise, context-induced noise carries useful information that can help the decoder disambiguate linguistically similar units. Therefore, by separating and modeling context-induced noise, we achieve improved decoding performance for language BCIs. Experiments on multiple datasets demonstrate improved decoding under controlled comparisons, and an enhanced system integrating our method performs well on the official Brain-to-Text Benchmark '24.

1 Introduction

Language brain-computer interfaces (BCIs), which decode spoken speech and handwritten text directly from neural activity, have recently made remarkable progress [Willett et al., 2021, Metzger et al., 2023, Willett et al., 2023, Tan et al., 2024, Wairagkar et al., 2025, Littlejohn et al., 2025, Qi et al., 2025]. However, high-performance neural decoding for language BCIs remains challenging.

One critical challenge in neural decoding is the nonstationary property of neural signals. Given the non-stationary nature of neural systems, neural signals could shift trial-by-trial, even when performing repetitive tasks [Renart and Machens, 2014]. This variation in neural signals makes it difficult for decoders to learn robust patterns, thereby degrading neural decoding performance.

While efforts have been made to construct robust BCIs, most existing neural decoding approaches treat neural variability as stochastic noise [Sussillo et al., 2016, Pandarinath et al., 2018], which can be oversimplified in the scenario of language BCIs. Unlike simple motor tasks such as reaching, language BCIs aim to decode a semantically rich sequence, composed of language-related units (such as letters in handwriting BCIs or phonemes in speech BCIs) [Silva et al., 2024], where the

*Equal contribution.
†Corresponding author (qiyu@zju.edu.cn).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

composition of neural variation could be far more complex. For instance, the pronunciation of the phoneme /b/ may differ subtly in words such as "bag" and "obtain" due to the surrounding phonemes and context. Similarly, in handwriting, the letter "t" could be written differently in "tap" and "mention", with the latter adopting a distinct style when combined with the suffix "-tion".

These slight differences in pronunciation or writing style could lead to shifts in neural representations, which we refer to as context-induced variations. Context-induced variations differ from stochastic neural variations, as they may contain useful patterns that can help decoders disambiguate linguistically similar units. While existing neural decoding approaches mostly overlooked contextual structure and treated language as independent units [Mugler et al., 2014, Ramsey et al., 2018, Liu et al., 2023a]. Some studies incorporated language models after unit-level decoding [Willett et al., 2021, Metzger et al., 2022, 2023, Willett et al., 2023]. However, these methods still fail to explicitly model the context-induced noise, limiting their overall decoding performance.

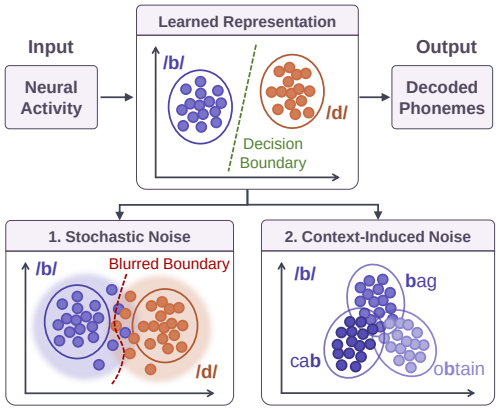

Figure 1: Illustration of neural variability in neural decoding. Top: The ideal latent space shows two well-separated clusters for /b/ and /d/. Bottom: The two key components contribute to neural variability. Left: the inherent Stochastic Noise increases the distribution variance of the two phonemes. Right: Context-Induced Noise leads to shifts in neural space.

In this work, we propose to explicitly model the context-induced noise in language BCIs by separately handling the two components in neural variability: (1) stochastic noise, arising from random neural firing or recording process [Faisal et al., 2008], which corrupts signal reliability and should be suppressed; and (2) context-induced noise, which can provide informative cues for disambiguation. This difference provides a basis for separating informative contextual structure from harmful fluctuations. Specifically, we propose a neural decoding framework that models the two components. Then they are subsequently processed, enabling the model to suppress stochastic noises while preserving information in context-induced variation. The proposed framework is evaluated on multiple intracortical recording datasets encompassing both speech and handwriting tasks. Our method consistently outperforms existing neural decoding and time-series analysis baselines, demonstrating the importance of explicitly modeling and disentangling neural variability for achieving high-performance decoding in language BCIs.

2 Related Work

Neural decoding algorithms have evolved from linear dynamical systems to deep sequence models. Early approaches, such as Kalman Filters [Wu et al., 2002, 2006], lacked the capacity to capture complex nonlinear dynamics. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997], significantly improved decoding by modeling temporal dependencies [Pandarinath et al., 2018, Willett et al., 2021, 2023, Yang et al., 2024, Qian et al., 2025, Jia et al., 2025, Sun et al., 2025, Wang et al., 2025]. Recently, Transformer-based architectures have achieved state-of-the-art performance by leveraging self-attention to capture long-term contextual relationships [Ye and Pandarinath, 2021, Le and Shlizerman, 2022, Azabou et al., 2023, Sun et al., 2024, Feghhi et al., 2025]. BIT explores cross-species neural pretraining and end-to-end speech decoding [Zhang et al., 2026]. Context-aware diphone targets explicitly represent phoneme transitions in the prediction space [Li et al., 2025]. Our approach focuses on learning feature-level reliability and context-ambiguity proxies to modulate temporal information aggregation, providing a complementary mechanism to adaptation, pretraining, and context-aware targets.

3 Methodology

3.1 Overall Framework

The goal of our framework is to decode intended information (e.g., phonemes or letters) from neural activity by explicitly modeling and leveraging different sources of neural variability. In particular, we focus on separating variability arising from stochastic neural noise and context-induced noise.

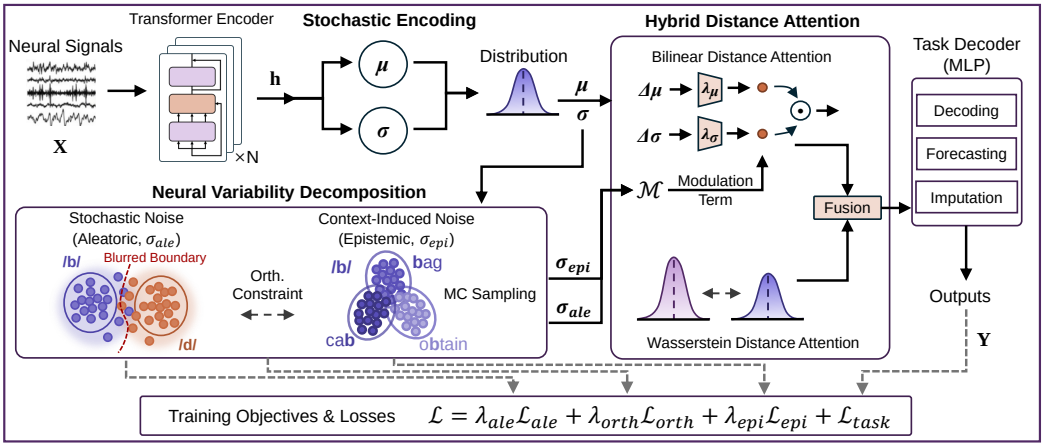

Figure 2: The uncertainty-aware decoding framework for language BCIs. The model consists of three modules: (1) Stochastic Encoder projects raw neural signals into a distribution latent space; (2) Neural Variability Decomposition disentangles the latent scale into an Aleatoric part (signal noise) and an Epistemic part (contextual ambiguity) under an orthogonality constraint; (3) Hybrid Distance Attention integrates Bilinear Distance Attention (refined by a Modulation Term $\mathcal{R}$) and Wasserstein Distance Attention to capture reliable temporal features for downstream tasks.

To achieve this decomposition, we adopt a probabilistic formulation that represents neural features in a stochastic latent space and characterizes their variability in a structured manner [Pandarinath et al., 2018, Triplett and Goodhill, 2019]. As illustrated in Figure 2, the proposed architecture departs from traditional deterministic mappings and processes neural activity through three core modules: (1) **Stochastic Encoding** of neural activity into Gaussian latent spaces to capture overall neural variability, (2) **Neural Variability Decomposition** into components associated with stochastic noise and context-induced variation, and (3) **Hybrid Distance Attention** for adaptively aggregating temporal features and leveraging the two noise components as attention modulators.

Please refer to the key notations in Table 5 for a clear understanding of the mathematical derivations.

3.2 Problem Formulation

We formulate neural decoding in language BCIs as a neural-to-linguistic units learning problem. Let $\mathcal{D} = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^{N}$ be a dataset of N paired neural recordings and corresponding linguistic label sequences. For a given trial, the input neural signal is represented as:

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}, \quad \mathbf{x}_t \in \mathbb{R}^C. \tag{1}$$

where T denotes the sequence length and C is the number of electrode channels.

The goal is to learn a mapping $f : \mathbf{X} \to \mathbf{Y}$ that accurately yields the sequence of linguistic units:

$$\mathbf{Y} = \{y_1, y_2, \ldots, y_T\}, \quad y_t \in \mathcal{V}, \tag{2}$$

where $\mathcal{V}$ denotes a phoneme or letter vocabulary.

3.3 Stochastic Encoding of Neural Activity

To handle the low signal-to-noise ratio (SNR) and the overall variability of neural signals, we introduce a Stochastic Encoding module inspired by variational representation learning [Pandarinath et al., 2018, Zhou and Wei, 2020, Aggarwal et al., 2025]. Unlike traditional BCI decoders that map inputs to deterministic hidden states, our module treats neural states as random variables. This allows the model to capture not only the content of the neural activity but also its associated reliability.

Module Architecture. Following the preliminary feature extraction by the Transformer encoder, we obtain the temporal features $\mathbf{h} \in \mathbb{R}^{T \times D}$:

$$\mathbf{h} = \text{Encoder}(\mathbf{X}), \tag{3}$$

To transition into the stochastic domain, we first process these features through linear projection:

$$\mathbf{h}' = \text{GELU}(\text{LN}(W_p\mathbf{h} + \mathbf{b}_p)), \tag{4}$$

where $\mathbf{h}' \in \mathbb{R}^{T\times D}$. Subsequently, two independent linear heads project $\mathbf{h}'$ into the parameters of Gaussian distributions:

$$\boldsymbol{\mu} = W_\mu \mathbf{h}' + \mathbf{b}_\mu, \tag{5}$$

$$\boldsymbol{\sigma} = \text{Softplus}(W_\sigma \mathbf{h}' + \mathbf{b}_\sigma) + \epsilon. \tag{6}$$

Here, $\boldsymbol{\mu} \in \mathbb{R}^{T\times \lfloor D/2 \rfloor}$ encodes the mean neural state, representing the most likely latent representation; $\boldsymbol{\sigma} \in \mathbb{R}^{T\times \lfloor D/2 \rfloor}$ captures local variation magnitude, reflecting variability in neural responses. At each time step t, the latent state is represented as $\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \text{diag}(\boldsymbol{\sigma}_t^2))$.

3.4 Learning Neural Variability from Latent Representations

The latent scale $\boldsymbol{\sigma}$ captures neural variability arising from multiple sources, including stochastic noise and context-induced ambiguity. To enable effective decoding, we propose a Neural Variability Decomposition mechanism that parameterizes two noise components as Aleatoric and Epistemic uncertainty [Hüllermeier and Waegeman, 2021], while emphasizing their functional roles in capturing stochastic and context-induced noise, respectively. Refer to Appendix F for formal definitions.

Learning Stochastic Noise via Predictive Entropy Supervision. The first component is designed to capture variability induced by stochastic neural noise, parameterized as Aleatoric uncertainty (σ_{ale}). We employ a dedicated projection head $\psi_{ale}(\cdot)$ to estimate it from the latent scale $\boldsymbol{\sigma}$:

$$\sigma_{ale} = \text{Softplus}(\psi_{ale}(\boldsymbol{\sigma})) + \epsilon. \tag{7}$$

We use normalized prediction entropy as a model-derived calibration target for unit-level neural unreliability:

$$\mathcal{L}_{ale} = \frac{1}{T}\sum_{t=1}^{T}\left(\sigma_{ale,t} - \text{sg}\left[\frac{\mathbb{H}(\hat{\mathbf{y}}_t)}{\log|\mathcal{V}|}\right]\right)^2, \tag{8}$$

where $\mathcal{V}$ is the output vocabulary and $\mathbb{H}(\hat{\mathbf{y}}_t) = -\sum_{v\in\mathcal{V}} \hat{y}_{t,v}\log\hat{y}_{t,v}$. The normalized entropy target lies in $[0, 1]$. The stop-gradient operator blocks gradients from the auxiliary MSE through the target probabilities.

Predictive entropy increases when neural noise corrupts evidence for any specific unit. By minimizing this Mean Squared Error (MSE), σ_{ale} is forced to track the sample-wise prediction confidence, effectively learning a "noise map" that identifies unreliable neural components. This supervision calibrates σ_{ale} to predictive unreliability, which can arise from stochastic corruption.

Learning Context-Induced Noise via Modeling Context Ambiguity. The second component is designed to capture variability arising from ambiguity between competing linguistic units under different contexts, parameterized as Epistemic uncertainty (σ_{epi}). Such ambiguity often occurs when multiple units share similar neural representations. We estimate this component via Monte Carlo sampling in the latent space [Gal and Ghahramani, 2016].

We perform K stochastic evaluations of the shared epistemic projection head on $\boldsymbol{\mu}$. At position t, these evaluations produce scalar outputs:

$$r_t^{(k)} = \psi_{epi}(\boldsymbol{\mu}_t; \xi_k), \qquad \bar{r}_t = \frac{1}{K}\sum_{k=1}^{K} r_t^{(k)}, \qquad \sigma_{epi,t} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(r_t^{(k)} - \bar{r}_t\right)^2}, \tag{9}$$

where ξ_k denotes the stochastic realization of the projection head. Thus, σ_{ale} is estimated from $\boldsymbol{\sigma}$, while σ_{epi} measures the sensitivity of a projection of $\boldsymbol{\mu}$ to stochastic perturbations.

To calibrate this measure, we supervise σ_{epi} using the margin between the top-two predicted classes:

$$\mathcal{L}_{epi} = \frac{1}{T}\sum_{t=1}^{T} ||\sigma_{epi,t} - \text{sg}\left[\text{Sigmoid}(-\Delta\ell_t)\right]||_2^2, \tag{10}$$

where $\Delta\ell_t = \text{logit}_{top1,t} - \text{logit}_{top2,t}$, representing the gap between the two top logits at step t.

A smaller logit gap $\Delta\ell_t$ indicates greater contextual ambiguity between competing units, leading to a larger supervision value. Unlike entropy, which conflates noise, logit margins indicate decision-level confusion. Together with latent perturbation sensitivity, this target encourages σ_{epi} to track task-relevant ambiguity, whose association with linguistic context is assessed by the matched-context analysis.

Orthogonality Constraint. To prevent the two noise components from capturing redundant information, we enforce an orthogonality constraint based on the Pearson correlation coefficient. Let $\bar{\sigma}_{ale}$ and $\bar{\sigma}_{epi}$ be the sequence-wise mean values, the orthogonality loss is formulated as:

$$\mathcal{L}_{orth} = \left| \frac{\sum_{t=1}^{T}(\sigma_{ale,t} - \bar{\sigma}_{ale})(\sigma_{epi,t} - \bar{\sigma}_{epi})}{\sqrt{\sum_{t=1}^{T}(\sigma_{ale,t} - \bar{\sigma}_{ale})^2 \sum_{t=1}^{T}(\sigma_{epi,t} - \bar{\sigma}_{epi})^2}} \right| \tag{11}$$

Minimizing absolute Pearson correlation discourages linear redundancy between the two proxies and encourages complementary information for decoding, ensuring a more interpretable decomposition. Without this constraint, the model may explain all variability using a single component.

3.5 Hybrid Distance Attention for Variability-Aware Temporal Modeling

Traditional self-attention often overlook both the variability and geometric structure of neural latent representations. To address this limitation, we propose the Hybrid Distance Attention (HDA), which performs temporal aggregation directly over variability representations. HDA combines feature-level interactions with distribution-level alignment [Fan et al., 2022], while using learned variability as a routing signal to regulate information flow across time through modulating attention scores.

The first component of HDA is a learnable Bilinear distance that captures interactions between mean and variability features. Given the parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma})$, we project them into H heads. For each head h, the Query $\mathbf{Q}^h$ and Key $\mathbf{K}^h$ are constructed from the mean and variance components. The distance between time step i and j is defined as:

$$D_{ij}^{bil} = ||\mathbf{Q}_{\mu,i}^h - \mathbf{K}_{\mu,j}^h||_{W_\mu^h}^2 + \alpha_{weight}||\mathbf{Q}_{\sigma,i}^h - \mathbf{K}_{\sigma,j}^h||_{W_\sigma^h}^2, \tag{12}$$

where $\|\cdot\|_W^2$ denotes the weighted Mahalanobis-like distance [De Maesschalck et al., 2000] controlled by learnable weights $W_\mu^h, W_\sigma^h \in \mathbb{R}^{D_H}$. This per-dimension weighting allows the model to prioritize highly selective neural channels while de-emphasizing noisy ones.

To incorporate signal quality and contextual ambiguity into temporal modeling, we introduce a routing bias $\mathcal{R}_{ij}$ that dynamically re-weights attention using learned variability:

$$\mathcal{R}_{ij} = -\alpha_{ale}\,||\sigma_{ale,i} - \sigma_{ale,j}|\,| + \alpha_{epi}\,||\sigma_{epi,i} - \sigma_{epi,j}|\,|, \tag{13}$$

where α_{ale} and α_{epi} are learnable non-negative coefficients restricted by boundaries.

The routing bias $\mathcal{R}_{ij}$ serves as an adaptive filter controlling information flow. Specifically, the aleatoric term penalizes attention between time steps with mismatched noise levels, since cross-attention between them may contaminate cleaner signals. In contrast, the epistemic term rewards attention between time steps with contrasting ambiguity levels, as a less ambiguous state can provide contextual evidence to disambiguate a more uncertain one. When variability differences are small, this routing bias would reduces to standard distance-based attention.

While the Bilinear distance captures feature-wise interactions, it may overlook the global geometric overlap between latent distributions. Thus, we incorporate the 2nd-order Wasserstein distance [Rüschendorf, 1985], which provides a principled metric for Gaussian distributions:

$$\mathcal{W}_2(\mathcal{N}_i, \mathcal{N}_j)^2 = ||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2^2 + ||\boldsymbol{\sigma}_i - \boldsymbol{\sigma}_j||_2^2. \tag{14}$$

This term provides a complementary measure of distributional alignment, capturing the overall geometric relationship between latent states.

Hybrid Distance Attention Fusion. To combine these complementary views, we adopt a learnable hybrid fusion strategy. The final attention score A_{ij} is computed as:

$$A_{ij} = \alpha_{hybrid} \cdot \left(\exp(-D_{ij}^{bil}) + \mathcal{R}_{ij}\right) + (1 - \alpha_{hybrid}) \cdot \exp(-\mathcal{W}_2^2), \tag{15}$$

where $\alpha_{hybrid} \in [0, 1]$ is a learnable coefficient.

This hybrid design separates relation estimation from information routing: the distance terms estimate feature-level and distribution-level relations, whereas $\mathcal{R}_{ij}$ biases the flow of information according to

reliability and contextual ambiguity. For each attention head h, the hybrid scores are normalized over the key index j and used to aggregate projected values:

$$\widetilde{A}_{ij}^{h} = \frac{\exp(A_{ij}^{h})}{\sum_{j'=1}^{T} \exp(A_{ij'}^{h})}, \qquad \mathbf{Z} = \mathrm{Concat}_{h=1}^{H} \left[\widetilde{\mathbf{A}}^{h} \mathbf{V}^{h}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \right] W_O. \tag{16}$$

Omitting the batch dimension, $\boldsymbol{\mu}, \boldsymbol{\sigma} \in \mathbb{R}^{T \times d}$ with $d = \lfloor D/2 \rfloor$, $\sigma_{ale}, \sigma_{epi} \in \mathbb{R}^{T}$, $\widetilde{\mathbf{A}}^{h} \in \mathbb{R}^{T \times T}$, and $\mathbf{V}^{h} \in \mathbb{R}^{T \times d_v}$. The output projection $W_O \in \mathbb{R}^{H d_v \times D}$ produces the contextual representation $\mathbf{Z} \in \mathbb{R}^{T \times D}$.

Appendix E establishes the metric property of the square-rooted weighted distance under positive weights and a shared coordinate representation. The Gaussian Wasserstein distance $\mathcal{W}_2$ also defines a metric on the latent distributions. Their squared forms serve as inputs to attention kernels. Temporal robustness is assessed empirically through the session-held-out and chronological evaluations.

3.6 Task-Specific Decoder Module

The contextual representation $\mathbf{Z}$ is passed to the task-specific prediction head. For unit decoding, the head produces class probabilities, while the imputation head reconstructs neural features:

$$\hat{\mathbf{y}}_t = \mathrm{softmax}(\phi_{dec}(\mathbf{Z}_t)), \qquad \hat{\mathbf{x}}_t = \phi_{imp}(\mathbf{Z}_t). \tag{17}$$

Linguistic-unit decoding and future-step forecasting use cross-entropy supervision, while neural-activity imputation uses MSE. The task-specific output definitions are provided in Appendix A.

3.7 Learning Objectives

Regardless of the output configuration, the model is trained by optimizing a joint loss that ensures both task performance and uncertainty calibration. The total loss $\mathcal{L}$ for a given task is formulated as:

$$\mathcal{L} = \mathcal{L}_{task} + \lambda_{ale}\mathcal{L}_{ale} + \lambda_{epi}\mathcal{L}_{epi} + \lambda_{orth}\mathcal{L}_{orth}, \tag{18}$$

where $\mathcal{L}_{task}$ is the task-specific objective.

In summary, the proposed framework reformulates neural decoding from a traditional deterministic estimation mapping into a principled probabilistic inference over a latent manifold. We use a Neural Statistic Initializer to initialize the variability heads and bilinear-distance weights from channel selectivity and global variability statistics computed only on the downstream training subset. Its construction is detailed in Appendix A.3. A summary of the training procedure is provided in Algorithm 1, detailing the transition from raw neural inputs to task predictions.

4 Experiments

4.1 Experimental Setup

Datasets. We validate our framework on three publicly available invasive language BCI datasets, covering both attempted speech and imagined handwriting tasks: **Speech I** [Willett et al., 2023] (covering 9,647 sentences and 39 phoneme classes), **Speech II** [Card et al., 2024] (covering 8,015 sentences and 39 phonemes), **Handwriting** [Willett et al., 2021] (covering 730 sentences and 26 letters). The unit-aligned evaluation uses offline CTC-derived phoneme segments for speech and cue-based letter segments for handwriting. For speech, the alignment models are fitted to the labeled corpus used for unit-aligned preprocessing. Accordingly, these experiments evaluate downstream decoding conditional on corpus-level offline alignment. The alignment procedure and boundary perturbation analysis are detailed in Appendix B.

Baseline Methods. We compare our framework with multiple methods for neural decoding and sequence modeling. We include recurrent and convolutional methods like LSTM [Hochreiter and Schmidhuber, 1997] and TCN [Bai, 2018], which are widely used in neural decoding. We select the vanilla Transformer [Vaswani et al., 2017] and Conformer [Gulati et al., 2020] encoder as attention-based baselines. We further include recent general time-series analysis models, including iTransformer [Liu et al., 2023b] for inverted temporal-channel modeling, Mamba [Gu and Dao, 2024] for efficient long-context state-space modeling, TimesNet [Wu et al., 2022] for multi-periodicity analysis, and TimeMixer [Wang et al., 2024] for multi-scale mixing. We also include uncertainty-estimation baselines [Valdenegro-Toro and Saromo, 2022] to evaluate the advantage of our uncertainty proxies for modeling neural variability. The unit-aligned baseline results are obtained from architectures

Table 1: Direct unit-aligned speech and handwriting decoding accuracy (%, ↑) on intracortical recording datasets under different sequence lengths T. Mean and standard deviation (Std in footnotesize) are reported over five runs. Best results are in **bold** and the second underlined.

Model	Speech I				Speech II				Handwriting			
	T=20	T=30	T=40	T=50	T=20	T=30	T=40	T=50	T=20	T=30	T=40	T=50
LSTM	52.4±0.4	54.3±0.3	55.5±0.2	55.3±0.7	73.0±0.2	73.7±0.4	74.2±0.5	74.2±0.2	57.7±1.2	60.4±1.7	61.6±1.4	61.8±1.1
TCN	68.4±0.1	70.8±0.3	71.9±0.2	72.2±0.2	81.5±0.4	83.0±0.4	83.4±0.4	83.4±0.7	65.5±1.1	68.0±0.8	69.3±1.9*	68.9±1.2*
Transformer	74.5±0.3	77.4±0.4	78.1±0.4	78.2±0.5	85.9±0.6	86.7±0.4	87.0±0.8*	87.5±0.4	59.2±1.8	61.2±1.8	62.6±1.6	61.4±2.0
Conformer	69.7±0.4	75.0±0.4	75.9±0.5	76.0±0.3	81.5±0.5	83.8±0.3	85.3±0.5	85.2±0.4	61.1±1.6	62.8±1.5	65.1±1.3	65.4±1.1
iTransformer	41.6±0.5	40.0±0.7	39.6±0.5	38.8±1.0	55.7±2.0	51.0±1.9	49.8±2.4	50.7±2.1	53.7±2.1	55.1±1.8	55.6±2.3	54.6±1.8
Mamba	42.9±0.3	44.0±0.4	44.8±0.6	45.0±0.5	63.1±1.5	65.9±1.0	65.8±3.7	67.6±0.4	44.6±12.7	49.8±1.8	48.0±2.8	45.7±3.5
TimesNet	70.6±0.3	73.6±0.3	75.3±0.3	75.3±0.3	81.9±0.3	83.3±0.2	84.2±0.2	84.3±0.4	53.2±1.3	56.0±2.0	57.7±2.1	56.9±0.7
TimeMixer	54.3±0.7	56.4±0.4	58.1±0.7	58.6±0.3	65.0±1.2	65.9±0.8	66.9±0.5	67.3±0.8	63.3±1.2	65.1±2.1	66.3±2.0	66.5±1.6
Ours	**76.1**±0.4	**79.7**±0.4	**80.2**±0.7	**80.1**±0.5	**87.0**±0.5	**87.0**±0.4	**87.3**±0.5	**88.6**±0.4	**66.7**±1.2	**69.1**±0.7	**69.7**±1.1	**69.1**±0.9

reimplemented or adapted under the same preprocessing, downstream splits, and task definitions. Each architecture is tuned within its search space using validation performance for model selection (Appendix D). The same-backbone comparison in Table 2 uses the TMT encoder in OursTMT, followed by the proposed stochastic projections, variability estimation, HDA, and task head.

Training Details. We evaluate our framework across three tasks to cover the clinical requirements: (1) **Linguistic Unit Decoding** to assess communication restoration; (2) **Future Step Forecasting** to evaluate low-latency intent detection; and (3) **Neural Activity Imputation** to test robustness against signal corruption or electrode failure. For the random-split unit-aligned experiments, datasets are divided into training, validation, and test sets with an 8:1:1 ratio, and all models are tuned using Ray Tune with the ASHA scheduler. We optimize all models with Adam, employing early stopping (20-epoch patience) based on validation accuracy for decoding/forecasting and validation loss for imputation. Experiments are conducted on a cluster of eight NVIDIA GeForce RTX 4090 GPUs.

4.2 Main Comparative Study

Linguistic Unit Decoding. Table 1 reports controlled unit-aligned decoding accuracy across three intracortical datasets. We report the decoding accuracy (mean ± std) over five runs. The results demonstrate that our method significantly outperforms both common neural decoders and modern time-series models in most cases ($p < 0.05$, two-sided paired t-tests). The asterisk marks indicate that there is no significant difference compared with our method.

Notably, our method significantly surpasses the second-best Transformer (e.g., +2.0% on **Speech I** and +5.7% on **Handwriting**), indicating that our model effectively captures temporal dependencies across a range of input lengths. Additionally, although TCN performs comparably to ours on **Handwriting**, our method outperforms by a large margin on speech tasks where long-term relationships are more complex (e.g., +8.2% on **Speech I** and +4.7% on **Speech II**). This stability validates our model's efficacy in suppressing neural noise and capturing long-term context dynamics.

In addition, we conduct a strict cross-session evaluation to demonstrate the advantages of our method in long-term stability (Table 7, Appendix C.1), where the performance gap between our method and the best baselines widens further. To examine whether our unit-level representation improvements can benefit downstream sentence decoding, we integrate all methods with the same language-model post-processing pipeline [Willett et al., 2023] (Tables 11 & 12). The results show that our method remains competitive with strong decoding backbones and that our enhanced variant further improves performance, demonstrating the scalability of our variability-aware modeling. Detailed fair comparison protocols and results are provided in Appendix C.2.

Sequence-Level and Benchmark Evaluation. Under the controlled unit-aligned pipeline, adding our module to the Time-Masked Transformer (TMT, [Feghhi et al., 2025]) backbone improves ACC from 79.5% to 81.7% and reduces WER from 19.5% to 17.8% with the same LLM post-processing (Appendix C.2). On the released benchmark validation partitions, paired TMT comparisons further improve sequence

Table 2: Official Brain-to-Text Benchmark '24 hidden-test WER (%, ↓).

Model	5-gram LM + OPT-6B	Fine-tuned Llama 3.1 8B
NPTL baseline	9.76	–
TMT	8.18	5.68
OursTMT	**7.25**	**4.85**

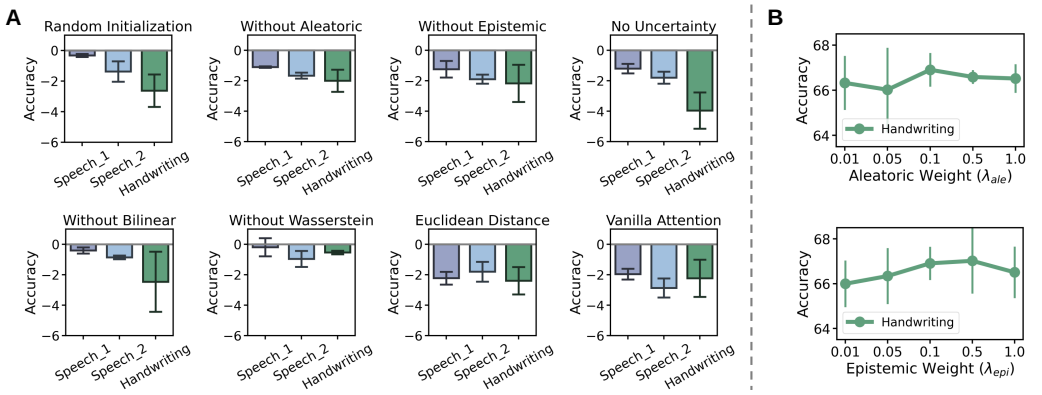

Figure 3: Ablation study and hyperparameter sensitivity analysis on linguistic-unit decoding.

PER from 15.4% to 14.8% on Speech I and
from 9.6% to 9.0% on Speech II, with corresponding WER improvements reported in Table 13. For standardized evaluation, we submit our method to the official Brain-to-Text Benchmark '24[3]. As shown in Table 2, our TMT-based system achieves 7.25% WER with a 5-gram LM and OPT-6B rescoring. Our enhanced system integrates ten independently trained decoders using a fine-tuned Llama 3.1 8B and achieves 4.85% WER. The controlled same-backbone comparisons assess the module contribution, and the official hidden-test results assess the complete decoding systems.

Future Step Forecasting. We perform forecasting on the two speech datasets (Table 3), as the handwriting dataset lacks sufficient temporal continuity. We compare against the top-4 performing baselines. Our framework achieves the highest forecasting accuracy across all settings. More results can be found in the Appendix C.3.

Table 3: Phoneme forecasting accuracy (%, ↑) on two speech datasets under different future steps K.

Model	Speech I		Speech II	
	K=2	K=4	K=2	K=4
LSTM	82.1±0.4	50.0±0.8	79.6±0.5	50.3±0.8
TCN	83.5±0.4	48.2±1.0	78.9±0.6	48.1±0.5
Transformer	82.2±0.6	49.8±0.5	78.1±0.8	50.3±0.4
TimesNet	83.5±0.3	51.4±1.0	79.2±1.1	51.6±0.9
Ours	**85.3**±1.3	**53.1**±1.4	**87.0**±0.7	**56.3**±1.4

On Speech II, our model improves forecasting accuracy over the strongest baseline by 7.4 percentage points at $K = 2$. This result supports short-horizon phoneme prediction, and the separate chronological and session-held-out evaluations in Appendix C.1 assess robustness across recording sessions. Furthermore, our framework is more stable as the forecasting horizon extends. For instance, on **Speech I**, the TCN experiences a great performance drop at $K = 4$ (-35.5%), whereas our method exhibits a more graceful decay (-32.2%). This may be attributed to our variability-aware attention mechanism, which selectively filters out historical accumulated errors.

Neural Activity Imputation. To evaluate the framework's robustness against electrode failure or signal corruption, we conduct neural activity imputation (Appendix C.4). As shown in Table 15 & 16, our model achieves the lowest Mean Squared Error (MSE) and Mean Absolute Error (MAE) across all datasets. A key observation is that our performance lead is significantly greater on the two speech datasets than on the handwriting task. Specifically, our model outperforms the best baseline by approximately 0.002 (MSE) on **Speech**, whereas the margin on **Handwriting** is notably smaller.

4.3 Ablation Study

To verify the contribution of each component, we conduct ablation on the linguistic-unit decoding task across eight configurations (Figure 3A), which consist of Variability Decomposition and Hybrid Distance Attention modules. The gray horizontal line represents the full model's average accuracy.

Effectiveness of Variability Decomposition. Removing the Neural Statistic Initializer results in noticeable performance degradation, supporting the use of training-data statistics for initialization. The "No Uncertainty" configuration in Figure 3 denotes the internal ablation "Ours w/o uncertainty heads and routing", which removes the uncertainty heads, their auxiliary losses, and uncertainty-based

[3] https://eval.ai/web/challenges/challenge-page/2099/overview

Table 4: Comparison with uncertainty baselines on direct decoding (%, ↑). ✓/× indicate whether using uncertainty estimates and uncertainty-based attention modulation, respectively.

Model	Uncertainty		Speech I		Speech II		Handwriting	
	Estimate	Modulation	T=20	T=50	T=20	T=50	T=20	T=50
MCDropout	✓	×	58.8±0.3	60.5±0.1	72.2±0.6	72.7±0.6	57.5±1.8	62.7±1.1
Ensembles	✓	×	62.2±0.1	63.7±0.1	75.8±0.7	76.2±0.7	62.9±1.6	68.0±1.5
OursTVD	✓	×	69.8±0.3	73.2±0.2	80.0±0.6	81.1±0.3	59.0±1.7	60.2±1.4
Ours	✓	✓	**76.1**±0.4	**80.1**±0.5	**87.0**±0.5	**88.6**±0.4	**66.7**±1.2	**69.1**±0.9

routing. We additionally compare against a retuned deterministic HDA model and a single-uncertainty variant, and perform capacity-matched target-shuffling and target-swapping controls (Table 23). The full model outperforms these controls, supporting the utility of the assigned calibration targets and complementary variability estimates. The same-TMT-backbone augmentation and routing comparisons in Table 24 further assess the contribution of uncertainty-based routing.

Superiority of Hybrid Distance Attention. Removing the Bilinear or Wasserstein distance has different negative effects across datasets, demonstrating that both are essential, with the Bilinear distance contributing more, especially for **Handwriting**. Furthermore, replacing our hybrid mechanism with either the Euclidean Distance or Vanilla Attention results in a substantial loss in decoding precision. This decline underscores that neither simple geometric distance nor standard context-matching can sufficiently capture the complex probabilistic manifold of neural activity.

4.4 Hyperparameter Sensitivity

We further investigate the sensitivity of our model to the loss weights of the two components: λ_{ale} and λ_{epi}. In Figure 3B, we evaluate the impact on **Handwriting** by varying coefficients from 0.01 to 1.0, and provide extended results on **Speech** in Appendix C.5. Our findings reveal a distinct contrast in how the model responds to these two types of variability. The accuracy shows a similar changing trend across the two weights on **Handwriting**. However, the **Speech II** reveals a catastrophic sensitivity to aleatoric weight, where increasing λ_{ale} beyond 0.1 causes accuracy to drop from 86% to 23%. Conversely, epistemic weight λ_{epi} is more stable and reaches an optimum around 0.1-0.5. This indicates that excessive denoising may lead the model to suppress components useful for decoding, whereas enhancing attention to context-induced noise consistently improves performance.

4.5 Validation and Utility of Learned Variability

Variability Modeling. To isolate the effect of the uncertainty estimate for neural variability modeling, we compare against generic uncertainty baselines (MC-Dropout and Ensembles [Valdenegro-Toro and Saromo, 2022]) and a simpler Total-Variance-Decomposition variant (OursTVD). These three methods use simpler uncertainty estimation and lack mechanisms for attention modulation. Table 4 shows that these alternatives improve over some deterministic baselines but remain below our full model, indicating that uncertainty estimates for variability modeling and attention modulation mechanisms are both important for language BCI decoding. More results are in Appendix C.6.

Selectivity of Learned Variability. To validate whether the learned variability responds selectively to signal reliability and context ambiguity, we examine their sensitivity to Gaussian noise and to contextual information such as phoneme boundaries (Appendix C.7, Figure 6). The results show stronger aleatoric responses to injected noise and stronger epistemic responses to phoneme boundary structure. This pattern supports using the two uncertainty estimates as proxies for stochastic noise and context-induced variation. We further quantify the correlations of the two uncertainty proxies with externally defined stochastic noise severity and context complexity, respectively (Table 18, 19 & 20). In addition, we explore the different responses of the variability ablation variant to high-noise and high-ambiguity samples (Table 21). A matched same-phoneme analysis further controls for session, phoneme frequency, neural SNR, σ_{ale}, and baseline prediction entropy/confidence. Rare neighboring contexts are associated with increases of 17.8% and 21.3% in σ_{epi} on Speech I and Speech II, respectively, with much smaller changes in σ_{ale} (Table 22). This supports the functional interpretation of σ_{epi} as a context-ambiguity proxy after controlling for baseline prediction difficulty.

Uncertainty Calibration Analysis. We analyze the calibration behavior of the two uncertainties using model-internal representations extracted from the final variability-decomposition layer during

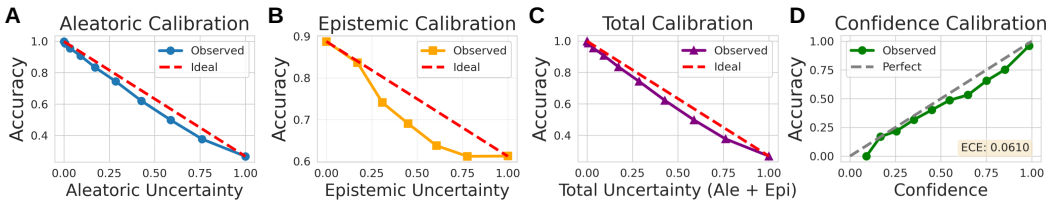

Figure 4: Quantitative calibration of uncertainty estimates for linguistic-unit decoding on **Speech I**.

decoding. In Figure 4A and 7A, aleatoric uncertainty exhibits a strong negative correlation with accuracy. This aligns with the intuition that increasing stochastic noise intensity leads to a linear decline in decoding accuracy. In contrast, the epistemic uncertainty may follow a negative exponential pattern (Figure 4B and 7B). A sharp drop in accuracy occurs at the early stage of context confusion, which is likely the key to addressing the model's performance bottlenecks. We further evaluate whether the uncertainties are consistent with the model's predictive confidence (Figure 4D and 7D). The predicted confidence closely follows empirical accuracy with a low expected calibration error (ECE). Results suggest that uncertainties not only improve performance but also lead to more reliable confidence estimates. More calibration analyses are in Appendix C.8.

Practical Application of Uncertainty. Beyond performance, the learned uncertainty proxies offer some practical benefits, such as decode rejection. To simulate clinical scenarios where high-uncertainty attempts should be rejected, we force the decoder to actively reject the current decoding when the total uncertainty exceeds a preset threshold. As the rejection rate increases, the effective decoding accuracy rises steadily (Appendix C.9).

5 Conclusions and Limitations

In this work, we presented a variability-aware neural decoding framework for language BCIs. Unlike simpler motor, language decoding involves phonemes and letters whose neural representations can be systematically shaped by context, introducing additional noise that can confuse neural decoders. Importantly, this context-induced variation contains useful cues for distinguishing similar linguistic units. Thus, the two noise components should be treated differently. By combining variability modeling with Hybrid Distance Attention, our model suppresses stochastic noise while preserving context-induced structure for linguistic disambiguation. Experiments across multiple intracortical datasets and complementary decoding tasks show consistent improvements over baselines, especially in settings with higher neural variability such as long-term recordings. These results highlight the value of modeling neural variability as structured information for robust language BCI decoding.

This work has several limitations that need further investigation. First, future work could explore more complex, non-parametric latent spaces or flow-based generative models to better capture the non-linear manifold of high-dimensional neural signals. In addition, the Hybrid Distance Attention increases computational complexity, aggravating the application cost to some extent. Finally, the learned uncertainty components are task-calibrated functional proxies, and their physiological interpretation requires external validation. Each dataset contains one participant, so the generalization evidence concerns within-subject robustness across sessions; cross-subject transfer remains a direction for future work.

Acknowledgments and Disclosure of Funding

This work was supported by grants from the National Natural Science Foundation of China (No. 32595493, 62276228), the Zhejiang Provincial Natural Science Foundation of China (LR24F020002), and the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study (SN-ZJU-SIAS-002).

References

Francis R Willett, Donald T Avansino, Leigh R Hochberg, Jaimie M Henderson, and Krishna V Shenoy. High-performance brain-to-text communication via handwriting. *Nature*, 593(7858):

249–254, 2021.

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.

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.

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*, 2024.

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*, pages 1–8, 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, et al. A streaming brain-to-voice neuroprosthesis to restore naturalistic communication. *Nature neuroscience*, pages 1–11, 2025.

Yu Qi, Xinyun Zhu, Xinzhu Xiong, Xiaomeng Yang, Nai Ding, Hemmings Wu, Kedi Xu, Junming Zhu, Jianmin Zhang, and Yueming Wang. Human motor cortex encodes complex handwriting through a sequence of stable neural states. *Nature Human Behaviour*, pages 1–12, 2025.

Alfonso Renart and Christian K Machens. Variability in neural activity and behavior. *Current opinion in neurobiology*, 25:211–220, 2014.

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(1):13749, 2016.

Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

Alexander B Silva, Kaylo T Littlejohn, Jessie R Liu, David A Moses, and Edward F Chang. The speech neuroprosthesis. *Nature Reviews Neuroscience*, 25(7):473–492, 2024.

Emily M Mugler, James L Patton, Robert D Flint, Zachary A Wright, Stephan U Schuele, Joshua Rosenow, Jerry J Shih, Dean J Krusienski, and Marc W Slutzky. Direct classification of all american english phonemes using signals from functional speech motor cortex. *Journal of neural engineering*, 11(3):035015, 2014.

Nick F Ramsey, Efraïm Salari, Erik J Aarnoutse, Mariska J Vansteensel, Martin G Bleichner, and ZV Freudenburg. Decoding spoken phonemes from sensorimotor cortex with high-density ecog grids. *Neuroimage*, 180:301–311, 2018.

Yan Liu, Zehao Zhao, Minpeng Xu, Haiqing Yu, Yanming Zhu, Jie Zhang, Linghao Bu, Xiaoluo Zhang, Junfeng Lu, Yuanning Li, et al. Decoding and synthesizing tonal language speech from brain activity. *Science Advances*, 9(23):eadh0478, 2023a.

Sean L Metzger, Jessie R Liu, David A Moses, Maximilian E Dougherty, Margaret P Seaton, Kaylo T Littlejohn, Josh Chartier, Gopala K Anumanchipalli, Adelyn Tu-Chan, Karunesh Ganguly, et al. Generalizable spelling using a speech neuroprosthesis in an individual with severe limb and vocal paralysis. *Nature communications*, 13(1):6510, 2022.

A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. Noise in the nervous system. *Nature reviews neuroscience*, 9(4):292–303, 2008.

Wei Wu, M Black, Yun Gao, M Serruya, A Shaikhouni, J Donoghue, and Elie Bienenstock. Neural decoding of cursor motion using a kalman filter. *Advances in neural information processing systems*, 15, 2002.

Wei Wu, Yun Gao, Elie Bienenstock, John P Donoghue, and Michael J Black. Bayesian population decoding of motor cortical activity using a kalman filter. *Neural computation*, 18(1):80–118, 2006.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Xiaomeng Yang, Xinzhu Xiong, Xufei Li, Qi Lian, Junming Zhu, Jianmin Zhang, Yu Qi, and Yueming Wang. Reconstructing multi-stroke characters from brain signals toward generalizable handwriting brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2024.

Youkun Qian, Changjiang Liu, Peixi Yu, Xingchen Ran, Shurui Li, Qinrong Yang, Yang Liu, Lei Xia, Yijie Wang, Jianxuan Qi, et al. Real-time decoding of full-spectrum chinese using brain-computer interface. *Science Advances*, 11(45):eadz9968, 2025.

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.

Huaqin Sun, Zhengyi Wang, Yu Qi, and Yueming Wang. Decoding multi-joint hand movements from brain signals by learning a synergy-based neural manifold. *Patterns*, 6(11), 2025.

Lei Wang, Xiaomeng Yang, Xinzhu Xiong, Yao Jia, Yueming Wang, and Yu Qi. State-dependent neural decoding of complex handwriting via mixture of experts with recurrent routing. *IEEE Transactions on Cognitive and Developmental Systems*, 2025.

Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 2021.

Trung Le and Eli Shlizerman. Stndt: Modeling neural population activity with spatiotemporal transformers. *Advances in Neural Information Processing Systems*, 35:17926–17939, 2022.

Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael Mendelson, Blake Richards, Matthew Perich, Guillaume Lajoie, and Eva Dyer. A unified, scalable framework for neural population decoding. *Advances in Neural Information Processing Systems*, 36:44937–44956, 2023.

Huaqin Sun, Yu Qi, Xiaodi Wu, Junming Zhu, Jianmin Zhang, and Yueming Wang. Decoding joint-level hand movements with intracortical neural signals in a human brain–computer interface. *IEEE Transactions on Cognitive and Developmental Systems*, 16(6):2100–2109, 2024.

Ebrahim Feghhi, Shreyas Kaasyap, Nima Hadidi, and Jonathan C Kao. Time-masked transformers with lightweight test-time adaptation for neural speech decoding. *arXiv preprint arXiv:2507.02800*, 2025.

Yizi Zhang, Linyang He, Chaofei Fan, Tingkai Liu, Han Yu, Trung Le, Jingyuan Li, Scott Linderman, Lea Duncker, Francis Willett, Nima Mesgarani, and Liam Paninski. A cross-species neural foundation model for end-to-end speech decoding. In *International Conference on Learning Representations*, 2026. URL `https://proceedings.iclr.cc/paper_files/paper/2026/hash/34307cd20546fa1c6de90f303973ccf7-Abstract-Conference.html`.

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.

Marcus A Triplett and Geoffrey J Goodhill. Probabilistic encoding models for multivariate neural data. *Frontiers in neural circuits*, 13:1, 2019.

Ding Zhou and Xue-Xin Wei. Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-vae. *Advances in neural information processing systems*, 33: 7234–7247, 2020.

Naman Aggarwal, Siddhartha R Dalal, and Vishal Misra. The bayesian geometry of transformer attention. *arXiv preprint arXiv:2512.22471*, 2025.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.

Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. Sequential recommendation via stochastic self-attention. In *Proceedings of the ACM web conference 2022*, pages 2036–2047, 2022.

Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

Ludger Rüschendorf. The wasserstein distance and approximation theorems. *Probability Theory and Related Fields*, 70(1):117–129, 1985.

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.

Shaojie Bai. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023b.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024.

Matias Valdenegro-Toro and Daniel Saromo. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1508–1516. IEEE, 2022. doi: 10.1109/CVPRW56347.2022.00157.

Joshua I Glaser, Ari S Benjamin, Raeed H Chowdhury, Matthew G Perich, Lee E Miller, and Konrad P Kording. Machine learning for neural decoding. *eneuro*, 7(4), 2020.

Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216, 2007.

Walter Rudin. *Function theory in the unit ball of Cn*. Springer Science & Business Media, 2008.

David Barber and Felix Agakov. The im algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16(320):201, 2004.

A Method Details

A.1 Detailed Architectures

Table 5: Summary of key notations in our framework.

Notation	Shape	Description
$\mathbf{X}$	$\mathbb{R}^{T\times C}$	Input neural spike sequence. T and C are the sequence length and the number of recording channels.
$\mathbf{h}$	$\mathbb{R}^{T\times D}$	Encoded temporal features.
$\boldsymbol{\mu}, \boldsymbol{\sigma}$	$\mathbb{R}^{T\times \lfloor \frac{D}{2} \rfloor}$	Stochastic mean and uncertain features of the Gaussian latent state.
K	Scalar	Number of Monte Carlo samples for epistemic estimation.
$\sigma_{ale}, \sigma_{epi}$	$\mathbb{R}^{T}$	Functional proxies estimated from $\boldsymbol{\sigma}$ and stochastic projections of $\boldsymbol{\mu}$, respectively.
$\alpha_{ale}, \alpha_{epi}$	Scalar	Learnable modulation weights for uncertainties.
$W_{\boldsymbol{\mu}}, W_{\boldsymbol{\sigma}}$	$\mathbb{R}^{H\times D_H}$	Learnable per-dimension weights of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ in Bilinear distance. H and H_D represent the number of attention heads and corresponding dimensions.
α_{weight}	Scalar	Learnable balancing coefficient between $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ for Bilinear distance.
α_{hybrid}	Scalar	Learnable coefficient for balancing the attention scores of Bilinear and Wasserstein distance.
$\lambda_{ale}, \lambda_{epi}$	Scalar	Fixed loss weights of modeling aleatoric and epistemic uncertainty.
λ_{orth}	Scalar	Fixed weight for the uncertainty orthogonality constraint.
$\mathbf{Y}$	$\mathbb{R}^{T}$	Target linguistic units and T is the sequence length.

A.2 Algorithm Pseudocode

Algorithm 1: Uncertainty-Aware Neural Decoding Framework

Input: Neural sequence $\mathbf{X} \in \mathbb{R}^{T\times C}$; Target labels $\mathbf{Y}$ (or mask $\mathcal{M}$); Task type $\mathcal{T}$.
Output: Optimized model parameters Θ.

Step 1. Stochastic Feature Extraction:
1. Compute deterministic hidden states: $\mathbf{h} = \text{Encoder}(\mathbf{X})$;
2. Map to Gaussian latent parameters: $\boldsymbol{\mu}, \boldsymbol{\sigma} = \phi(\mathbf{h})$;

Step 2. Uncertainty Decomposition:
3. Estimate Aleatoric noise: $\sigma_{ale} = \text{Softplus}(\psi_{ale}(\boldsymbol{\sigma}))$;
4. Estimate Epistemic ambiguity via MC sampling: $\sigma_{epi} = \text{Var}(\{\psi_{epi}^{(k)}(\boldsymbol{\mu})\}_{k=1}^{K})^{1/2}$;

Step 3. Hybrid Distance Attention:
5. Compute modulation $\mathcal{R}_{ij}$ based on uncertainty differences;
6. Calculate hybrid attention scores: $A_{ij} = \alpha \cdot \left(\exp(-D_{ij}^{bil}) + \mathcal{R}_{ij}\right) + (1-\alpha)\cdot \exp(-\mathcal{W}_2^2)$;
7. Aggregate features: $\mathbf{z}_{context} = \text{Softmax}(\mathbf{A}) \cdot \mathbf{V}(\boldsymbol{\mu}, \boldsymbol{\sigma})$;

Step 4. Task-Specific Optimization:
8. Execute task-specific mapping: $\hat{\mathbf{y}} = \phi_{task}(\mathbf{z}_{context})$;
9. Update Θ by minimizing: $\mathcal{L}_{total} = \mathcal{L}_{task} + \sum \lambda_i \mathcal{L}_{reg,i}$.

A.3 Neural Statistic Initializer

Here, we provide the technical details for the Neural Statistic Initializer used to anchor the stochastic manifold. The primary goal of this module is to transition the model from a random state to a data-informed starting point and provide neural basis priors for the Uncertainty Decomposition and the Bilinear distance attention modules.

A.3.1 Statistical Estimation

Before training, we perform a single pass over a subset of the training data $\mathcal{D}_{train}$ to compute class-specific and global statistics. For each neural channel $c \in \{1, \ldots, C\}$, we calculate two neural statistics metrics.

Selectivity Index. We define the neuron selectivity S_c based on the variance of task-category means. A high S_c indicates that channel c provides high discriminative power across different linguistic units:

$$S_c = \text{Norm}\left(\text{Std}_{t\in\mathcal{T}}(\mu_{t,c})\right), \tag{19}$$

where $\mu_{t,c}$ is the mean activity of channel c for task category $t \in \mathcal{T}$, and $\text{Norm}(\cdot)$ scales the values to $[0, 1]$.

Global Noise Floor. We compute the average standard deviation $\bar{\sigma}_{data}$ across all samples to estimate the intrinsic aleatoric noise of the recording sessions.

A.3.2 Component-wise Initialization Logic

The initializer maps these statistics to the model's parameters to bypass the unstable early phases of stochastic optimization:

Aleatoric Uncertainty Prior. The bias of the final layer in the Aleatoric Net is initialized to $\log(\bar{\sigma}_{data} + \epsilon)$. This forces the model to start with a noise estimate that matches the empirical variance of the dataset, preventing the stochastic gradients from exploding.

Epistemic Uncertainty Prior. The Epistemic Net is initialized based on both inverse selectivity and task frequency. For a task category t with count N_t, the initial epistemic prior is scaled by $(1 - S_c) \times \frac{1}{\sqrt{N_t}}$. This ensures the model starts with higher "caution" (epistemic uncertainty) for rare task categories or noisy, non-selective channels.

Bilinear Distance Weights (W_μ). We initialize the learnable weights of the Mahalanobis-like distance using the selectivity index S_c. By setting $\log W_\mu^h \approx \log(S_c)$, the Hybrid Distance Attention is biased to attend to highly informative neural channels from the first epoch, significantly accelerating convergence.

Modulation Coefficients. The learnable log-coefficients for aleatoric (α_{ale}) and epistemic (α_{epi}) modulation are initialized to -0.693 (≈ 0.5) and -1.204 (≈ 0.3) respectively, providing a balanced starting point for the dynamic reliability filtering.

A.4 Task-Specific Decoder Module

Linguistic Unit Decoding. For standard linguistic-unit tasks, such as phoneme or letter decoding, we append an MLP head that maps features to labels via Cross-Entropy loss:

$$\mathcal{L}_{task} = -\frac{1}{T}\sum_{t=1}^{T} \mathbf{y}_t \log(\hat{\mathbf{y}}_t). \tag{20}$$

Future Step Forecasting. To facilitate low-latency intent detection, the model predicts k subsequent units:

$$\mathcal{L}_{task} = -\frac{1}{K}\sum_{j=1}^{k} \mathbf{y}_{T+j} \log(\hat{\mathbf{y}}_{T+j}). \tag{21}$$

Neural Activity Imputation. In scenarios where neural recordings are corrupted or partially missing due to electrode failure, the model reconstructs masked inputs $\mathbf{X}$ via MSE loss:

$$\mathcal{L}_{task} = \frac{1}{|M|}\sum_{t\in M} ||\mathbf{x}_t - \hat{\mathbf{x}}_t||^2, \tag{22}$$

where M is the set of masked indices.

B Experimental Details

B.1 Dataset Details

Public Datasets. As described in Section 4.1, we evaluate the proposed method on three publicly available invasive datasets that provide intracortical neural recordings during attempted speech and handwriting tasks. Here, we give their details and download links:

- **Speech I**[4] Willett et al. [2023]. In the Speech I dataset, the neural activity was recorded from two microelectrode arrays implanted in the participant's ventral motor cortex (each array includes 64 channels), capturing binned threshold crossings and spike band power during the attempted production of complete English sentences. The participant, who had amyotrophic lateral sclerosis and was unable to produce intelligible speech, was visually cued to attempt to speak thousands of distinct sentences. The dataset contains neural activity associated with approximately 12,000 sentences recorded from 24 sessions.
- **Speech II**[5] Card et al. [2024]. In the Speech II dataset, neural activity was recorded from a 256-electrode intracortical array implanted in the left precentral gyrus of an ALS patient during the prompted speech task. The participant was cued to attempt to speak a variety of sentences in structured sessions. Neural signals include high-frequency threshold crossings and multiunit activity. The dataset contains neural data and sentence labels from over 11,000 prompted speech task trials, recorded from 45 data collection sessions.
- **Handwriting**[6] Willett et al. [2021]. In the Handwriting dataset, neural recordings were obtained from two microelectrode arrays (each with 96 channels) implanted in the hand area of the participant's motor cortex. During data collection, the participant was visually cued to attempt to handwrite complete English sentences by imagining pen movements on paper, even though the hand was functionally paralyzed. The dataset comprises neural activity for 1,000 sentences covering 40,000+ characters, recorded over approximately 10.7 hours of experimental time and including 11 sessions. Each trial captures multi-channel spiking thresholds and firing rate features time-aligned to cue onset and attempted letter production.

Data Preprocessing. For each speech dataset, we train five CTC transcription models with different random seeds on the full labeled corpus used to construct the unit-aligned dataset. The phoneme boundaries obtained from the five models are averaged to define the preprocessing boundaries. Sentences with high disagreement among the alignment models are excluded, and all downstream methods use the same retained segments. Within each retained phoneme segment, neural features are pooled to obtain one feature vector per phoneme, producing a sequence of shape $[N, C]$, where N is the number of units and C is the feature dimension. For handwriting, letter segments are constructed from cue timing and the known character sequences, as described above.

Table 6 summarizes the key characteristics of each dataset, including some basic information, the number of valid trials after preprocessing, and the decoding tasks supported.

Table 6: Summary of preprocessed datasets used in this study. All datasets are based on intracortical recordings during attempted speech or handwriting tasks, and are preprocessed and standardized into the same format.

Dataset	Paradigm	Target	Channels	Sessions	Data Scale	Decoding Task
Speech I	Attempted speech	39 phonemes	128	24	9647 sentences	Units decoding, Forecasting, Imputation
Speech II	Attempted speech	39 phonemes	256	45	8015 sentences	Units decoding, Forecasting, Imputation
Handwriting	Imagined handwriting	26 letters	192	11	730 sentences	Units decoding, Imputation

B.2 Training Details

For the random-split unit-aligned experiments, we use an 8:1:1 training/validation/test partition, with validation used for hyperparameter selection and early stopping and results reported on the held-out test set. Session-based evaluations use the partitions described in Appendix C.1. The

[4] https://datadryad.org/dataset/doi:10.5061/dryad.x69p8czpq

[5] https://datadryad.org/dataset/doi:10.5061/dryad.dncjsxm85

[6] https://datadryad.org/dataset/doi:10.5061/dryad.wh70rxwmv

released-validation and official hidden-test evaluations follow the separate benchmark protocols described in Appendix C.2 and Table 2.

To ensure fair and thorough comparison, we perform systematic hyperparameter tuning for all baselines and our model using Ray Tune with the ASHA scheduler. Hyperparameters are tuned independently for each dataset and each task setting. The search space covers learning rate, model depth and width, and task-specific regularization parameters where applicable. For each method, the configuration that achieves the best validation performance is selected for final evaluation.

All models are optimized using the Adam optimizer. For unit-decoding and future-step-forecasting tasks, early stopping is based on validation accuracy. For the imputation task, validation loss is used as the stopping criterion. In all cases, the patience for early stopping is set to 20 epochs. The original unit-decoding, forecasting, and imputation comparisons are repeated with five different random seeds, and their tables report mean performance and standard deviation across these runs. Additional session-based and benchmark results use the reporting conventions specified in their respective tables.

B.3 Metric Details

We evaluate model performance using task-appropriate metrics that reflect decoding accuracy and reconstruction quality.

For the unit-decoding task, we adopt the **Accuracy (ACC)** metric to evaluate decoding performance (39 phonemes and 26 letters), which measures the proportion of correctly predicted units in a sentence. The metric is defined as:

$$\text{ACC}_{\text{decoding}} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i = y_i), \tag{23}$$

where $\mathbb{I}(\cdot)$ is the indicator function, y_i and $\hat{y}_i$ represent the true and predicted label, and N is the total number of samples.

For the future-step-forecasting task, given a fixed window of historical neural activity $\mathbf{X}_{1:T}$, the model is required to predict the phoneme sequence for the subsequent k time steps. The task metric is evaluated using the average Top-1 accuracy across all predicted steps:

$$\text{ACC}_{\text{forecasting}} = \frac{1}{k}\sum_{i=1}^{k}\mathbb{I}(\hat{y}_{T+i} = y_{T+i}), \tag{24}$$

where $\hat{y}_{T+i}$ and y_{T+i} represent the predicted and ground-truth phonemes at future step i, respectively.

For the neural-imputation task, we report both **Mean Squared Error (MSE)** and **Mean Absolute Error (MAE)** between the predicted and ground-truth neural activity:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\|\hat{x}_i - x_i\|_2^2\,, \tag{25}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\|\hat{x}_i - x_i\|_1\,, \tag{26}$$

where x_i and $\hat{x}_i$ denote the ground-truth and imputed neural activity, respectively.

C Additional Experiments

C.1 Cross-session Evaluation

Given that intracortical recording datasets typically span multiple days or sessions, cross-session evaluation is crucial for assessing the long-term stability and generalization of BCI systems. We evaluated our framework and baselines on the linguistic-unit decoding task across three datasets using two sequence lengths ($T \in \{20, 50\}$).

The datasets were partitioned into non-overlapping sessions for training, validation, and testing. For instance, in the **Handwriting** dataset, the training set comprised sessions $\{0, 1, 2, 4, 5, 7, 8, 9\}$,

the validation set consisted of session $\{3\}$, and testing was conducted on session $\{6\}$. All other training hyperparameters and settings remained the same as in the main experiments to ensure a fair comparison.

As summarized in Table 7, our framework achieves the highest cross-session accuracy, demonstrating superior robustness to inter-session neural variability. For example, on **Speech I**, our model achieves a peak accuracy of 80.6% ($T = 50$), surpassing the second-best Transformer by 2.8% and significantly outperforming TCN by 9.6%, highlighting our method's advantage in handling the intricate, time-varying contexts of natural speech. Even on **Handwriting**, where data is more structured but session-specific noise is prevalent, our approach maintains a clear lead over attention-based models such as the Conformer (67.3% vs. 61.4%), demonstrating that explicit uncertainty modeling effectively prevents overfitting to session-dependent artifacts and ensures stable decoding performance in low-data, high-drift scenarios.

Table 7: Cross-session speech and handwriting decoding accuracy (%, ↑) on intracortical recording datasets under different sequence lengths T. Mean and standard deviation (Std in footnotesize) are reported. Best results are in **bold** and the second underlined.

Model	Speech I		Speech II		Handwriting	
	T=20	T=50	T=20	T=50	T=20	T=50
LSTM	52.9±0.3	55.8±0.2	71.0±0.3	72.0±0.2	57.1±1.1	61.2±1.4
TCN	67.2±0.2	71.0±0.2	80.3±0.5	82.3±0.4	63.4±1.7	66.4±0.8*
Transformer	73.5±0.3	77.8±0.3	85.2±0.4	86.5±0.2	56.2±2.0	58.4±1.2
Conformer	71.0±0.5	75.8±0.4	82.1±0.5	85.0±0.4	58.9±0.8	61.4±1.6
iTransformer	41.4±0.6	38.5±1.3	55.4±1.5	51.6±2.0	45.1±13.2	37.3±9.8
Mamba	41.7±0.4	44.7±0.3	65.6±0.7	67.4±0.4	53.6±0.5	57.1±1.1
TimesNet	70.5±0.3	74.9±0.2	81.9±0.4	84.0±0.3	62.3±1.0	65.5±2.1
TimeMixer	51.0±0.5	55.1±0.5	61.8±0.5	64.2±0.5	53.6±1.6	58.6±1.4
Ours	**76.1**±0.2	**80.6**±0.3	**86.8**±0.3	**88.1**±0.3	**64.9**±1.5	**67.3**±1.0

Repeated Session-Held-Out Evaluation. We repeat the session-held-out evaluation using four test sessions for Speech I, five for Speech II, and three for Handwriting. Each fold holds out one complete session for testing and another complete session for validation. Table 8 summarizes the results over the held-out sessions, with standard deviations computed across random seeds. Each dataset contains one participant, so this evaluation assesses within-subject cross-session robustness.

Table 8: Repeated session-held-out decoding ACC (%, ↑). The ± values denote standard deviations across random seeds.

Dataset	Test sessions	Best baseline	Ours
Speech I	4	77.8 ± 0.4	$\mathbf{80.4 \pm 0.3}$
Speech II	5	86.5 ± 0.3	$\mathbf{88.0 \pm 0.4}$
Handwriting	3	66.2 ± 1.0	$\mathbf{67.0 \pm 1.1}$

Chronological Early-to-Late Evaluation. We additionally partition sessions in recording order, training on earlier sessions and evaluating on later sessions. For Speech I, we use the first 12 sessions for training, the next two for validation, and the final ten for testing. For Speech II, the corresponding session counts are 24, four, and 17. For Handwriting, we use the first six sessions for training, the seventh for validation, and the final four for testing. Table 9 shows consistent improvements over the best deterministic baseline across the three datasets.

Representation Stability. We quantify representation stability using same-phoneme cross-session centroid drift, within-phoneme variance, and silhouette score. Table 10 reports centroid drift separately for Speech I and Speech II. Within-phoneme variance and silhouette score are averaged over these two speech datasets. Our method reduces cross-session drift and within-phoneme variance, and

Table 9: Chronological early-to-late decoding ACC (%, ↑). Train/Val/Test gives the numbers of successive sessions in recording order.

Dataset	Train/Val/Test	Best baseline	Ours
Speech I	12/2/10	73.8	**77.1**
Speech II	24/4/17	82.0	**85.6**
Handwriting	6/1/4	63.9	**66.8**

improves silhouette score. These results provide quantitative evidence for more stable, discriminative phoneme representations across sessions.

Table 10: Representation stability on the speech datasets. Drift is reported separately for each dataset. Variance and silhouette are averaged over Speech I and Speech II.

Model	Speech I drift ↓	Speech II drift ↓	Variance ↓	Silhouette ↑
Transformer	0.312	0.286	0.421	0.184
TMT	0.274	0.241	0.376	0.221
Ours	0.203	0.188	0.314	0.276
OursTMT	**0.176**	**0.161**	**0.292**	**0.304**

C.2 Scalability of Variability-Aware Decoding

Our framework provides a neural representation modeling method from the perspective of neural variability to enhance unit-level language decoding capability, and can be naturally combined with existing language-model-based decoding pipelines. Specifically, all methods are evaluated under the same post-processing setting, where +LM denotes a 5-gram language model and +LLM denotes large-language-model correction following Willett et al. [2023]. The final decoded sentences are evaluated by CER and WER.

Table 11 compares our method with the BiGRU backbone [Willett et al., 2023], and Table 12 further compares it with the stronger Time-Masked Transformer (TMT) backbone [Feghhi et al., 2025]. Results show that our method substantially improves over BiGRU and achieves performance comparable to TMT without using any data augmentation, while our method is slightly worse than TMT after LLM correction. To further validate scalability, we replace our encoder backbone with TMT, forming the enhanced variant OursTMT. Compared with TMT, OursTMT consistently improves unit decoding, +LM decoding, and +LLM correction, indicating that our variability-aware modeling is complementary to strong decoding backbones. Finally, we clarify that these results are not directly comparable to the original paper, since our standardized segmentation-and-pooling preprocessing produces unit-level aligned data with much lower temporal resolution than the raw recordings. Nevertheless, all methods here use the same preprocessed data, training protocol, and post-processing pipeline, ensuring a strictly fair comparison within our setting.

Table 11: Comparison with BiGRU baseline on Speech I under the same post-processing settings. ACC (%, ↑), CER (%, ↓), and WER (%, ↓) are reported.

Model	Decoding	+LM		+LLM	
	ACC↑	CER↓	WER↓	CER↓	WER↓
BiGRU	72.4±0.2	24.7±0.2	30.2±0.2	22.5±0.2	26.7±0.3
Ours	**80.1**±0.5	**17.4**±0.4	**21.3**±0.4	**16.4**±0.3	**19.9**±0.3

We additionally evaluate sequence decoding using the released training and validation partitions of the speech benchmarks. Within each dataset, the paired comparison uses the same TMT temporal backbone, preprocessing and filtering, output-head architecture, hyperparameter budget, and fixed 5-gram LM and lexicon. Our variant inserts stochastic projection, variability estimation, and HDA/routing between the temporal encoder and the output head. Sequence PER is computed from phoneme edit distance as $100(S + D + I)/N_{\mathrm{ph}}$, where S, D, and I denote substitutions, deletions, and insertions, and N_{ph} is the number of reference phonemes. Table 13 reports the paired released-validation results.

Table 12: Comparison with the SOTA Time-Masked Transformer (TMT) and OursTMT variant on Speech I under the same post-processing settings. ACC (%, ↑), CER (%, ↓), and WER (%, ↓) are reported.

Model	Decoding	+LM		+LLM	
	ACC↑	CER↓	WER↓	CER↓	WER↓
TMT	79.5±0.2	17.5±0.2	21.3±0.3	16.1±0.2	19.5±0.2
Ours	80.1±0.5	17.4±0.4	21.3±0.4	16.4±0.3	19.9±0.3
OursTMT	**81.7±0.1**	**16.4±0.3**	**20.2±0.6**	**14.9±0.2**	**17.8±0.4**

Table 13: Sequence-level evaluation on the released validation partitions. PER and WER are percentages (↓). WER uses the same fixed 5-gram LM and lexicon within each paired comparison.

Dataset	PER		WER	
	TMT	OursTMT	TMT	OursTMT
Speech I	15.4	**14.8**	17.1	**16.4**
Speech II	9.6	**9.0**	7.1	**6.7**

C.3 Complete Results on Future Step Forecasting

We provide detailed performance results for the future-step-forecasting task across two speech datasets in Table 14. Our proposed framework outperforms all baseline models across all look-ahead horizons ($K \in \{2, 3, 4, 5\}$) and both datasets. Specifically, on **Speech II** with $K = 2$, our model achieves a Top-1 accuracy of 87.0%, surpassing the strongest baseline (LSTM at 79.6%) by a significant margin of 7.4%. Even as the prediction difficulty increases with larger K, our model maintains a clear lead. For instance, at $K = 5$ on **Speech I**, our model reaches 47.7%, while the best-performing TimesNet only achieves 41.8%. Furthermore, our method demonstrates greater robustness in long-horizon forecasting. On **Speech I**, as the forecasting horizon increases from 2 to 5, the performance lead of our method over TCN and TimesNet is substantially amplified (from 2% to 7%).

Table 14: Forecasting accuracy (%, ↑) on two speech datasets under different future steps K. Mean and standard deviation (Std in footnotesize) are reported over five runs. Best results are in **bold** and the second underlined.

Model	**Speech I**				**Speech II**			
	K=2	K=3	K=4	K=5	K=2	K=3	K=4	K=5
LSTM	82.1±0.4	65.2±0.6	50.0±0.8	41.3±0.6	79.6±0.5	60.3±0.9	50.3±0.8	46.1±0.7*
TCN	83.5±0.4	63.4±0.4	48.2±1.0	39.2±0.4	78.9±0.6	60.3±0.5	48.1±0.5	42.8±1.1
Transformer	82.2±0.6	64.9±0.4	49.8±0.5	40.7±0.4	78.1±0.8	61.2±2.0	50.3±0.4	44.2±0.7
Conformer	81.1±0.5	61.5±0.8	46.7±0.6	39.1±0.3	74.8±0.7	54.8±0.7	45.9±0.6	41.7±1.0
iTransformer	49.5±0.0	38.4±2.9	31.7±2.2	29.1±0.5	49.6±0.2	33.1±0.0	28.8±0.2	30.1±0.3
Mamba	81.8±0.6	60.0±0.8	44.8±0.5	37.2±0.4	69.5±5.1	50.2±3.6	43.1±1.6	39.6±2.3
TimesNet	83.5±0.3	65.2±0.6	51.4±1.0	41.8±0.5	79.2±1.1	62.4±1.0	51.6±0.9	45.6±0.9
TimeMixer	77.5±0.7	58.5±0.3	45.0±1.4	38.9±1.1	74.9±0.5	53.2±1.5	45.0±0.9	40.0±0.6
Ours	**85.3**±1.3	**72.2**±0.7	**53.1**±1.4	**47.7**±1.1	**87.0**±0.7	**66.6**±1.4	**56.3**±1.4	**46.6**±1.3

C.4 Complete Results on Neural Activity Imputation

In this section, we present a comprehensive evaluation of our framework for neural activity imputation, which simulates real-world BCI challenges such as electrode failure or signal dropout. Tables 15 and 16 provide the MSE and MAE results for two mask ratios ($R \in \{0.2, 0.5\}$) across three datasets. Our framework consistently achieves the lowest reconstruction error, demonstrating its superior ability to capture the underlying latent manifold even under heavy data corruption. For example, across all three datasets, our method maintains a larger performance lead over the second-best method at higher mask ratios (e.g., 0.0018 for $R = 0.5$ vs. 0.0010 for $R = 0.2$ on **Speech II**, Table 15).

Table 15: Imputation loss (MSE, ↓) on intracortical recording datasets under different mask ratio R. Mean and standard deviation (Std in footnotesize) are reported over five runs. Best results are in **bold** and the second underlined.

Model	Speech I		Speech II		Handwriting	
	R=0.2	R=0.5	R=0.2	R=0.5	R=0.2	R=0.5
LSTM	0.1626±0.0009	0.1679±0.0003	0.1219±0.0006	0.1285±0.0008	0.0214±0.0003	0.0213±0.0004
TCN	0.1515±0.0010	0.1573±0.0003	0.1126±0.0007	0.1182±0.0007	0.0213±0.0002	0.0211±0.0003
Transformer	0.1572±0.0216	0.1517±0.0005	0.1516±0.0206	0.1341±0.0226	0.0211±0.0003	0.0207±0.0004
Conformer	0.1553±0.0010	0.1592±0.0005	0.1175±0.0007	0.1233±0.0009	0.0211±0.0002	0.0207±0.0003
iTransformer	0.1838±0.0043	0.1771±0.0083	0.1572±0.0024	0.1544±0.0016	0.0204±0.0003	0.0203±0.0005
Mamba	0.1711±0.0008	0.1753±0.0005	0.1399±0.0022	0.1461±0.0077	0.0221±0.0002	0.0223±0.0005
TimesNet	0.1515±0.0010	0.1558±0.0004	0.1144±0.0006	0.1198±0.0006	0.0206±0.0002	0.0207±0.0003
TimeMixer	0.1514±0.0008	0.1568±0.0004	0.1174±0.0007	0.1239±0.0010	0.0204±0.0002	0.0200±0.0003
Ours	**0.1499**±0.0005	**0.1496**±0.0003	**0.1116**±0.0006	**0.1164**±0.0007	**0.0200**±0.0001	**0.0195**±0.0001

Table 16: Imputation loss (MAE, ↓) on intracortical recording datasets under different mask ratio R. Mean and standard deviation (Std in footnotesize) are reported over five runs. Best results are in **bold** and the second underlined.

Model	Speech I		Speech II		Handwriting	
	R=0.2	R=0.5	R=0.2	R=0.5	R=0.2	R=0.5
LSTM	0.3077±0.0005	0.3126±0.0003	0.2572±0.0006	0.2636±0.0008	0.1091±0.0008	0.1084±0.0007
TCN	0.2975±0.0004	0.3032±0.0002	**0.2486**±0.0007*	0.2540±0.0007*	0.1090±0.0006	0.1084±0.0007
Transformer	0.3026±0.0194	0.2979±0.0004	0.2866±0.0197	0.2697±0.0220	0.1083±0.0006	0.1075±0.0007
Conformer	0.3012±0.0004	0.3049±0.0003	0.2537±0.0009	0.2593±0.0006	0.1087±0.0005	0.1076±0.0007
iTransformer	0.3279±0.0032	0.3222±0.0069	0.2931±0.0029	0.2894±0.0019	0.1069±0.0008	0.1067±0.0009
Mamba	0.3158±0.0006	0.3197±0.0004	0.2756±0.0021	0.2819±0.0072	0.1100±0.0005	0.1100±0.0008
TimesNet	0.2973±0.0006	0.3015±0.0003	0.2502±0.0009	0.2552±0.0007	0.1074±0.0005	0.1074±0.0007
TimeMixer	0.2977±0.0004	0.3029±0.0002	0.2543±0.0009	0.2606±0.0008	0.1071±0.0007	0.1063±0.0007
Ours	**0.2961**±0.0007	**0.2955**±0.0004	**0.2486**±0.0009	**0.2536**±0.0009	**0.1060**±0.0004	**0.1046**±0.0005

C.5 Additional Hyperparameter Sensitivity Analysis

In this section, we investigate the sensitivity of our model to the loss weights associated with the two uncertainty components (λ_{ale} and λ_{epi}) on the two speech datasets (Figure 5). This analysis aims to identify the optimal balance between noise suppression and contextual ambiguity modeling.

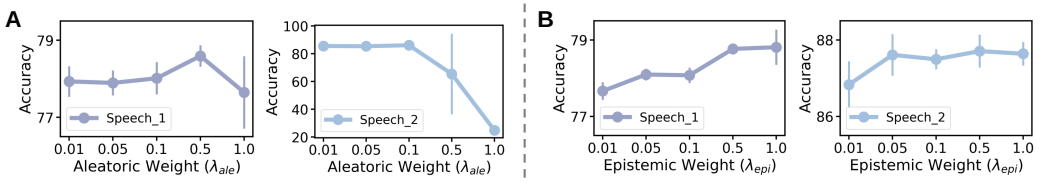

Figure 5: Hyperparameter sensitivity analysis on speech datasets.

Sensitivity to Aleatoric Weight (λ_{ale}). As shown in Figure 5A, the impact of the aleatoric weight is highly dataset-dependent. For **Speech I**, performance gradually peaks at $\lambda_{ale} = 0.5$, whereas **Speech II** exhibits a sharp decline in accuracy if the weight exceeds 0.1, crashing from roughly 86% to 23% when $\lambda_{ale} = 1.0$. This suggests that while suppressing intrinsic noise is beneficial, excessive aleatoric filtering risks removing essential task-related neural variance.

Sensitivity to Epistemic Weight (λ_{epi}). Figure 5B demonstrates that modeling epistemic uncertainty is a more robust and beneficial strategy. Across both datasets, accuracy either increases steadily or remains stable as λ_{epi} grows to 0.5, with **Speech II** reaching a high-performance of 87.8% at

Table 17: Comparison with uncertainty baselines on phoneme forecasting accuracy (%, ↑) under different future steps K.

Model	Uncertainty		Speech I		Speech II	
	Estimate	Modulation	K=2	K=4	K=2	K=4
MCDropout	✓	×	83.0±1.0	47.5±0.5	78.2±0.4	47.2±0.4
Ensembles	✓	×	85.2±0.4*	50.1±0.6	80.4±0.3	50.4±0.7
OursTVD	✓	×	82.7±0.6	48.4±0.8	78.2±0.7	49.1±1.7
Ours	✓	✓	**85.3**±1.3	**53.1**±1.4	**87.0**±0.7	**56.3**±1.4

$\lambda_{epi} \geq 0.5$. Unlike the aleatoric part, increasing the penalty on contextual ambiguity does not lead to catastrophic performance drops, confirming the stability of our epistemic decomposition module.

C.6 Comparison with Uncertainty Baselines

Generic uncertainty baselines test whether predictive uncertainty alone is sufficient for language BCI decoding. Direct-decoding results are reported in the main text (Table 4). For forecasting, MC-Dropout and Ensembles estimate predictive uncertainty, while OursTVD uses a total-variance-decomposition variant without the proposed attention modulation. Table 17 shows that the full model performs best, indicating that uncertainty-modulated attention remains useful beyond direct decoding.

C.7 Selectivity Analysis of Learned Variability

Controlled perturbations test whether the learned components respond differently to reliability loss and linguistic ambiguity. We inject graded Gaussian noise and compare phoneme boundary-defined linguistic contexts, as summarized in Figure 6. The results show stronger aleatoric responses to injected noise and stronger epistemic responses to phoneme boundary structure. This pattern supports using the two estimates as task-relevant reliability and ambiguity signals.

Table 18 reports the ordered noise-severity analysis. Lower dB values correspond to stronger injected noise, so the association is described in terms of noise severity. The monotonic rise of σ_{ale} under stronger perturbation indicates that the aleatoric component tracks degraded neural reliability.

Table 19 reports the context-rarity-level (CRL) analysis. The epistemic component increases as the context becomes rarer (frequency of phoneme triples), whereas the aleatoric component remains comparatively stable. This result is consistent with interpreting σ_{epi} as a representation-level ambiguity signal.

Table 20 summarizes external proxy alignment with noise severity and CRL, complementing the proxy-value results in Tables 18–19.

Table 18: Uncertainty statistics under ordered noise-severity levels. Lower dB values correspond to stronger injected noise.

Noise severity	$\bar{\sigma}_{\text{ale}}$	$\Delta\sigma_{\text{ale}}$ (%)	$\bar{\sigma}_{\text{epi}}$	$\Delta\sigma_{\text{epi}}$ (%)	σ_{ale} (std)	σ_{epi} (std)
Clean	0.243	0	0.298	0	0.108	0.131
30 dB	0.268	10.3	0.291	2.3	0.112	0.129
20 dB	0.327	34.6	0.303	1.7	0.117	0.133
15 dB	0.406	67.1	0.296	-0.7	0.123	0.130
10 dB	0.497	104.5	0.314	5.4	0.131	0.137
5 dB	0.593	144.0	0.339	13.8	0.139	0.142
0 dB	**0.714**	**193.8**	**0.371**	**24.5**	**0.143**	**0.148**

In Table 21, we further constructed three externally controlled subsets: high-noise, high-ambiguity (high-CRL), and a clean & easy set, and compared the full model with two ablated variants. The degradation patterns of the two variants are clearly selective. Under high-noise, removing σ_{ale} causes a much larger drop than removing σ_{epi} (63.6 vs. 69.9), while under high-ambiguity, removing σ_{epi} hurts more (63.7 vs. 68.7). This suggests that the two uncertainties have distinct, non-interchangeable roles.

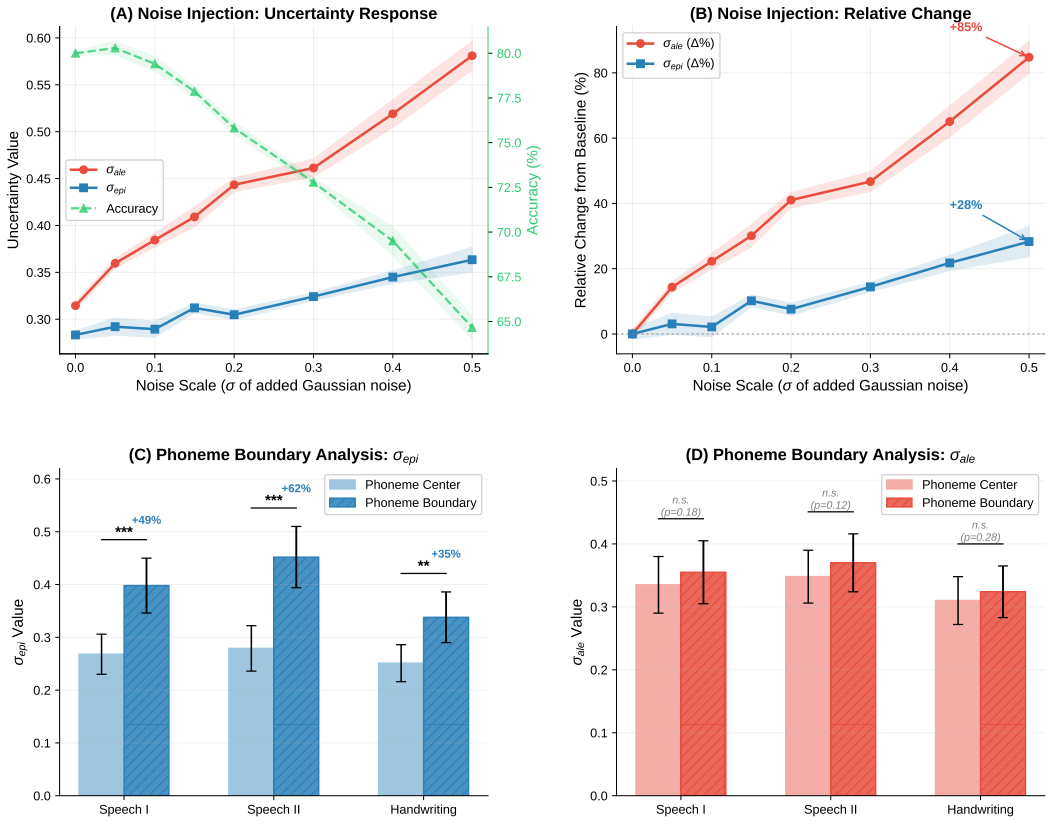

Figure 6: Controlled perturbation experiments validating the selective responses of learned uncertainty estimates. (A) Absolute values of σ_{ale} and σ_{epi} and decoding accuracy under increasing Gaussian noise injection on Speech I. (B) Relative change from baseline, showing that σ_{ale} increases up to +85% while σ_{epi} responds more moderately (+28%). (C) σ_{epi} at phoneme boundaries vs. centers across three datasets, showing significant boundary elevation (Speech I: +49%, Speech II: +62%, Handwriting: +35%). (D) σ_{ale} at phoneme boundaries vs. centers, showing no significant boundary effect.

Table 19: Uncertainty statistics under different context rarity levels (CRL).

CRL	$\bar{\sigma}_{\text{ale}}$	$\Delta\sigma_{\text{ale}}$ (%)	$\bar{\sigma}_{\text{epi}}$	$\Delta\sigma_{\text{epi}}$ (%)	σ_{ale} (std)	σ_{epi} (std)
Fixed	0.253	0	0.218	0	0.111	0.096
Common	**0.258**	**2.0**	0.251	15.1	**0.113**	0.108
Medium	0.249	-1.6	0.289	32.6	0.109	0.119
Rare	0.255	0.8	0.327	50.0	0.112	0.131
Rarest	0.252	-0.4	**0.372**	**70.6**	0.110	**0.141**

Table 20: Correlation of learned uncertainty components with externally controlled noise severity and context rarity.

Component	Noise severity		CRL	
	r	R^2	r	R^2
σ_{ale}	**0.58**	**33.6%**	0.02	0.04%
σ_{epi}	0.13	7.0%	**0.38**	**14.4%**

Matched Same-Phoneme Analysis. We compare rare and common neighboring contexts while fixing the central phoneme and matching session, phoneme frequency, trial-level neural SNR, σ_{ale}, and baseline prediction entropy/confidence. Table 22 shows that rare contexts remain associated with

Table 21: Condition-specific decoding accuracy under externally controlled high-noise, high-ambiguity, and clean & easy subsets.

Variant	High-noise	High-ambiguity	Clean & easy
Ours (w/o Ale)	63.6	<u>68.7</u>	88.5
Ours (w/o Epi)	<u>69.9</u>	63.7	<u>88.9</u>
Ours (Full)	**71.2**	**70.5**	**89.3**

higher σ_{epi} after these controls. The corresponding changes in σ_{ale} are small, and the rare-common accuracy gaps are 0.9 and 1.0 percentage points on Speech I and Speech II, respectively. These results provide functional evidence for context-related sensitivity of the learned proxy.

Table 22: Matched same-phoneme analysis. Uncertainty changes compare rare neighboring contexts with common contexts. Accuracy gaps are measured in percentage points (pp).

Dataset	$\Delta\sigma_{epi}$	$\Delta\sigma_{ale}$	ACC gap (pp)
Speech I	$+17.8\%$	$+1.6\%$	0.9
Speech II	$+21.3\%$	$+2.1\%$	1.0

Decomposition and Deterministic Controls. The shuffled and swapped controls preserve the encoder, both uncertainty heads, HDA, loss weights, parameter count, and training schedule. The shuffled control randomly permutes the detached margin-based target across training segments, preserving its distribution and disrupting its sample correspondence. The swapped control exchanges the entropy and margin targets assigned to σ_{ale} and σ_{epi}. The single-uncertainty variant collapses the two-component decomposition into one uncertainty estimate. We also include the internal uncertainty-removal ablation, a separately retuned deterministic HDA model, and the best deterministic baseline from Table 1. Table 23 supports the contribution of the assigned decomposition and calibration targets.

Table 23: Additional unit-aligned decoding controls. ACC (%, ↑) is reported; deterministic comparisons use $T = 50$. The shuffled and swapped variants are capacity-matched to the full model.

Variant	Speech I	Speech II	Handwriting
Ours w/o uncertainty heads and routing	73.6	81.9	64.8
Retuned deterministic HDA	76.9	85.2	66.7
Best deterministic baseline, Table 1	78.2	87.5	68.9
Single uncertainty	77.3	85.0	66.3
Epistemic target shuffled	76.8	84.9	65.9
Aleatoric/epistemic targets swapped	76.4	84.2	65.8
Ours, full model	**80.1**	**88.6**	**69.1**

Context Augmentation and Routing. We compare context augmentation and uncertainty-based routing using the same TMT backbone and the common unit-aligned Speech I pipeline. Table 24 includes the TMT baseline, its context-augmentation variant, our variability-aware variant without routing, and the full OursTMT model. The full model achieves the highest ACC and lowest WER among these variants, supporting the additional utility of routing learned variability through attention.

Table 24: Same-TMT-backbone controls on unit-aligned Speech I. ACC (%, ↑) and WER (%, ↓) are reported. All rows use the same LLM post-processing pipeline.

Variant	Aug.	Proxies	Routing	ACC	+LLM WER
TMT-style	No	No	No	79.5	19.5
TMT + context aug.	Yes	No	No	80.0	19.2
OursTMT w/o routing	No	Yes	No	80.6	18.8
OursTMT	No	Yes	Yes	**81.7**	**17.8**

C.8 Additional Calibration Analysis

Calibration analysis evaluates whether confidence estimates are useful for ranking and interpreting decoding outcomes. We report Speech I uncertainty calibration in the main text (Figure 4) and Speech II uncertainty calibration in Figure 7. The curves compare predicted confidence with empirical accuracy across uncertainty bins, providing confidence-utility checks for the learned uncertainty estimates. Notably, our analysis reveals that the relationship between decoding performance and epistemic uncertainty follows a clear negative exponential pattern across both speech datasets (Figure 7B and 4B). The sharp decline in accuracy during the initial stages of increasing epistemic ambiguity suggests that the model is extremely sensitive to contextual uncertainty at its onset. This functional relationship implies that addressing "contextual blur" during these early stages is the critical bottleneck for enhancing overall system performance.

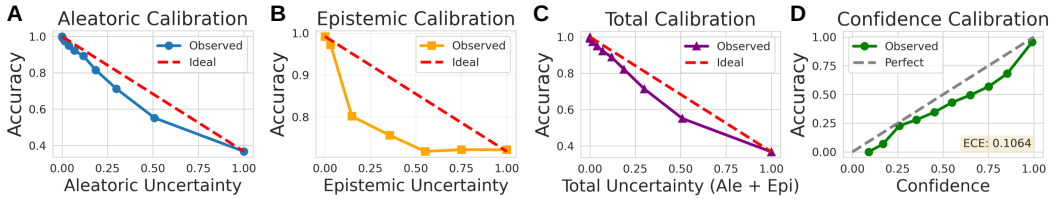

Figure 7: Quantitative calibration of uncertainty estimates for linguistic-unit decoding on Speech II.

We then compare predictive calibration for uncertainty-aware baselines and ablation variants on Speech I and Speech II in Figures 8 and 9. These comparisons separate calibration quality from raw decoding accuracy, which is important because a low ECE can coincide with low confidence and lower accuracy.

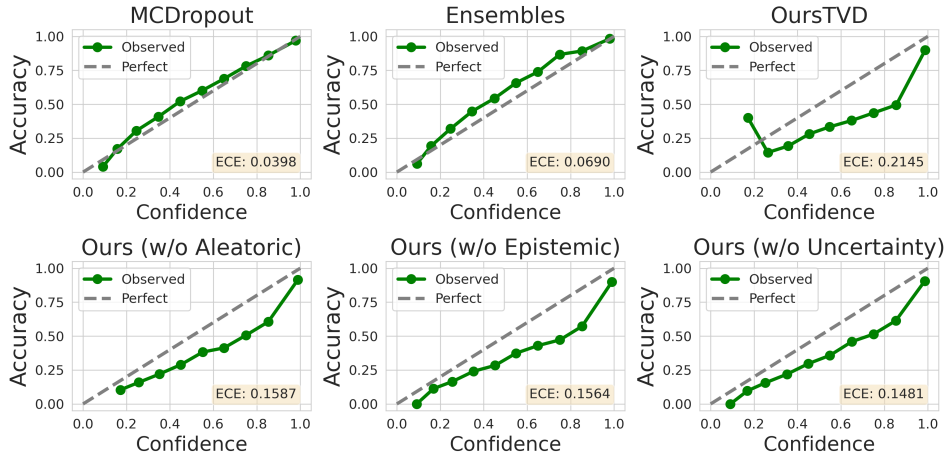

Figure 8: Confidence calibration curves for uncertainty-aware baselines and ablation variants on Speech I.

Table 25 reports ECE and mean confidence, and Table 26 reports ECE, MCE, and ACE. Generic uncertainty baselines can have low ECE but remain substantially less accurate, while ablation variants tend to be over-confident relative to the full model. The full model maintains high decoding accuracy with competitive calibration, supporting the utility of the complete uncertainty pathway.

Numerical Ranges of Proxies and Calibration Targets. Table 27 reports the 5th–95th percentile ranges on held-out segments. These are empirical percentile intervals rather than hard bounds on the learned outputs.

C.9 Reject-Decoding Simulation

Reject decoding examines whether learned uncertainty can rank samples by expected reliability in an offline coverage-accuracy setting. We reject samples whose uncertainty exceeds a threshold and

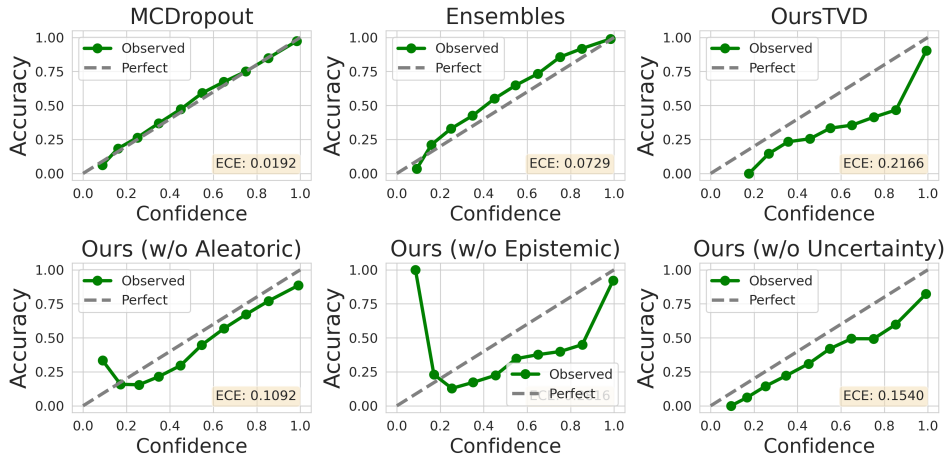

Figure 9: Confidence calibration curves for uncertainty-aware baselines and ablation variants on Speech II.

Table 25: Calibration comparison on the decoding task. ECE (↓) and mean confidence (Conf, ↑) are reported. Sign denotes ACC − Conf: "+" indicates under-confidence and "−" indicates over-confidence.

Model	Speech I			Speech II		
	ECE↓	Conf↑	Sign	ECE↓	Conf↑	Sign
MCDropout	**0.040**	0.579	+	**0.019**	0.715	+
Ensembles	0.069	0.581	+	<u>0.073</u>	0.706	+
OursTVD	0.215	**0.874**	−	0.217	0.937	−
Ours (w/o Ale)	0.159	0.850	−	0.109	0.844	−
Ours (w/o Epi)	0.156	<u>0.871</u>	−	0.282	<u>0.953</u>	−
Ours (w/o Un)	0.148	0.786	−	0.154	0.844	−
Ours (Full)	<u>0.061</u>	0.862	−	0.106	**0.992**	−

Table 26: Calibration metrics comparison on the decoding task. ECE, MCE (Maximum Calibration Error), and ACE (Average Calibration Error) are reported (↓ for all).

Model	Speech I			Speech II		
	ECE↓	MCE↓	ACE↓	ECE↓	MCE↓	ACE↓
MCDropout	**0.036**	**0.076**	**0.033**	**0.019**	**0.048**	**0.016**
Ensembles	0.072	0.116	0.057	<u>0.074</u>	<u>0.104</u>	<u>0.048</u>
OursTVD	0.183	0.338	0.146	0.229	0.372	0.125
Ours (w/o Ale)	0.150	0.266	0.114	0.101	0.134	0.109
Ours (w/o Epi)	0.159	0.289	0.125	0.225	0.406	0.102
Ours (w/o Un)	0.143	0.241	0.120	0.161	0.279	0.164
Ours (Full)	<u>0.061</u>	<u>0.115</u>	<u>0.045</u>	0.106	0.182	0.064

compute accuracy only on the retained subset. Table 28 shows that accuracy increases monotonically as the rejection ratio rises on both speech datasets. The monotonic coverage-accuracy trade-off indicates that learned uncertainty can support offline reliability ranking, and prospective deployment can further evaluate coverage, latency, and user-interaction constraints.

C.10 Visualization of Uncertainty Embedding

To provide a more intuitive understanding of how the model captures neural reliability, we present the t-SNE visualization of the learned uncertainty embeddings in Figure 10. For each dataset (**Speech I** in panel A and **Speech II** in panel B), we randomly sampled 1,000 instances and projected

Table 27: Held-out 5th–95th percentile ranges of learned proxies and detached calibration targets.

Quantity	Speech I	Speech II	Handwriting
σ_{ale}	0.18-0.74	0.16-0.71	0.09-0.52
σ_{epi}	0.12-0.49	0.11-0.50	0.06-0.43
Normalized entropy target	0.10-0.88	0.08-0.91	0.05-0.79
Margin target	0.04-0.49	0.03-0.49	0.05-0.48

Table 28: Reject-decoding simulation. Samples with uncertainty above a threshold are rejected from a sampled batch, and accuracy is computed only on the retained valid samples.

Reject ratio	10%	20%	30%	40%	50%
Speech I	76.9	81.8	86.4	90.4	**93.6**
Speech II	86.8	90.7	93.5	95.4	**96.7**

combined aleatoric and epistemic embeddings into a 2D space. The visualization reveals a clear spatial alignment between the model's predictive performance and the magnitude of uncertainty.

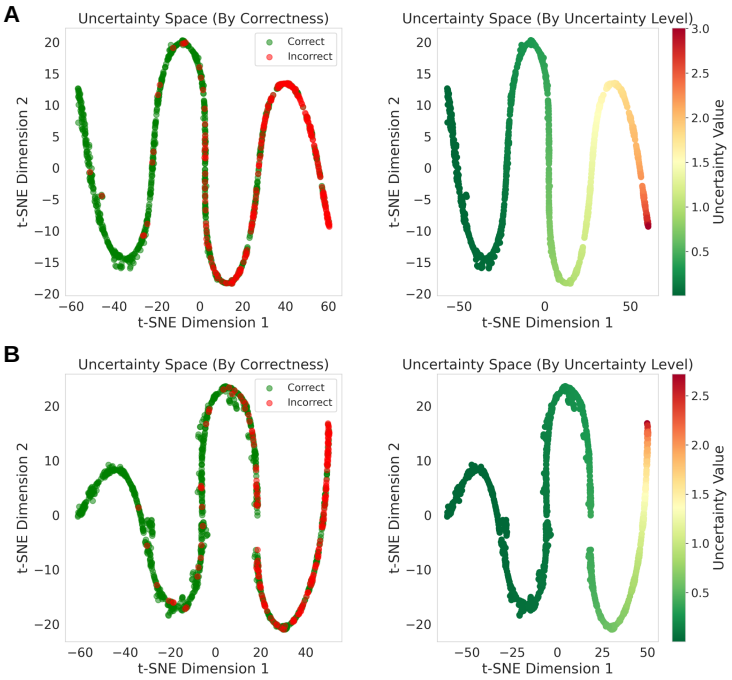

Figure 10: Visualization of the uncertainty embedding on **Speech I** (A) and **Speech II** (B).

Classification Behavior (Left Column). Samples are labeled based on the final classification outcome: green indicates correct decoding, and red indicates incorrect ones. A clear topological structure emerges, with correct predictions occupying stable, dense regions of the manifold, while incorrect predictions are predominantly clustered at the distal "tails" or transition zones.

Uncertainty Gradient (Right Column). When the same manifold is colored by the predicted uncertainty magnitude (from green/low to red/high), we observe a good alignment with the correctness plots. Areas with low uncertainty values (green) correspond to high-accuracy zones, whereas regions with high uncertainty values (yellow and red) are associated with decoding failures.

This visualization confirms that the latent uncertainty manifold effectively encodes the model's confidence in decision-making. The transition from the green, high-density, correct regions to the red, high-uncertainty, erroneous regions validates our earlier findings that the model's performance is negatively correlated with uncertainty. The clear separation between these states demonstrates

that the Uncertainty Decomposition and Hybrid Distance Attention mechanisms have successfully learned to distinguish reliable neural patterns from ambiguous ones and provide effective guidance for downstream tasks.

C.11 Computational Efficiency

We evaluate the computational efficiency of all models by reporting their training speed and model size. All measurements are conducted under a fixed experimental configuration, including the same hardware environment, batch size (16), and sequence length (50), to ensure fair comparison.

Table 29 reports the computational statistics of all models. Although some lightweight models, such as iTransformer and TimeMixer, are highly efficient, they perform poorly on multiple tasks. Our method has a moderate number of parameters and training speed while performing well on most tasks. The reported model size is measured in MB. The timing and FLOP measurements cover the neural-model configuration in this table. The enhanced benchmark system additionally combines ten independently trained decoders and language-model post-processing, whose full training and inference costs are outside these measurements.

Table 29: Computational efficiency comparison under the same experimental configuration (batch size = 16, sequence length = 50).

Model	Model size	Iterations/s	Latency (ms)	Train FLOPs (G)
LSTM	130.0 MB	21.38 it/s	13.6233	1.4938
TCN	98.2 MB	28.31 it/s	3.4710	1.5004
Transformer	49.3 MB	34.73 it/s	4.6232	62.1491
Conformer	29.7 MB	28.12 it/s	12.6726	27.4577
iTransformer	6.54 MB	25.46 it/s	4.0772	20.0976
Mamba	55.1 MB	13.19 it/s	4.6806	66.8204
TimesNet	426.4 MB	5.69 it/s	22.1534	176.3644
TimeMixer	5.10 MB	39.24 it/s	5.1291	2.7773
Ours	32.7 MB	19.65 it/s	7.9304	96.5803

C.12 Robustness to Alignment Perturbations

We evaluate sensitivity to alignment errors by perturbing the CTC-derived boundaries by ± 2 and ± 4 original recording bins. All compared methods use the same perturbed segments. Both speech datasets provide neural features at 20 ms resolution, so these perturbations correspond to ± 40 and ± 80 ms. Table 30 shows that accuracy decreases with larger perturbations, while the proposed method retains its advantage over the best baseline.

Table 30: Sensitivity of unit-aligned decoding to boundary perturbations. ACC is reported in percent.

Dataset	Boundary perturbation	Best baseline	Ours
Speech I	Original	78.2	80.1
	± 2 bins (± 40 ms)	76.8	78.9
	± 4 bins (± 80 ms)	74.5	76.7
Speech II	Original	87.5	88.6
	± 2 bins (± 40 ms)	85.9	87.3
	± 4 bins (± 80 ms)	83.7	85.4

D Hyperparameter Settings

We carefully tune the hyperparameters of all baselines and our method to ensure a fair and competitive comparison. All methods are trained and evaluated under the same data splits and optimization protocol described in the Section B.2.

The search space covers common optimization and architecture-related parameters, including batch size, learning rate, hidden dimension, and number of layers. Unless otherwise specified, the following shared search space is applied to all methods: the learning rate selected from {1e-4, 1e-3, 1e-2}, the hidden dimension selected from {64, 128, 256, 512, 1024, 2048}, and the number of layers from {1, 2, 3, 4, 5, 6, 8}. Below, we briefly summarize the main idea of each method and report the selected hyperparameters for the units decoding task.

D.1 Our Method

Our proposed uncertainty-aware decoding model is tuned using hyperparameter optimization strategies and search spaces that are roughly the same as those of the baseline. In addition, we conduct searches over some key parameters, including the uncertainty modeling. The optimal hyperparameter settings for the units decoding task are listed in Table 31.

Table 31: Hyperparameter settings for our model.

Hyperparameter	**Speech I**	**Speech II**	**Handwriting**
Batch size	16	16	16
Learning rate	1e-4	1e-4	1e-4
Hidden dimension	512	1024	1024
Distribution dimension	256	512	512
Encoder layers	4	4	2
Uncertainty attention layers	3	3	2
λ_{ale}	0.5	0.1	0.1
λ_{epi}	0.5	0.5	0.1
λ_{orth}	0.05	0.05	0.05

D.2 Recurrent and Convolutional Models

LSTM and TCN serve as classical sequence modeling baselines for neural decoding. LSTM captures temporal dependencies through recurrent hidden states and has been widely used in speech and motor decoding tasks Hochreiter and Schmidhuber [1997], Glaser et al. [2020], Yang et al. [2024]. TCN leverages dilated convolutions to model long-term temporal structure with reduced sequential dependencies Bai [2018], making it effective for high-dimensional neural activity.

Both models are tuned using the shared hyperparameter search space. For TCN, we also search over convolutional kernel sizes in the set {2, 3, 4, 5}. Their optimal configurations are shown in Table 32 and 33.

Table 32: Hyperparameter settings for LSTM.

Hyperparameter	**Speech I**	**Speech II**	**Handwriting**
Batch size	16	16	16
Learning rate	1e-3	1e-4	1e-3
Hidden dimension	1024	2048	512
Number of layers	4	3	2

Table 33: Hyperparameter settings for TCN.

Hyperparameter	**Speech I**	**Speech II**	**Handwriting**
Batch size	16	16	16
Learning rate	1e-4	1e-4	1e-3
Hidden dimension	1024	1024	512
Number of layers	4	6	3
Convolution kernel size	4	3	3

D.3 Attention-Based Sequence Models

Transformer and Conformer models apply self-attention mechanisms to capture global temporal context in neural activity. Transformer encoder-based architectures have demonstrated strong performance in neural speech and handwriting decoding by modeling long-term dependencies and contextual interactions Ye and Pandarinath [2021], Feghhi et al. [2025]. Conformer further integrates convolutional modules to enhance local temporal modeling Gulati et al. [2020], which can be beneficial for fine-grained neural dynamics.

For both models, we additionally search for the number of attention heads from {4, 8, 12}, while for Conformer, we additionally search for the convolution kernel size from {2, 3, 4, 5}. The hyperparameters are shown in Table 34 and 35.

Table 34: Hyperparameter settings for Transformer.

Hyperparameter	**Speech I**	**Speech II**	**Handwriting**
Batch size	16	16	16
Learning rate	1e-4	1e-4	1e-4
Hidden dimension	512	1024	256
Number of layers	4	6	2
Number of attention heads	8	8	4

Table 35: Hyperparameter settings for Conformer.

Hyperparameter	**Speech I**	**Speech II**	**Handwriting**
Batch size	16	16	16
Learning rate	1e-3	1e-3	1e-4
Hidden dimension	256	256	256
Number of layers	5	5	2
Number of attention heads	8	8	4
Convolution kernel size	4	3	3

D.4 Modern Time-Series Models

iTransformer Liu et al. [2023b], Mamba Gu and Dao [2024], TimesNet Wu et al. [2022], and TimeMixer Wang et al. [2024] represent recent advances in time-series modeling that emphasize efficiency, long-context modeling, or frequency-aware representations. While these models are not specifically designed for neural decoding, they achieve strong performance for evaluating the generalization of modern sequence models to neural data.

For iTransformer, we also search over the number of attention heads from {4, 8, 12}. For Mamba, we search over convolution kernel sizes ({2, 3, 4, 5}) and the number of states ({10, 20, 40, 80}). For TimesNet, we search over the convolution kernel size ({2, 3, 4, 5}). For TimeMixer, we also search over the convolution kernel size ({2, 3, 4, 5}), as well as the window length ({1, 2, 3, 4}) and number of layers ({1, 2, 3}) for downsampling. All settings are shown in Table 36, 37, 38, and 39.

Table 36: Hyperparameter settings for iTransformer.

Hyperparameter	**Speech I**	**Speech II**	**Handwriting**
Batch size	16	16	16
Learning rate	1e-3	1e-3	1e-3
Hidden dimension	256	512	64
Number of layers	2	2	2
Number of attention heads	8	8	8

Table 37: Hyperparameter settings for Mamba.

Hyperparameter	Speech I	Speech II	Handwriting
Batch size	16	16	16
Learning rate	1e-4	1e-3	1e-3
Hidden dimension	512	512	64
Number of layers	8	4	1
Convolution kernel size	3	3	3
Number of states	40	40	20

Table 38: Hyperparameter settings for TimesNet.

Hyperparameter	Speech I	Speech II	Handwriting
Batch size	16	16	16
Learning rate	1e-4	1e-4	1e-4
Hidden dimension	256	256	256
Number of layers	6	6	1
Convolution kernel size	3	3	3

Table 39: Hyperparameter settings for TimeMixer.

Hyperparameter	Speech I	Speech II	Handwriting
Batch size	16	16	16
Learning rate	1e-4	1e-4	1e-4
Hidden dimension	1024	1024	512
Number of layers	4	3	1
Convolution kernel size	4	3	3
Window length for downsampling	2	2	1
Number of layers for downsampling	1	1	1

E Bilinear Distance-Based Metric

E.1 Metric Definition

We consider representations $\mathbf{x}_i = (\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$, where $\boldsymbol{\sigma}_i$ contains Gaussian standard deviations and the corresponding diagonal covariance is $\mathrm{diag}(\boldsymbol{\sigma}_i^2)$. The metric results below assume that all representations share the same coordinate map and strictly positive distance weights. They characterize the square-rooted base distance; their applicability to attention scores depends on the query/key parameterization.

Definition E.1 (Bilinear Distance Metric)**.** For two representations $\mathbf{x}_i = (\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$ and $\mathbf{x}_j = (\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)$, we define the bilinear distance as:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{d_\mu^2(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) + \lambda \cdot d_\sigma^2(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j)} \tag{27}$$

where the component distances are weighted squared Euclidean distances:

$$d_\mu^2(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = \sum_{k=1}^{D_\mu} w_k^{(\mu)} \left(\mu_i^{(k)} - \mu_j^{(k)}\right)^2 \tag{28}$$

$$d_\sigma^2(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) = \sum_{k=1}^{D_\sigma} w_k^{(\sigma)} \left(\sigma_i^{(k)} - \sigma_j^{(k)}\right)^2 \tag{29}$$

with learnable parameters $w_k^{(\mu)} > 0$, $w_k^{(\sigma)} > 0$, and $\lambda > 0$.

Remark E.2 (Connection to Mahalanobis Distance). The bilinear distance in Definition E.1 can be interpreted as a special case of the Mahalanobis distance with a block-diagonal precision matrix. Specifically, by concatenating the probabilistic components $\mathbf{z}_i = [\boldsymbol{\mu}_i; \boldsymbol{\sigma}_i] \in \mathbb{R}^{D_\mu + D_\sigma}$, we have:

$$d^2(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{z}_i - \mathbf{z}_j)^\top \mathbf{M} (\mathbf{z}_i - \mathbf{z}_j) \tag{30}$$

where $\mathbf{M}$ is the block-diagonal precision matrix:

$$\mathbf{M} = \begin{bmatrix} \text{diag}(w_1^{(\mu)}, \ldots, w_{D_\mu}^{(\mu)}) & \mathbf{0} \\ \mathbf{0} & \lambda \cdot \text{diag}(w_1^{(\sigma)}, \ldots, w_{D_\sigma}^{(\sigma)}) \end{bmatrix} \tag{31}$$

This structure enforces three key properties:

1. **Dimensional decoupling**: Each dimension within $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ is weighted independently, allowing the model to learn dimension-specific importance.

2. **Subspace separation**: The mean and variance components do not interact (zero off-diagonal blocks), which aligns with the natural factorization of probabilistic representations.

3. **Computational efficiency**: The number of parameters scales as $O(D_\mu + D_\sigma)$ rather than $O((D_\mu + D_\sigma)^2)$ for a full Mahalanobis distance, making it tractable for high-dimensional representations.

Unlike general metric learning approaches that learn a full covariance structure [Weinberger and Saul, 2009, Davis et al., 2007], our block-diagonal constraint preserves the semantic separation between distributional parameters while providing theoretical guarantees of validity as a metric (see Theorem E.3).

E.2 Metric Properties

We now establish that the bilinear distance satisfies the fundamental axioms of a metric space, particularly the triangle inequality, which ensures geometric consistency in the representation space.

Theorem E.3 (Triangle Inequality)**.** *The bilinear distance $d(\mathbf{x}_i, \mathbf{x}_j)$ defined in* (27) *satisfies the triangle inequality:*

$$d(\mathbf{x}_i, \mathbf{x}_\ell) \leq d(\mathbf{x}_i, \mathbf{x}_j) + d(\mathbf{x}_j, \mathbf{x}_\ell), \quad \forall i, j, \ell \tag{32}$$

Proof. We prove this by showing that the bilinear distance is equivalent to the Euclidean distance in an appropriately transformed space, which inherits the triangle inequality from the standard ℓ^2 norm.

Step 1: Dimension-wise rescaling. Define the rescaled mean components:

$$\tilde{\boldsymbol{\mu}}_i = \begin{bmatrix} \sqrt{w_1^{(\mu)}}\mu_i^{(1)} \\ \sqrt{w_2^{(\mu)}}\mu_i^{(2)} \\ \vdots \\ \sqrt{w_{D_\mu}^{(\mu)}}\mu_i^{(D_\mu)} \end{bmatrix} \in \mathbb{R}^{D_\mu} \tag{33}$$

and similarly define the rescaled variance components:

$$\tilde{\boldsymbol{\sigma}}_i = \begin{bmatrix} \sqrt{w_1^{(\sigma)}}\sigma_i^{(1)} \\ \sqrt{w_2^{(\sigma)}}\sigma_i^{(2)} \\ \vdots \\ \sqrt{w_{D_\sigma}^{(\sigma)}}\sigma_i^{(D_\sigma)} \end{bmatrix} \in \mathbb{R}^{D_\sigma} \tag{34}$$

Then the component distances become standard Euclidean distances:

$$\begin{aligned} d_\mu^2(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) &= \sum_{k=1}^{D_\mu} w_k^{(\mu)} \left(\mu_i^{(k)} - \mu_j^{(k)}\right)^2 \\ &= \sum_{k=1}^{D_\mu} \left(\sqrt{w_k^{(\mu)}}\mu_i^{(k)} - \sqrt{w_k^{(\mu)}}\mu_j^{(k)}\right)^2 \\ &= \sum_{k=1}^{D_\mu} \left(\tilde{\mu}_i^{(k)} - \tilde{\mu}_j^{(k)}\right)^2 \\ &= \|\tilde{\boldsymbol{\mu}}_i - \tilde{\boldsymbol{\mu}}_j\|_2^2 \end{aligned} \tag{35}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. An analogous relationship holds for d_σ^2:

$$d_\sigma^2(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) = \|\tilde{\boldsymbol{\sigma}}_i - \tilde{\boldsymbol{\sigma}}_j\|_2^2 \tag{36}$$

Step 2: Augmented space construction. We construct an augmented representation by concatenating the rescaled components with appropriate weighting:

$$\mathbf{z}_i = \begin{bmatrix} \tilde{\boldsymbol{\mu}}_i \\ \sqrt{\lambda}\,\tilde{\boldsymbol{\sigma}}_i \end{bmatrix} \in \mathbb{R}^{D_\mu + D_\sigma} \tag{37}$$

Using (35) and (36), we can express the squared bilinear distance as:

$$\begin{aligned} d^2(\mathbf{x}_i, \mathbf{x}_j) &= d_\mu^2(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) + \lambda \cdot d_\sigma^2(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) \\ &= \|\tilde{\boldsymbol{\mu}}_i - \tilde{\boldsymbol{\mu}}_j\|_2^2 + \lambda\|\tilde{\boldsymbol{\sigma}}_i - \tilde{\boldsymbol{\sigma}}_j\|_2^2 \\ &= \|\tilde{\boldsymbol{\mu}}_i - \tilde{\boldsymbol{\mu}}_j\|_2^2 + \|\sqrt{\lambda}\,\tilde{\boldsymbol{\sigma}}_i - \sqrt{\lambda}\,\tilde{\boldsymbol{\sigma}}_j\|_2^2 \\ &= \left\| \begin{bmatrix} \tilde{\boldsymbol{\mu}}_i - \tilde{\boldsymbol{\mu}}_j \\ \sqrt{\lambda}(\tilde{\boldsymbol{\sigma}}_i - \tilde{\boldsymbol{\sigma}}_j) \end{bmatrix} \right\|_2^2 \\ &= \left\| \begin{bmatrix} \tilde{\boldsymbol{\mu}}_i \\ \sqrt{\lambda}\,\tilde{\boldsymbol{\sigma}}_i \end{bmatrix} - \begin{bmatrix} \tilde{\boldsymbol{\mu}}_j \\ \sqrt{\lambda}\,\tilde{\boldsymbol{\sigma}}_j \end{bmatrix} \right\|_2^2 \\ &= \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \end{aligned} \tag{38}$$

Therefore, the bilinear distance is simply the Euclidean distance in the augmented space:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{z}_i - \mathbf{z}_j\|_2 \tag{39}$$

Step 3: Triangle inequality via Minkowski inequality. Since $d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{z}_i - \mathbf{z}_j\|_2$ is a Euclidean norm, it satisfies the Minkowski inequality (triangle inequality for ℓ^p norms with $p \geq 1$). Specifically, for any three points $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_\ell$:

$$\begin{aligned} d(\mathbf{x}_i, \mathbf{x}_\ell) &= \|\mathbf{z}_i - \mathbf{z}_\ell\|_2 \\ &= \|(\mathbf{z}_i - \mathbf{z}_j) + (\mathbf{z}_j - \mathbf{z}_\ell)\|_2 \\ &\leq \|\mathbf{z}_i - \mathbf{z}_j\|_2 + \|\mathbf{z}_j - \mathbf{z}_\ell\|_2 \\ &= d(\mathbf{x}_i, \mathbf{x}_j) + d(\mathbf{x}_j, \mathbf{x}_\ell) \end{aligned} \tag{40}$$

where the inequality follows from the triangle inequality for Euclidean norms[Rudin, 2008].

□

Remark E.4 (Geometric Interpretation). The proof shows that the bilinear distance can be interpreted geometrically as the Euclidean distance in a rescaled and augmented feature space. The transformation in (37) performs two operations:

1. **Dimension-wise rescaling**: Each coordinate is multiplied by $\sqrt{w_k}$, effectively stretching or compressing the space along each axis according to learned importance weights.

2. **Subspace weighting**: The variance subspace is globally rescaled by $\sqrt{\lambda}$, controlling the relative contribution of uncertainty information to the overall distance.

This construction ensures that the representation space has a well-defined geometric structure consistent with Euclidean geometry, while preserving the semantic separation between mean and variance components. The learned weights $\{w_k^{(\mu)}, w_k^{(\sigma)}, \lambda\}$ can be interpreted as an anisotropic metric tensor that adapts to the data distribution.

Proposition E.5 (Complete Metric Axioms)**.** *The bilinear distance $d(\mathbf{x}_i, \mathbf{x}_j)$ satisfies all axioms of a metric:*

1. ***Non-negativity**: $d(\mathbf{x}_i, \mathbf{x}_j) \geq 0$ for all i, j.*
2. ***Identity of indiscernibles**: $d(\mathbf{x}_i, \mathbf{x}_j) = 0$ if and only if $\boldsymbol{\mu}_i = \boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_i = \boldsymbol{\sigma}_j$.*
3. ***Symmetry**: $d(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_j, \mathbf{x}_i)$ for all i, j.*
4. ***Triangle inequality**: $d(\mathbf{x}_i, \mathbf{x}_\ell) \leq d(\mathbf{x}_i, \mathbf{x}_j) + d(\mathbf{x}_j, \mathbf{x}_\ell)$ for all i, j, ℓ (Theorem E.3).*

Proof. **Property (i):** Since $w_k^{(\mu)}, w_k^{(\sigma)}, \lambda > 0$, both d_μ^2 and d_σ^2 are non-negative. Therefore, $d^2(\mathbf{x}_i, \mathbf{x}_j) = d_\mu^2 + \lambda \cdot d_\sigma^2 \geq 0$, which implies $d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{d^2(\mathbf{x}_i, \mathbf{x}_j)} \geq 0$.

Property (ii): The distance is zero if and only if both component distances are zero:

$$\begin{aligned} d(\mathbf{x}_i, \mathbf{x}_j) = 0 &\Leftrightarrow d_\mu^2(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) + \lambda \cdot d_\sigma^2(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) = 0 \\ &\Leftrightarrow d_\mu^2(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = 0 \text{ and } d_\sigma^2(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) = 0 \\ &\Leftrightarrow \boldsymbol{\mu}_i = \boldsymbol{\mu}_j \text{ and } \boldsymbol{\sigma}_i = \boldsymbol{\sigma}_j \end{aligned}$$

where the second equivalence uses the positivity of λ and the third uses the fact that weighted squared Euclidean distance is zero if and only if all coordinates match (since $w_k > 0$).

Property (iii): Symmetry follows immediately from the symmetry of squared Euclidean distance:

$$\begin{aligned} d(\mathbf{x}_i, \mathbf{x}_j) &= \sqrt{d_\mu^2(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) + \lambda \cdot d_\sigma^2(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j)} \\ &= \sqrt{d_\mu^2(\boldsymbol{\mu}_j, \boldsymbol{\mu}_i) + \lambda \cdot d_\sigma^2(\boldsymbol{\sigma}_j, \boldsymbol{\sigma}_i)} \\ &= d(\mathbf{x}_j, \mathbf{x}_i) \end{aligned}$$

Property (iv): This is established in Theorem E.3. □

F Variability-Aware Neural Decoding Framework for Language BCIs

F.1 Mathematical Framework

This section provides a conceptual information-theoretic motivation under an explicitly specified context distribution. The implemented stochastic projection head measures latent perturbation sensitivity. All entropy and mutual-information expressions in this section use natural logarithms.

Assumption F.1 (Problem Setup)**.**

1. Input neural signal $\mathbf{x} \in \mathbb{R}^d$, target word/token $y \in \mathcal{Y}$ with $|\mathcal{Y}| = C$.
2. Context variable $c \in \mathcal{C}$ represents linguistic/semantic context (syntactic structure, discourse state, pragmatic intent).
3. Model parameters θ are assumed fixed after training; epistemic uncertainty arises from context ambiguity $p(c|\mathbf{x}, \mathcal{D})$.
4. Decoder output conditioned on context:

$$p_\theta(y = k \mid \mathbf{x}, c) = \frac{e^{z_k(c)}}{\sum_{j=1}^{C} e^{z_j(c)}}, \quad z_k(c) = \mathbf{w}_k^\top \mathbf{h}(\mathbf{x}, c) + b_k$$

where $\mathbf{h}(\mathbf{x}, c)$ is a context-dependent feature representation.

5. Context prior follows asymmetric Dirichlet distribution: $\boldsymbol{\pi}_c \sim \text{Dir}(\boldsymbol{\alpha}_c)$ with $\alpha_{c,0} = \sum_i \alpha_{c,i}$.

Remark F.2 (Linguistic Interpretation of Context). In language decoding from neural signals, context ambiguity manifests as:

- **Syntactic ambiguity**: Same signal pattern may correspond to different grammatical structures (e.g., noun vs. verb usage)
- **Semantic ambiguity**: Polysemy or homonymy in intended words (e.g., "bank" as a financial institution vs. river edge)
- **Pragmatic uncertainty**: Speaker's communicative intent unclear from signal alone (e.g., literal vs. metaphorical meaning)
- **Discourse context**: Previous utterances influence current word probabilities

The context variable c encapsulates these sources of linguistic uncertainty.

F.2 Context-Aware Uncertainty Decomposition

Theorem F.3 (Total Uncertainty Decomposition)**.** *Under Assumption F.1, the predictive uncertainty admits the following additive decomposition:*

$$\underbrace{\mathbb{H}[y \mid \mathbf{x}, \mathcal{D}]}_{\textit{Total Uncertainty}} = \underbrace{\mathbb{E}_{c\sim p(\cdot|\mathbf{x})}\mathbb{H}[y \mid \mathbf{x}, c, \theta]}_{\textit{Aleatoric Uncertainty}} + \underbrace{\mathbb{I}[y; c \mid \mathbf{x}, \theta]}_{\textit{Epistemic Uncertainty (Context)}}$$

where $\mathbb{H}$ denotes Shannon entropy and $\mathbb{I}$ denotes mutual information.

Proof. By the chain rule for conditional entropy:

$$\begin{aligned}\mathbb{H}[y \mid \mathbf{x}, \mathcal{D}] &= \mathbb{H}[y, c \mid \mathbf{x}, \mathcal{D}] - \mathbb{H}[c \mid y, \mathbf{x}, \mathcal{D}] \\ &= \mathbb{H}[y \mid c, \mathbf{x}, \theta] + \mathbb{H}[c \mid \mathbf{x}] - \mathbb{H}[c \mid y, \mathbf{x}, \theta] \\ &= \mathbb{E}_c[\mathbb{H}[y \mid c, \mathbf{x}, \theta]] + \mathbb{I}[y; c \mid \mathbf{x}, \theta]\end{aligned}$$

where we used the definition of mutual information in the last step [Barber and Agakov, 2004]. □

Remark F.4 (Interpretation and Sampling). With fixed decoder parameters, context-related uncertainty can be studied through $\mathbb{I}[y; c \mid \mathbf{x}, \theta]$. The experiments operationalize context primarily through neighboring phonemes or letters. The theoretical context samples used below are independent draws from a specified context distribution. Stochastic projection outputs and attention-head diversity motivate sensitivity proxies, while an interpretation as samples from that context distribution requires additional assumptions.

Remark F.5 (Finite Sample Approximation). For independent context samples $c^{(1)}, \ldots, c^{(M)}$, define

$$\widetilde{\mathbb{I}} = \mathbb{H}(\bar{\mathbf{p}}) - \frac{1}{M}\sum_{m=1}^{M} \mathbb{H}(\mathbf{p}^{(m)}), \qquad \bar{\mathbf{p}} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{p}^{(m)}.$$

Entropy concavity gives $\mathbb{E}_{MC}[\widetilde{\mathbb{I}}] \leq \mathbb{I}[y; c \mid \mathbf{x}, \theta]$. This inequality holds in expectation over repeated context sampling; an individual realization has no deterministic lower-bound guarantee.

F.3 Softmax Lipschitz Analysis

Lemma F.6 (Jacobian Spectral Norm)**.** *For $\mathbf{p} = \text{softmax}(\mathbf{z})$, the Jacobian $J = \text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top$ satisfies $\|J\|_2 \leq 1/2$.*

Proof. The matrix J is symmetric, and the absolute sum of row i is $2p_i(1 - p_i) \leq 1/2$. Hence $\|J\|_2 \leq \|J\|_\infty \leq 1/2$. □

Corollary F.7 (Probability Vector Lipschitz Bound)**.** *For any two logit vectors,*

$$\|\,\text{softmax}(\mathbf{z}) - \text{softmax}(\mathbf{z}')\|_2 \leq \frac{1}{2}\|\mathbf{z} - \mathbf{z}'\|_2.$$

Proof. Integrate the Jacobian along the line segment joining the two logit vectors. □

F.4 Variance-Based Analysis

For $M > 1$ probability vectors, define the unbiased sample covariance

$$\widehat{\mathbf{\Sigma}}_p = \frac{1}{M-1} \sum_{m=1}^{M} (\mathbf{p}^{(m)} - \bar{\mathbf{p}})(\mathbf{p}^{(m)} - \bar{\mathbf{p}})^\top.$$

Lemma F.8 (Covariance Propagation Through Softmax)**.** *For matched samples of logits and their softmax probabilities,*

$$\mathrm{Tr}(\widehat{\mathbf{\Sigma}}_p) \leq \frac{1}{4} \mathrm{Tr}(\widehat{\mathbf{\Sigma}}_z).$$

Proof. For any sample vectors $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(M)}$,

$$\mathrm{Tr}(\widehat{\mathbf{\Sigma}}_u) = \frac{1}{2M(M-1)} \sum_{m,n=1}^{M} \|\mathbf{u}^{(m)} - \mathbf{u}^{(n)}\|_2^2.$$

Apply Corollary F.7 to each pair. □

Theorem F.9 (Context-Dependent MI Lower Bound)**.** *For a fixed input and decoder, let* $\mathbf{q} = \mathbb{E}_c[\mathbf{p}(c)]$*. Then*

$$\mathbb{I}[y; c \mid \mathbf{x}, \theta] \geq \frac{1}{2} \mathbb{E}_c \|\mathbf{p}(c) - \mathbf{q}\|_1^2 \geq \frac{1}{2} \mathrm{Tr}\left(\mathrm{Cov}_c[\mathbf{p}(c)]\right).$$

Proof. Use $\mathbb{I}[y; c \mid \mathbf{x}, \theta] = \mathbb{E}_c D_{\mathrm{KL}}(\mathbf{p}(c) \| \mathbf{q})$, Pinsker's inequality with natural logarithms, and $\|\mathbf{u}\|_1 \geq \|\mathbf{u}\|_2$. □

Corollary F.10 (Sample Covariance Interpretation)**.** *For independent context samples,*

$$\mathbb{I}[y; c \mid \mathbf{x}, \theta] \geq \frac{1}{2} \mathbb{E}_{MC}\left[\mathrm{Tr}(\widehat{\mathbf{\Sigma}}_p)\right].$$

Proof. The sample covariance is unbiased for the population covariance. □

The guarantee concerns probability covariance in expectation. The logit covariance relation in Lemma F.8 provides an upper bound on probability variation and does not imply a logit-based MI lower bound.

F.5 Entropy Under Context-Dependent Dirichlet Priors

Theorem F.11 (Expected Dirichlet Entropy)**.** *Let* $\boldsymbol{\pi} \mid c \sim \mathrm{Dir}(\boldsymbol{\alpha}_c)$ *with positive parameters,* $\alpha_{c,0} = \sum_k \alpha_{c,k}$*, and* $q_{c,k} = \alpha_{c,k}/\alpha_{c,0}$*. Then*

$$\mathbb{E}_{\boldsymbol{\pi}|c} \mathbb{H}(\boldsymbol{\pi}) = \psi(\alpha_{c,0} + 1) - \sum_k q_{c,k} \psi(\alpha_{c,k} + 1),$$

and

$$0 \leq G_c := \mathbb{H}(\mathbf{q}_c) - \mathbb{E}_{\boldsymbol{\pi}|c} \mathbb{H}(\boldsymbol{\pi}) \leq \mathbb{H}(\mathbf{q}_c).$$

Proof. Differentiating the Dirichlet moment $\mathbb{E}[\pi_k^r]$ at $r = 1$ gives $\mathbb{E}[\pi_k \log \pi_k] = q_{c,k}[\psi(\alpha_{c,k} + 1) - \psi(\alpha_{c,0} + 1)]$. Summation gives the entropy identity. The gap bounds follow from entropy concavity and nonnegative entropy. □

Corollary F.12 (Context-Averaged Entropy Gap)**.** *Averaging over contexts gives*

$$0 \leq \mathbb{E}_c G_c \leq \mathbb{E}_c \mathbb{H}(\mathbf{q}_c).$$

F.6 Finite-Sample Convergence Analysis

Theorem F.13 (MC Approximation Error Rates)**.** *Assume independent context samples, a fixed finite vocabulary, and $p_k(c) \geq \epsilon > 0$ for every class and sampled context. Then the entropy-difference estimator above satisfies*

$$\left| \mathbb{E}_{MC}[\widetilde{\mathbb{I}}] - \mathbb{I}[y; c \mid \mathbf{x}, \theta] \right| = O(M^{-1}), \qquad \mathrm{Var}_{MC}(\widetilde{\mathbb{I}}) = O(M^{-1}).$$

Proof. Entropy has bounded derivatives on the assumed probability domain. A second-order Taylor expansion around $\mathbb{E}_c \mathbf{p}(c)$ gives an $O(M^{-1})$ bias for $\mathbb{H}(\bar{\mathbf{p}})$. Its variance is $O(M^{-1})$, while the sample mean of conditional entropies is unbiased with variance $O(M^{-1})$. Combining the two terms yields the stated rates. □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, our abstract and introduction include the claims that reflect our contribution and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, in the last section, we discuss the limitations of our work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Yes, in the appendix, we provide the key assumptions and proofs of our approach.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, in the experimental section and appendix, we provide the hyperparameters and experimental procedure for reproducing our method.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, we list the link to the open data source in the appendix.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, in the appendix, we provide the hyperparameters of our method.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, we report paired t-test results in our experiment section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, in the experiment section and appendix, we provide the information on the computer resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, our work conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our work does not address societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, we have mentioned it in the appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Our work does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We perform secondary analysis of de-identified publicly released human neural datasets; no new participant recruitment, intervention, or data collection was conducted. Original studies reported their ethical approval/consent procedures.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We perform secondary analysis of de-identified publicly released human neural datasets; no new participant recruitment, intervention, or data collection was conducted. Original studies reported their ethical approval/consent procedures.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The benchmark system evaluations describe language-model post-processing, including a fine-tuned Llama 3.1 8B model.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.