GACE: GUIDANCE, ASSESSMENT, AND CORRECTION FOR EEG FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

EEG foundation models (EFMs) learn rich internal knowledge from large-scale EEG pretraining, encoding transferable spatiotemporal representations and task-relevant priors. We observe two complementary patterns in fine-tuned EFMs: correct and incorrect predictions exhibit different spatial reliance, while an intermediate readout can give the true class a larger probability margin over its strongest competitor than the final erroneous readout. These observations motivate Decision Evidence for assessing spatial reliance and Decision Trajectory as a class-wise logit-alignment feature for supervised correction. This work aims to turn the internal decision structures of EFMs into actionable signals for self-guidance, self-assessment, and self-correction. We propose GACE, a plug-and-play framework that first uses true-class support and response stability to guide fine-tuning toward clearer Decision Evidence and Decision Trajectory, performs reliability modeling using class probabilities and Decision Evidence, and further incorporates Decision Trajectory to correct high-risk predictions, improving decision reliability without modifying the EFM backbone.

1 INTRODUCTION

Electroencephalography (EEG) is a non-invasive neural modality widely used in motor imagery (Lee et al., 2019), sleep staging (Perslev et al., 2021; Zhou et al., 2024), emotion recognition (Liu et al., 2024b), and other neural decoding tasks (Song et al., 2022). Recently, EEG foundation models (EFMs) have emerged as a prominent research direction (Jiang et al., 2024; Wang et al., 2025a). Through large-scale self-supervised pretraining on EEG signals across subjects, sessions, and tasks, EFMs acquire rich internal knowledge of spatiotemporal structure and discriminative patterns, providing transferable EEG priors for downstream adaptation (Liu et al., 2026b; Zhou et al., 2025b).

Rich pretrained knowledge, however, does not always ensure accurate downstream predictions. Prior work on large language models (LLMs) has explored feeding generated answers back into the model and using its own feedback to evaluate and revise them (Madaan et al., 2023; Lee et al., 2025; Kumar et al., 2025; Wang et al., 2024c), demonstrating the potential of a model's own responses to support

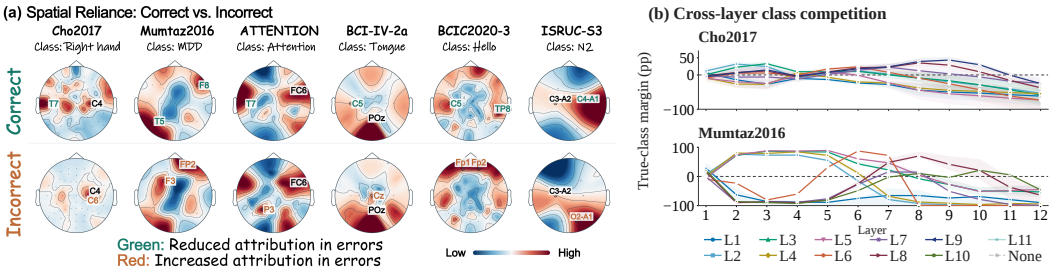

Figure 1: **CSBrain decision patterns (seed 0). (a)** Spatial reliance differs between correct and incorrect predictions. **(b)** True-class margins over the strongest competing class, grouped by the earlier layer of greatest margin advantage over the final readout (means and sample interquartile ranges).

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

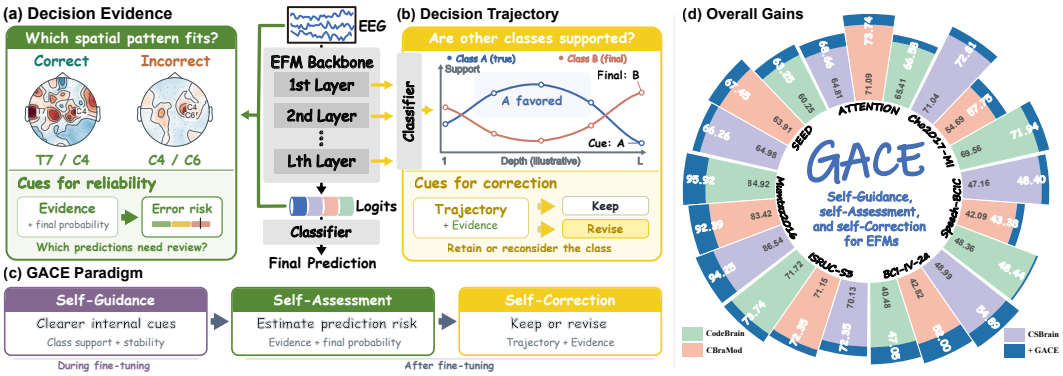

Figure 2: Overview of GACE: **(a)** spatial evidence; **(b)** illustrative class probabilities motivating cross-layer alignment features; **(c)** Self-Guidance, Self-Assessment, and Self-Correction; **(d)** balanced accuracy on six main tasks and supplementary Cho2017-MI (gains in blue).

prediction assessment and improvement. Although EFMs express their decisions as class probabilities rather than text, this perspective motivates us to *listen* to their internal responses for cues that can inform reliability assessment and selective correction. Analyzing the inference behavior of fine-tuned EFMs reveals two patterns:

I. Correct and incorrect EFM predictions often exhibit different patterns of spatial reliance. For right-hand motor imagery in Figure 1(a), the two highest-attribution channels are T7 and C4 for correct predictions, versus C4 and C6 for incorrect predictions. Incorrect predictions also exhibit a structured spatial preference. The spatial organization of motor-imagery-related electrophysiological activity provides a task-specific context (Saha & Baumert, 2020; Ding et al., 2024), while these attribution differences suggest an association between spatial reliance and prediction outcomes. We refer to this spatial reliance, which can inform reliability assessment, as **Decision Evidence**, as illustrated in Figure 2(a).

II. Intermediate readouts can favor the true class more than the final erroneous output. We read intermediate representations in a shared class space. Figure 1(b) plots the true-class probability minus the strongest competing-class probability, grouping final errors by the earlier layer of greatest margin advantage over the final readout. A positive sample margin indicates a correct intermediate prediction; an increase below zero indicates improved relative support without a correct prediction. These retrospective patterns motivate cross-layer analysis, while the operational **Decision Trajectory** summarizes class-wise alignment between intermediate and final logits as a feature for supervised correction (Figure 2(b); Appendix E.2).

These two patterns provide complementary cues: Decision Evidence helps assess prediction reliability, while Decision Trajectory provides complementary features for learning correction actions. This leads to our central question: **Can EFMs use these internal cues to detect and correct prediction errors?**

To this end, we propose **GACE**, a simple, effective plug-and-play framework for different EFM backbones without architectural changes. It improves prediction reliability through three connected steps (Figure 2(c)):

(I) Self-Guidance: How can we make internal cues clearer? We adjust sample weights online during fine-tuning based on true-class support and prediction stability to encourage clearer Decision Evidence and Decision Trajectory.

(II) Self-Assessment: How can we identify high-risk predictions? We train a risk head on calibration data using Decision Evidence and final-output information to estimate prediction error risk.

(III) Self-Correction: How can we revise high-risk predictions? We combine Decision Trajectory and Decision Evidence in a correction head that selects between retaining the original prediction and switching to another class, enabling selective correction of high-risk decisions.

We present results for GACE on six downstream EEG datasets with CodeBrain (Ma et al., 2026), CBraMod (Wang et al., 2025a), and CSBrain (Zhou et al., 2025c). Across the 18 main-benchmark combinations, GACE improves balanced accuracy and weighted F1 over fine-tuned baselines by 4.05 and 3.86 percentage points on average, respectively, with balanced-accuracy gains reaching 11.00 points. Ablation studies and diagnostic analyses examine how individual modules and internal cues contribute to fine-tuning, risk identification, and selective correction. Together, these results show that EFM internal responses provide actionable cues for risk assessment and selective correction, improving downstream prediction reliability.

2 PRELIMINARY

2.1 EEG FOUNDATION MODEL

Given an EEG sample $\mathbf{X} \in \mathbb{R}^{C\times T}$, where C denotes the number of channels and T denotes sampling points, an EFM consists of an encoder $\mathcal{F}(\cdot;\theta)$ and a classification head $\mathcal{H}(\cdot;\phi)$. For an L-layer encoder, we denote the layer-wise **Representation** and final **Logits** as:

$$\mathbf{Z}^{(l)}(\mathbf{X}) = \mathcal{F}^{(l)}(\mathbf{X};\theta), \quad \boldsymbol{\ell}(\mathbf{X}) = \mathcal{H}(\mathbf{Z}^{(L)}(\mathbf{X});\phi), \quad l = 1,\dots,L, \tag{1}$$

where $\mathbf{Z}^{(l)}(\mathbf{X})$ denotes the representation at the l-th encoder layer, $\mathbf{Z}^{(L)}(\mathbf{X})$ is the final representation used for classification, and $\boldsymbol{\ell}(\mathbf{X}) \in \mathbb{R}^K$ contains the pre-softmax class responses over K candidate classes. Applying softmax to the logits yields the predicted class probabilities:

$$\mathbf{R}(\mathbf{X}) = \mathrm{Softmax}\big(\boldsymbol{\ell}(\mathbf{X})\big), \tag{2}$$

where $\mathbf{R}(\mathbf{X}) \in [0,1]^K$ is the class probability vector, and the final predicted label is $\hat{y} = \arg\max_k [\mathbf{R}(\mathbf{X})]_k$.

2.2 DECISION STRUCTURE

The class probability vector $\mathbf{R}(\mathbf{X})$ summarizes the model's final predictive distribution but does not fully capture its internal decision information. We therefore formalize the two complementary internal cues introduced above: Decision Evidence describes spatial reliance, and Decision Trajectory summarizes class-wise alignment across layer-wise and final logits.

- **Decision Evidence $\mathbf{E}(\mathbf{X})$:** denotes the normalized channel-level attribution associated with the model's decision. Specifically, an attribution-based evidence function estimates a non-negative score $a_c(\mathbf{X}) \geq 0$ for each EEG channel using the maximum-logit response $g(\mathbf{U}) = \max_k \ell_k(\mathbf{U})$ along the Integrated Gradients path from zero to $\mathbf{X}$; the maximizing class may change along this path (details in Appendix E.1). Decision Evidence is defined as the normalized vector of these channel-wise scores:

$$\mathbf{E}(\mathbf{X}) = [\bar{a}_1(\mathbf{X}),\dots,\bar{a}_C(\mathbf{X})]^\top, \quad \bar{a}_c(\mathbf{X}) = \frac{a_c(\mathbf{X})}{\sum_{j=1}^{C} a_j(\mathbf{X}) + \epsilon}, \tag{3}$$

  where $\mathbf{E}(\mathbf{X}) \in \mathbb{R}^C$ and $\bar{a}_c(\mathbf{X})$ represents the relative magnitude of the c-th EEG channel's attribution to the maximum-logit response along the path.

- **Decision Trajectory $\mathbf{T}(\mathbf{X})$:** summarizes class-wise alignment between intermediate and final logits. Specifically, we feed each layer-wise representation $\mathbf{Z}^{(l)}(\mathbf{X})$ into the classification head to obtain layer-wise logits $\boldsymbol{\ell}^{(l)}(\mathbf{X}) = \mathcal{H}(\mathbf{Z}^{(l)}(\mathbf{X});\phi)$, from which we compute the normalized trajectory score $p^{(l)}_{\log}(c \mid \mathbf{X})$ and aggregation weight $w_l(\mathbf{X})$ (details in Appendix E.2). We aggregate these layer-wise trajectory scores into a class-wise alignment vector:

$$\mathbf{T}(\mathbf{X}) = [p_{\log}(1 \mid \mathbf{X}),\dots,p_{\log}(K \mid \mathbf{X})]^\top, \quad p_{\log}(c \mid \mathbf{X}) = \sum_{l=1}^{L} w_l(\mathbf{X}) p^{(l)}_{\log}(c \mid \mathbf{X}), \tag{4}$$

  where $\mathbf{T}(\mathbf{X}) \in \mathbb{R}^K$ and $p_{\log}(c \mid \mathbf{X})$ denotes the aggregated alignment score assigned to class c by the intermediate-versus-final logit differences.

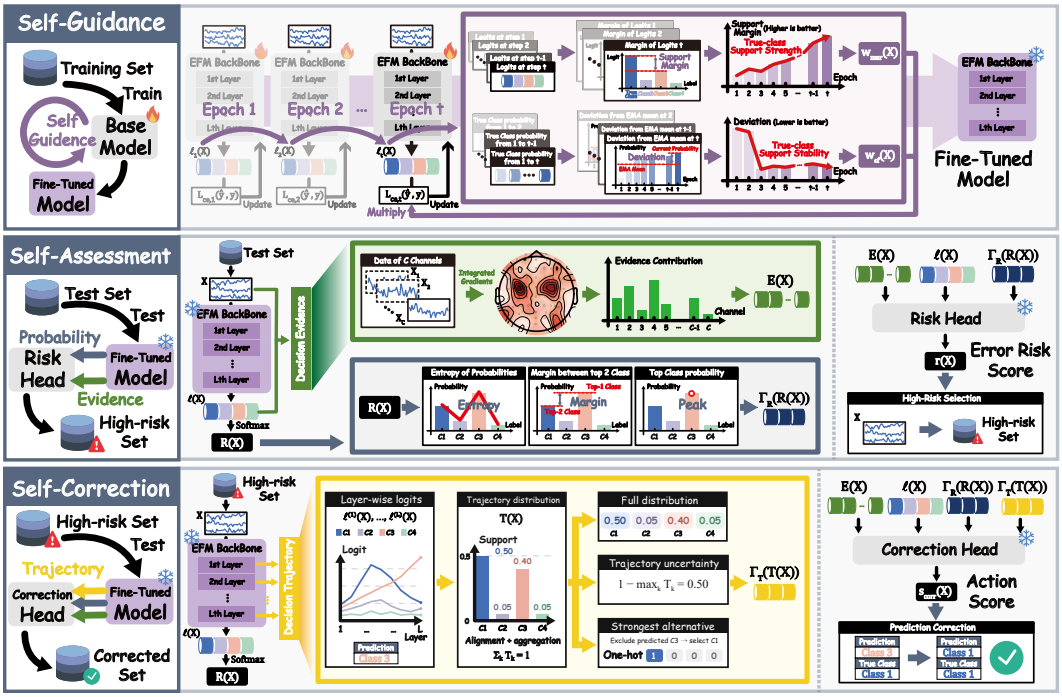

Figure 3: Overview of GACE: **Self-Guidance** uses class support and response stability to guide fine-tuning; **Self-Assessment** estimates prediction risk; **Self-Correction** selects keep-or-change actions for high-risk predictions.

3 METHODOLOGY

As summarized in Fig. 3, GACE contains three stages: self-guided fine-tuning to encourage clearer Decision Evidence and Decision Trajectory, risk modeling for self-assessment, and conservative self-correction. Sections 3.1–3.3 detail these stages, with complete computational procedures provided in Appendix D.

3.1 SELF-GUIDED FINE-TUNING

Intuition. Self-assessment and correction benefit from discriminative Decision Evidence and informative cross-layer alignment patterns in Decision Trajectory. Since fine-tuning shapes both cues, we examine training responses for guidance. Stronger online true-class margins are associated with larger margin-based training weights, while removing high-weight training samples reduces evidence concentration on task-relevant channels (Section 4.4 and Appendix L). These observations motivate Self-Guidance: using true-class support and confidence stability as online feedback, we emphasize samples with stronger support and more stable responses to encourage clearer internal cues for subsequent assessment and correction.

Principle 3.1. Response-guided sample weighting. Let $\boldsymbol{\ell}_t(\mathbf{X})$ denote the logits when training sample $\mathbf{X}$ is observed at stage t, and let Ψ extract a response statistic. We aggregate these responses as:

$$\mathbf{D}(\mathbf{X};\Psi) = \mathrm{EMA}_{t\le\tau}\left[\Psi\left(\boldsymbol{\ell}_t(\mathbf{X})\right)\right], \tag{5}$$

Here, $\mathrm{EMA}_{t\le\tau}$ is the exponential moving average through stage τ, used for sample weighting (Appendix E.3).

Let $U(\mathbf{D}(\mathbf{X};\Psi))$ denote a scalar guidance utility, oriented so that larger values indicate stronger true-class support or more stable confidence responses. For each statistic, Self-Guidance should preserve this utility ordering with the other weighting factor held fixed:

$$U(\mathbf{D}(\mathbf{X}_i;\Psi)) > U(\mathbf{D}(\mathbf{X}_j;\Psi)) \Rightarrow w_i \ge w_j, \tag{6}$$

Here, w_i and w_j are the online weights of $\mathbf{X}_i$ and $\mathbf{X}_j$.

Operationalization. Following Principle 3.1, **margin support** and **confidence fluctuation** yield online weights $w_{\mathrm{mar}}(\mathbf{X})$ and $w_{\mathrm{cf}}(\mathbf{X})$ for class support and response stability, respectively.

Margin Support and Confidence Fluctuation. For a training sample $(\mathbf{X}, y)$, the model produces logits $\boldsymbol{\ell}_t(\mathbf{X})$ at training stage t. We instantiate the response extractor Ψ with two operators:

$$\Psi\big(\boldsymbol{\ell}_t(\mathbf{X})\big) = \begin{cases} \ell_{t,y}(\mathbf{X}) - \max_{k \neq y} \ell_{t,k}(\mathbf{X}), & \Psi = \Psi_{\mathrm{mar}}, \\ \big(p_{t,y}(\mathbf{X}) - \mu_{t,y}(\mathbf{X})\big)^2, & \Psi = \Psi_{\mathrm{cf}} \end{cases}, \tag{7}$$

where Ψ_{mar} measures true-class logit advantage over the strongest competitor, and Ψ_{cf} measures historical true-class confidence fluctuation, with $p_{t,y}(\mathbf{X}) = \mathrm{Softmax}(\boldsymbol{\ell}_t(\mathbf{X}))_y$ and $\mu_{t,y}(\mathbf{X})$ denoting its historical mean. Larger Ψ_{mar} indicates clearer class support, while larger Ψ_{cf} indicates less stable training signals.

Response-Guided Online Weighting. Given the accumulated response statistics, we construct a weight increasing with margin support and a weight decreasing with confidence fluctuation:

$$\begin{aligned} w_{\mathrm{mar}}(\mathbf{X}) &= (1 - \lambda_{\mathrm{mar}}) + \lambda_{\mathrm{mar}} \sigma\big(\,\mathrm{RNorm}(\mathbf{D}(\mathbf{X}; \Psi_{\mathrm{mar}}))\big), \\ w_{\mathrm{cf}}(\mathbf{X}) &= \exp\left(-\lambda_{\mathrm{cf}} \max\left(\mathrm{RNorm}(\mathbf{D}(\mathbf{X}; \Psi_{\mathrm{cf}})), 0\right)\right), \end{aligned} \tag{8}$$

where $\mathrm{RNorm}(\cdot)$ denotes median/MAD (median absolute deviation) normalization of the response statistics, $\sigma(\cdot)$ is the sigmoid function, and λ_{mar} and λ_{cf} control weighting strengths. Finally, the two weights are injected multiplicatively into the standard cross-entropy objective:

$$\mathcal{L}_{\mathrm{sg}} = \frac{1}{|B|} \sum_{(\mathbf{X}, y) \in B} w_{\mathrm{mar}}(\mathbf{X}) w_{\mathrm{cf}}(\mathbf{X}) \cdot \ell_{\mathrm{ce}}(\mathbf{X}, y), \tag{9}$$

where B denotes a mini-batch. This objective emphasizes stronger true-class support and more stable confidence, encouraging clearer Decision Evidence and Decision Trajectory for subsequent assessment and correction.

3.2 Risk Modeling for Self-Assessment

Intuition. Decision Evidence captures the spatial attribution pattern associated with an EFM decision, while final logits and output-probability statistics characterize its posterior confidence and uncertainty. We therefore formulate self-assessment as risk modeling: a post-hoc risk input built from these signals estimates the probability that the current prediction is incorrect.

Lemma 3.2. Conditional risk is optimal for fixed-budget error selection. Given a test sample $\mathbf{X}$, let $\tau^{\mathrm{risk}}(\mathbf{X}) = \mathbb{I}[\hat{y} \neq y]$ denote whether the current prediction is incorrect, and let $\mathbf{v}^{\mathrm{risk}}(\mathbf{X})$ denote its post-hoc risk input. Define the conditional post-hoc risk as $\eta(\mathbf{v}^{\mathrm{risk}}(\mathbf{X})) = P(\tau^{\mathrm{risk}}(\mathbf{X}) = 1 \mid \mathbf{v}^{\mathrm{risk}}(\mathbf{X}))$. For a test set $\mathcal{D}_{\mathrm{test}} = \{\mathbf{X}_i\}_{i=1}^N$, under a fixed rejection budget $\rho \in (0, 1)$, if S_ρ^* denotes the optimal subset of $m = \lceil \rho N \rceil$ high-risk predictions that maximizes the expected number of captured errors, then

$$S_\rho^* = \mathrm{Top}_m \left\{\eta\big(\mathbf{v}^{\mathrm{risk}}(\mathbf{X}_i)\big)\right\}_{i=1}^N. \tag{10}$$

Therefore, $\eta(\mathbf{v}^{\mathrm{risk}}(\mathbf{X}))$ is the optimal ranking statistic for fixed-budget self-assessment and error selection. The full derivation is provided in Appendix F.1.

Operationalization. Following Lemma 3.2, a lightweight risk estimator learns conditional error risk from labeled calibration samples and ranks predictions for rejection or correction.

Post-hoc Risk Pair Construction. Each calibration sample becomes a risk modeling pair supervised by whether the frozen EFM predicts correctly. For a sample $(\mathbf{X}, y)$, the frozen EFM produces the softmax class-probability vector $\mathbf{R}(\mathbf{X})$, logits $\boldsymbol{\ell}(\mathbf{X})$, and predicted label $\hat{y}$. We construct the pair:

$$\big(\mathbf{v}^{\mathrm{risk}}(\mathbf{X}), \tau^{\mathrm{risk}}(\mathbf{X})\big) = \left([\mathbf{E}(\mathbf{X}); \boldsymbol{\ell}(\mathbf{X}); \Gamma_{\mathrm{R}}(\mathbf{R}(\mathbf{X}))], \mathbb{I}[\hat{y} \neq y]\right), \tag{11}$$

where $\mathbf{v}^{\mathrm{risk}}$ concatenates Decision Evidence $\mathbf{E}(\mathbf{X})$, final logits $\boldsymbol{\ell}(\mathbf{X})$, and output-probability statistics $\Gamma_{\mathrm{R}}(\mathbf{R}(\mathbf{X}))$. The operator Γ_{R} summarizes posterior confidence and uncertainty from the output probabilities, such as peak probability, entropy, and probability margin (Appendix F.1). The output τ^{risk} is a calibration-derived error indicator that supervises prediction-risk estimation.

Risk Optimization. The risk head combines final-prediction uncertainty with a learned residual and, when configured, auxiliary class support to produce a scalar score $s(\mathbf{X})$, with $r(\mathbf{X}) = \sigma(s(\mathbf{X}))$. On labeled calibration samples, it fits the original prediction's error indicator with binary cross-entropy and, when enabled, trains the auxiliary class head with class cross-entropy. Internal K-fold validation on the calibration set selects the risk-head checkpoint epoch and supplies out-of-fold scores. Self-Correction trains on the 12% highest-risk calibration samples, while the risk head fitted on the full calibration set ranks test predictions. At inference, the risk score ranks predictions under the fixed selection budget; the score and training objective are specified in Appendix F.1.

3.3 Conservative Self-Correction

Intuition. Decision Evidence and Decision Trajectory provide post-hoc signals that are not captured by the final prediction; the trajectory encodes class-wise alignment between intermediate and final logits. We therefore formulate self-correction as conservative post-hoc action selection: for high-risk predictions, the model selects a candidate among class-label actions and applies it only when the change gates pass; otherwise, it preserves the original prediction.

Principle 3.3. Conservative conditional self-correction. For a high-risk sample, self-correction should not re-classify the raw EEG signal, but conditionally choose between preserving the original prediction and making a limited correction from the residual decision structure. Given the correction input $\mathbf{v}^{\mathrm{corr}}(\mathbf{X})$, the ideal correction action is

$$a^*(\mathbf{X}) = \arg\max_{a \in \mathcal{A}} P\big(y_{\mathrm{corr}}(a) = y \mid \mathbf{v}^{\mathrm{corr}}(\mathbf{X})\big), \tag{12}$$

where $\mathcal{A} = \{\mathrm{keep}, 1, \ldots, K\}$, keep preserves the original prediction, $a = k$ changes it to class k, and $y_{\mathrm{corr}}(a)$ denotes the final prediction after applying action a.

Operationalization. Following Principle 3.3, we instantiate self-correction as a conservative $K+1$-way keep-or-change action policy. The correction head builds a correction input from post-hoc decision signals and the prediction already expressed by the EFM, including the original-prediction identity. Its supervision τ^{corr} directly follows action selection: if the original prediction is correct, the target action is keep; otherwise, the target action is the ground-truth class y.

Post-hoc Correction Pair Construction. The correction pair extends the risk pair toward action selection by retaining the post-hoc risk signals from Section 3.2 and additionally incorporating Decision Trajectory. Even when the output probabilities favor an incorrect class, the intermediate-versus-final logit differences provide additional features for a supervised correction head. For a high-risk sample $(\mathbf{X}, y)$, we construct:

$$\big(\mathbf{v}^{\mathrm{corr}}(\mathbf{X}), \tau^{\mathrm{corr}}(\mathbf{X})\big) = \left([\mathbf{E}(\mathbf{X}); \boldsymbol{\ell}(\mathbf{X}); \Gamma_{\mathrm{R}}(\mathbf{R}(\mathbf{X})); \Gamma_{\mathrm{T}}(\mathbf{T}(\mathbf{X}))], \begin{cases} \mathrm{keep}, & \hat{y} = y, \\ y, & \hat{y} \neq y, \end{cases}\right), \tag{13}$$

where Γ_{T} summarizes class-wise trajectory alignment cues (Appendix F.2). The output τ^{corr} is a keep-or-change action label: correct predictions map to keep, while erroneous predictions map to the ground-truth class y.

Conservative Correction Optimization. The correction head learns keep-or-change scores; explicit gates restrict changes to eligible alternative classes. Given a correction pair $(\mathbf{v}^{\mathrm{corr}}(\mathbf{X}), \tau^{\mathrm{corr}}(\mathbf{X}))$, a lightweight correction head $h(\cdot)$ outputs action scores $\mathbf{s}_{\mathrm{corr}}(\mathbf{X}) = h(\mathbf{v}^{\mathrm{corr}}(\mathbf{X})) \in \mathbb{R}^{K+1}$ over $\mathcal{A} = \{\mathrm{keep}, 1, \ldots, K\}$. We optimize it with a target-action ranking constraint:

$$s_{\tau^{\mathrm{corr}}(\mathbf{X})}(\mathbf{X}) > s_a(\mathbf{X}), \qquad \forall a \in \mathcal{A},\ a \neq \tau^{\mathrm{corr}}(\mathbf{X}), \tag{14}$$

where $s_a(\mathbf{X})$ denotes the score assigned to action a. This ranks keep above all changes for correct predictions, and ranks the ground-truth correction above keep and other changes for erroneous predictions. At inference time, among high-risk predictions, modification is applied only when the most probable class action differs from the original class and passes both the class-action-mass and candidate-versus-original-class probability-margin gates (Eq. (65)); otherwise, the original prediction is preserved. The keep probability determines the class-action mass through $1 - \pi_{\mathrm{keep}}$; it is not compared directly with the candidate class probability.

4 EXPERIMENTS

Table 1: Main comparison across EEG foundation backbones and downstream datasets. Each dataset is reported with B-ACC, Kappa, and F1.

Backbone	Setting	ATTENTION			SEED		
		B-ACC	Kappa	F1	B-ACC	Kappa	F1
CodeBrain	Baseline	0.6541±0.0195	0.3082±0.0390	0.6498±0.0254	0.6025±0.0365	0.2049±0.0730	0.5866±0.0510
	GACE	**0.6658±0.0113**	**0.3317±0.0225**	**0.6633±0.0099**	**0.6325±0.0056**	**0.2650±0.0111**	**0.6293±0.0094**
	Δ	+0.0117	+0.0235	+0.0135	+0.0300	+0.0601	+0.0427
CBraMod	Baseline	0.7109±0.0046	0.4218±0.0091	0.7089±0.0044	0.6391±0.0058	0.2782±0.0117	0.6316±0.0087
	GACE	**0.7374±0.0143**	**0.4749±0.0285**	**0.7370±0.0146**	**0.6745±0.0166**	**0.3490±0.0332**	**0.6733±0.0177**
	Δ	+0.0265	+0.0531	+0.0281	+0.0354	+0.0708	+0.0417
CSBrain	Baseline	0.6481±0.0063	0.2963±0.0125	0.6452±0.0086	0.6498±0.0125	0.2996±0.0250	0.6484±0.0137
	GACE	**0.6866±0.0069**	**0.3733±0.0139**	**0.6862±0.0076**	**0.6626±0.0087**	**0.3251±0.0173**	**0.6618±0.0097**
	Δ	+0.0385	+0.0770	+0.0410	+0.0128	+0.0255	+0.0134
Backbone	**Setting**	**Mumtaz2016**			**BCI_IV_2a**		
		B-ACC	Kappa	F1	B-ACC	Kappa	F1
CodeBrain	Baseline	0.8492±0.0137	0.7479±0.0191	0.8963±0.0075	0.4048±0.0152	0.2064±0.0203	0.3467±0.0409
	GACE	**0.9592±0.0140**	**0.8866±0.0202**	**0.9512±0.0085**	**0.4705±0.0211**	**0.2940±0.0282**	**0.4403±0.0326**
	Δ	+0.1100	+0.1387	+0.0549	+0.0657	+0.0876	+0.0936
CBraMod	Baseline	0.8342±0.0077	0.7065±0.0256	0.8787±0.0111	0.4282±0.0105	0.2377±0.0140	0.3892±0.0154
	GACE	**0.9239±0.0381**	**0.7821±0.1112**	**0.9014±0.0525**	**0.5200±0.0257**	**0.3600±0.0343**	**0.4996±0.0332**
	Δ	+0.0897	+0.0756	+0.0227	+0.0918	+0.1223	+0.1104
CSBrain	Baseline	0.8654±0.0038	0.7596±0.0253	0.8999±0.0117	0.4899±0.0420	0.3198±0.0560	0.4675±0.0524
	GACE	**0.9425±0.0197**	**0.8418±0.0716**	**0.9305±0.0325**	**0.5489±0.0265**	**0.3985±0.0353**	**0.5375±0.0285**
	Δ	+0.0771	+0.0822	+0.0306	+0.0590	+0.0787	+0.0700
Backbone	**Setting**	**ISRUC_S3**			**Speech_BCIC2020_3**		
		B-ACC	Kappa	F1	B-ACC	Kappa	F1
CodeBrain	Baseline	0.7172±0.0126	0.5896±0.0181	0.6566±0.0145	0.4836±0.0094	0.3544±0.0117	0.4824±0.0084
	GACE	**0.7374±0.0177**	**0.6168±0.0284**	**0.6804±0.0267**	**0.4844±0.0267**	**0.3556±0.0334**	**0.4837±0.0276**
	Δ	+0.0202	+0.0272	+0.0238	+0.0008	+0.0012	+0.0013
CBraMod	Baseline	0.7115±0.0320	0.5945±0.0376	0.6706±0.0273	0.4209±0.0121	0.2761±0.0151	0.4197±0.0118
	GACE	**0.7235±0.0457**	**0.6241±0.0531**	**0.7005±0.0405**	**0.4338±0.0191**	**0.2922±0.0238**	**0.4334±0.0191**
	Δ	+0.0120	+0.0296	+0.0299	+0.0129	+0.0161	+0.0137
CSBrain	Baseline	0.7013±0.0135	0.5649±0.0099	0.6384±0.0012	0.4716±0.0020	0.3394±0.0025	0.4716±0.0016
	GACE	**0.7235±0.0297**	**0.6132±0.0274**	**0.6898±0.0176**	**0.4840±0.0074**	**0.3550±0.0093**	**0.4833±0.0075**
	Δ	+0.0221	+0.0482	+0.0514	+0.0124	+0.0156	+0.0117

4.1 EXPERIMENTAL SETUP

Datasets and Tasks: We report results on six EEG tasks: **attention recognition** (ATTENTION (Shin et al., 2018)), **motor imagery** (BCI-IV-2a (Brunner et al., 2008)), **emotion recognition** (SEED (Zheng & Lu, 2015), using a binary protocol (Liu et al., 2026a)), **sleep staging** (ISRUC-S3 (Khalighi et al., 2016)), **major depressive disorder classification** (Mumtaz2016 (Mumtaz, 2016)), and **speech decoding** (Speech-BCIC2020-3 (Jeong et al., 2022)). Supplementary datasets, including Cho2017-MI, are described in Appendix C.

Backbones and Baselines: We evaluate GACE on CodeBrain (Ma et al., 2026), CBraMod (Wang et al., 2025a), and CSBrain (Zhou et al., 2025c). Baselines use standard fine-tuning of the same pretrained backbones, with identical data splits and preprocessing (Appendix B).

Metrics and Reporting: We report B-ACC, Kappa, and weighted F1. Multi-seed summaries use seeds $(0, 1, 2)$ and report mean±sample standard deviation where available; mean-only entries use the same three seeds. The reported Δ denotes the gain over the corresponding baseline.

Implementation Details: GACE keeps the backbone architecture unchanged. Self-Assessment is trained on the calibration set and selects the highest-risk 12% of calibration samples through internal K-fold risk scoring. Self-Correction is trained on this selected calibration subset. The checkpoint epoch for each head is selected by internal K-fold validation on the calibration set. Hyperparameters and the complete procedure are provided in Appendices G and D, respectively.

Table 2: Stage-wise comparison across six tasks and three seeds, with equal task weights. Correction gains are relative to self-guided predictions. Best per stage and backbone in bold.

Backbone	Training: B-ACC (%)			Risk identification: RP@12% (%)					
	Cross-entropy	Active Bias SGD-WPV[a]	Self-Guidance	Random	MSP[b]	TS-MSP[c]	TrustScore[d]	Mahalanobis[e]	Self-Assessment
CodeBrain	61.86	62.14	**63.59**	36.50	54.46	54.88	49.14	33.77	**63.29**
CSBrain	63.77	63.61	**65.20**	35.06	54.48	54.67	48.30	36.68	**63.81**
CBraMod	62.41	63.66	**64.62**	35.37	54.14	54.66	50.13	35.16	**62.05**

Backbone	Correction: ΔB-ACC (pp)					
	Second Softmax	Trajectory Top	Probability Margin	Entropy Trigger	Nearest Mahalanobis[e]	Self-Correction
CodeBrain	+1.59	+1.11	+0.15	+0.47	+1.67	**+2.24**
CSBrain	+1.96	+1.92	+0.16	+1.48	+1.71	**+2.27**
CBraMod	+1.70	+1.60	+0.12	+0.34	+1.45	**+2.27**

[a](Chang et al., 2017); [b](Hendrycks & Gimpel, 2017); [c](Guo et al., 2017); [d](Jiang et al., 2018); [e](Lee et al., 2018). TS: temperature-scaled; pp: percentage points.

Table 3: CodeBrain component ablations (three-seed mean±sample SD). Blue cells fall below the full model. Random uses reproducible 12% random selection in calibration and test for correction.

Variant	Mumtaz2016			BCI_IV_2a		
	B-ACC	Kappa	F1	B-ACC	Kappa	F1
Full	0.9592±0.0140	0.8866±0.0202	0.9512±0.0085	0.4705±0.0211	0.2940±0.0282	0.4403±0.0326
Baseline	0.8492±0.0137	0.7479±0.0191	0.8963±0.0075	0.4048±0.0152	0.2064±0.0203	0.3467±0.0409
+ Self-Guidance	0.8691±0.0125	0.7699±0.0220	0.9045±0.0091	0.4447±0.0289	0.2596±0.0385	0.4062±0.0397
w/o Self-Assessment (Random)	0.8770±0.0153	0.7757±0.0253	0.9064±0.0104	0.4372±0.0183	0.2496±0.0245	0.4117±0.0247

4.2 Performance Comparison with Baseline

(a) Overall performance across backbones and tasks. Table 1 shows improvements in all 18 settings, averaging 0.0405, 0.0574, and 0.0386 in B-ACC, Kappa, and weighted F1. These gains use unchanged backbone architectures and are larger on Mumtaz2016 and BCI-IV-2a than on sleep staging and speech decoding, suggesting greater benefits where initial predictions leave more room for error detection and correction.

(b) Comparison with stage-wise alternatives. Table 2 compares training, risk identification, and correction methods. Our approach leads each stage's aggregate metric for all three backbones; correction adds 2.24–2.27 B-ACC percentage points over self-guided predictions. Appendix H.1 provides implementations and per-task results.

4.3 Ablation Study

We examine GACE through framework-level ablations of self-guidance, risk selection, and correction, and input-group ablations within self-assessment and self-correction. Detailed results are provided in Appendices I.1 and I.3.

Framework-Level Ablation: Table 3 shows that **Self-Guidance** alone improves both tasks. Replacing risk selection with random selection of 12% of samples slightly raises B-ACC on Mumtaz2016 but lowers it on BCI-IV-2a relative to Self-Guidance alone; the full method outperforms random selection on both tasks. These results indicate that Self-Guidance improves the model from which internal cues are extracted, while learned risk selection helps subsequent correction yield gains.

Operator-Level Ablation: We further ablate features derived from Decision Evidence $\mathbf{E}(\mathbf{X})$ and trajectory statistics $\Gamma_{\mathrm{T}}(\mathbf{T}(\mathbf{X}))$ in Self-Assessment and Self-Correction. Across seven datasets, removing either group lowers the mean correction gain, and removing both groups lowers it further on average. Removing the evidence features also lowers mean RP@12%. The results appear in Appendix I.3.

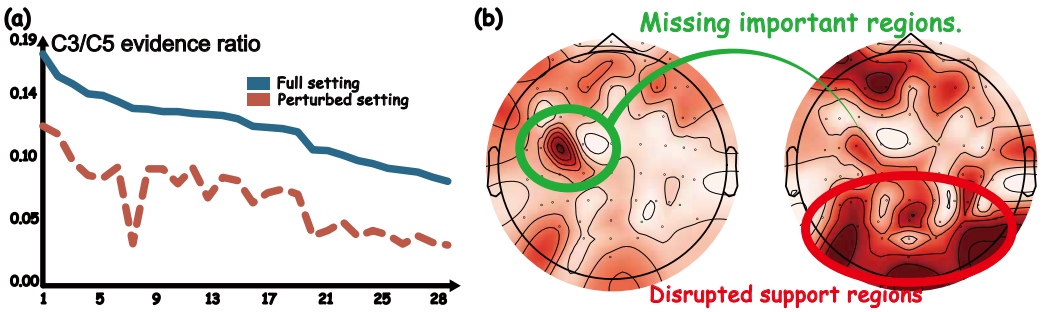

Figure 4: Self-Guidance and Decision Evidence: **(a)** task-relevant channel evidence ratios after removing high-weight training samples; **(b)** representative full/drop comparisons showing more dispersed attribution after removal.

4.4 Self-Guidance and Internal Decision Cues

Figure 4 examines how samples emphasized during Self-Guidance relate to the spatial evidence learned through fine-tuning. In the reported comparison, removing training samples with high online weights reduces the concentration of Decision Evidence on task-relevant channels such as C3/C5. The paired examples show more dispersed channel attributions after removal, supporting the role of Self-Guidance in promoting clearer task-related spatial cues for subsequent assessment.

The matched-sample analysis further shows that samples with higher layer-aggregated trajectory alignment scores tend to receive larger margin-based weights during fine-tuning. Together, these observations connect the weighting used by Self-Guidance to task-relevant spatial reliance and cross-layer alignment patterns, providing empirical support for its role as a preparatory fine-tuning stage aimed at clearer Decision Evidence and Decision Trajectory. A matched comparison with ordinary CE on the same full training data also finds greater correct–incorrect mean spatial-attribution separation under Self-Guidance on the three reported datasets (Table 24). More visualizations and quantitative analyses are provided in Appendix L.

5 Related Work

EEG modeling has shifted from task-specific pipelines toward EEG foundation models (EFMs), which learn transferable representations from heterogeneous EEG corpora (Yang et al., 2025; Zhou et al., 2025b; Liu et al., 2026b). Recent EFMs improve backbone architectures, tokenization, and scalable pretraining (Wang et al., 2025a; Zhou et al., 2025c; Ma et al., 2026), but most work still evaluates EFM knowledge mainly through downstream accuracy, leaving internal decision structures underexplored.

Foundation-model research also shows that internal signals can support reliability estimation, calibration, and self-correction (Wu et al., 2024; Gui et al., 2024c; Madaan et al., 2023). Yet these mechanisms are mostly developed for vision, language, or generic prediction settings; LLM-style self-correction relies on textual outputs that can be fed back into the model (Huang et al., 2023). For EFMs, we examine spatial evidence and cross-layer logit patterns as complementary cues, and use self-guided fine-tuning to encourage clearer expressions of these cues before assessment and correction. Details are provided in Appendix A.

6 Conclusion

We presented GACE, a plug-and-play framework that reuses EFM internal decision structures. Self-Guidance uses class support and response stability to encourage clearer internal cues; Self-Assessment uses spatial evidence to assess reliability; and Self-Correction uses cross-layer alignment features to revise high-risk predictions. Across EEG tasks and backbones, these cues improve downstream performance without architectural changes, demonstrating the value of EFM knowledge beyond final outputs.

AI USE STATEMENT

We used ChatGPT, Codex, and Claude to assist with implementing and debugging experimental and analysis code, polishing manuscript text, preparing and refining scientific figures, checking manuscript consistency, and organizing the manuscript structure. The authors are responsible for the accuracy of all claims, citations, figures, and experimental results in the final manuscript, including content prepared with AI assistance.

ETHICS STATEMENT

This study uses previously released EEG datasets and does not collect new human-subject data or redistribute raw EEG recordings. EEG data can be privacy-sensitive, and the proposed risk assessment and correction are research tools rather than clinical diagnostic systems. They should not be used for clinical or other safety-critical decisions without separate validation. Dataset sources and usage terms, along with broader limitations and impacts, are discussed in Appendix O.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe the method and evaluation protocol. Appendix O.1 documents the dataset splits, preprocessing, calibration procedure, seeds, and implementation settings. Appendix O.1 links an anonymized supplementary package with code, configurations, and reproduction commands; the sources and usage terms for externally obtained datasets and pretrained backbones are listed in Table 28.

REFERENCES

Salma Alhagry, Aly Aly Fahmy, and Reda A El-Khoribi. Emotion recognition based on eeg using lstm recurrent neural network. *International Journal of Advanced Computer Science and Applications*, 8(10), 2017.

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *The Eleventh International Conference on Learning Representations*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Naseem Babu, Jimson Mathew, and AP Vinod. Large language models for eeg: A comprehensive survey and taxonomy. *arXiv preprint arXiv:2506.06353*, 2025.

Yarin Bar, Shalev Shaer, and Yaniv Romano. Protected test-time adaptation via online entropy matching: A betting approach. *Advances in Neural Information Processing Systems*, 37:85467–85499, 2024.

Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, and Gert Pfurtscheller. Bci competition 2008–graz data set a. *Institute for knowledge discovery (laboratory of brain-computer interfaces), Graz University of Technology*, 16(1-6):34, 2008.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*, 2017.

Chi-Sheng Chen, Ying-Jung Chen, and Aidan Hung-Wen Tsai. Large cognition model: Towards pretrained eeg foundation model. *arXiv preprint arXiv:2502.17464*, 2025.

Yan Chen, Jie Peng, Moajjem Hossain Chowdhury, Tianlong Chen, and Yunmei Liu. Neurocanvas: Vllm-powered robust seizure detection by reformulating multichannel eeg as image. *arXiv preprint arXiv:2602.04769*, 2026a.

Zhisheng Chen, Yingwei Zhang, Qizhen Lan, Tianyu Liu, Huacan Wang, Yi Ding, Ziyu Jia, Ronghao Chen, Kun Wang, and Xinliang Zhou. Uni-ntfm: A unified foundation model for eeg signal representation learning. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Hsiang-Yun Sherry Chien, Hanlin Goh, Christopher Michael Sandino, and Joseph Yitan Cheng. Maeeg: Masked auto-encoder for eeg representation learning. In *NeurIPS 2022 Workshop on Learning from Time Series for Health*, 2022.

Hohyun Cho, Minkyu Ahn, Sangtae Ahn, Moonyoung Kwon, and Sung Chan Jun. EEG datasets for motor imagery brain-computer interface. *GigaScience*, 6(7):1–8, 2017. doi: 10.1093/gigascience/gix034.

Mingzhe Cui, Tao Chen, Yang Jiao, Yiqin Wang, Lei Xie, Yi Pan, and Luca Mainardi. Brainrvq: A high-fidelity eeg foundation model via dual-domain residual quantization and hierarchical autoregression. *arXiv preprint arXiv:2602.16951*, 2026.

Wenhui Cui, Woojae Jeong, Philipp Thölke, Takfarinas Medani, Karim Jerbi, Anand A Joshi, and Richard M Leahy. Neuro-gpt: Towards a foundation model for eeg. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5. IEEE, 2024.

Xia Deng, Shen Chen, Jiale Zhou, and Lei Li. Mind2matter: Creating 3d models from eeg signals. *arXiv preprint arXiv:2504.11936*, 2025.

Yi Ding, Yong Li, Hao Sun, Rui Liu, Chengxuan Tong, Chenyu Liu, Xinliang Zhou, and Cuntai Guan. Eeg-deformer: A dense convolutional transformer for brain-computer interfaces. *IEEE Journal of Biomedical and Health Informatics*, 29(3):1909–1918, 2024.

Yi Ding, Muyun Jiang, Weibang Jiang, Shuailei Zhang, Xinliang Zhou, Chenyu Liu, Shanglin Li, Yong Li, and Cuntai Guan. Brainpro: Towards large-scale brain state-aware eeg representation learning. *arXiv preprint arXiv:2509.22050*, 2025.

Berkay Döner, Thorir Mar Ingolfsson, Luca Benini, and Yawei Li. Luna: Efficient and topology-agnostic foundation model for eeg signal analysis. In *Advances in Neural Information Processing Systems*, 2025.

Danny Dongyeop Han, Ahhyun Lucy Lee, Taeyang Lee, Yonghyeon Gwon, Sebin Lee, Seongjin Lee, David Keetae Park, Shinjae Yoo, Jiook Cha, and Chun Kee Chung. Diver-0: A fully channel equivariant eeg foundation model. *arXiv e-prints*, pp. arXiv–2507, 2025.

Yassine El Ouahidi, Jonathan Lys, Philipp Thölke, Nicolas Farrugia, Bastien Pasdeloup, Vincent Gripon, Karim Jerbi, and Giulia Lioi. Reve: A foundation model for eeg-adapting to any setup with large-scale pretraining on 25,000 subjects. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *Advances in Neural Information Processing Systems*, 36:16222–16239, 2023.

Zitao Fang, CHENXUAN LI, Zhou Hongting, Shuyang Yu, Guodong DU, Ashwaq Qasem, Yang Lu, Jing Li, Junsong Zhang, and Sim Kuan Goh. Neuript: Foundation model for neural interfaces. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Sam Gijsen and Kerstin Ritter. Eeg-language pretraining for highly label-efficient clinical phenotyping. In *Forty-second International Conference on Machine Learning*, 2025.

Taesik Gong, Yewon Kim, Taeckyung Lee, Sorn Chottananurak, and Sung-Ju Lee. Sotta: Robust test-time adaptation on noisy data streams. *Advances in Neural Information Processing Systems*, 36:14070–14093, 2023.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819, 2021.

Yuxian Gu, Li Dong, Hongning Wang, Yaru Hao, Qingxiu Dong, Furu Wei, and Minlie Huang. Data selection via optimal control for language models. In *The Thirteenth International Conference on Learning Representations*.

Haokun Gui, Xiucheng Li, and Xinyang Chen. Vector quantization pretraining for eeg time series with random projection and phase alignment. In *ICML*, 2024a.

Yiyu Gui, MingZhi Chen, Yuqi Su, Guibo Luo, and Yuchao Yang. Eegmamba: Bidirectional state space model with mixture of experts for eeg multi-task classification. *arXiv preprint arXiv:2407.20254*, 2024b.

Yu Gui, Ying Jin, and Zhimei Ren. Conformal alignment: Knowing when to trust foundation models with guarantees. *Advances in Neural Information Processing Systems*, 37:73884–73919, 2024c.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.

Ji-Hoon Jeong, Jeong-Hyun Cho, Young-Eun Lee, Seo-Hyun Lee, Gi-Hwan Shin, Young-Seok Kweon, José del R. Millán, Klaus-Robert Müller, and Seong-Whan Lee. 2020 international brain-computer interface competition: A review. *Frontiers in Human Neuroscience*, 16, 2022.

Ziyu Jia, Youfang Lin, Jing Wang, Ronghao Zhou, Xiaojun Ning, Yuanlai He, and Yaoshuai Zhao. Graphsleepnet: Adaptive spatial-temporal graph convolutional networks for sleep stage classification. In *Ijcai*, volume 2021, pp. 1324–1330, 2020.

Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems*, 2018.

Muyun Jiang, Shuailei Zhang, Zhenjie Yang, Mengjun Wu, Weibang Jiang, Zhiwei Guo, Wei Zhang, Rui Liu, Shangen Zhang, Yong Li, et al. Elastiq: Eeg-language alignment with semantic task instruction and querying. *arXiv preprint arXiv:2509.24302*, 2025a.

Weibang Jiang, Liming Zhao, and Bao-liang Lu. Large brain model for learning generic representations with tremendous eeg data in bci. In *The Twelfth International Conference on Learning Representations*, 2024.

Weibang Jiang, Yansen Wang, Bao-liang Lu, and Dongsheng Li. Neurolm: A universal multi-task foundation model for bridging the gap between language and eeg signals. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*.

Sirvan Khalighi, Teresa Sousa, José Moutinho dos Santos, and Urbano J. Nunes. Isruc-sleep: A comprehensive public dataset for sleep researchers. *Computer methods and programs in biomedicine*, 124:180–92, 2016.

S. Koelstra, C. Muhl, M. Soleymani, Jong-Seok Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras. DEAP: A Database for Emotion Analysis ;Using Physiological Signals. *IEEE Transactions on Affective Computing*, 3(1):18–31, January 2012. ISSN 1949-3045. doi: 10.1109/T-AFFC.2011.15. URL `http://ieeexplore.ieee.org/document/5871728/`.

Demetres Kostas, Stephane Aroca-Ouellette, and Frank Rudzicz. Bendr: Using transformers and a contrastive self-supervised learning task to learn from massive amounts of eeg data. *Frontiers in Human Neuroscience*, 15:653659, 2021.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 54523–54549, 2025.

Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

Hyunseok Lee, Seunghyuk Oh, Jaehyung Kim, Jinwoo Shin, and Jihoon Tack. Revise: Learning to refine at test-time via intrinsic self-verification. *Forty-Second International Conference on Machine Learning*, 2025.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, 2018.

Min-Ho Lee, O-Yeon Kwon, Yong-Jeong Kim, Hong-Kyung Kim, Young-Eun Lee, John Williamson, Siamac Fazli, and Seong-Whan Lee. EEG dataset and OpenBMI toolbox for three BCI paradigms: an investigation into BCI illiteracy. *GigaScience*, 8(5):giz002, May 2019. ISSN 2047-217X. doi: 10.1093/gigascience/giz002. URL https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giz002/5304369.

Aoyang Liu, Haodong Jing, Yulong Liu, Yongqiang Ma, and Nanning Zheng. Hidden states in llms improve eeg representation learning and visual decoding. In *ECAI*, volume 392, pp. 2130–7, 2024a.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 57226–57243. Curran Associates, Inc., 2023.

Chenyu Liu, Xinliang Zhou, Jiaping Xiao, Zhengri Zhu, Liming Zhai, Ziyu Jia, and Yang Liu. Vsgt: Variational spatial and gaussian temporal graph models for eeg-based emotion recognition. In *IJCAI*, pp. 3078–3086, 2024b.

Chenyu Liu, Yuqiu Deng, Tianyu Liu, Jinan Zhou, Xinliang Zhou, Ziyu Jia, and Yi Ding. Echo: Toward contextual seq2seq paradigms in large eeg models. In *The Fourteenth International Conference on Learning Representations*, 2026a.

Dingkun Liu, Yuheng Chen, Zhu Chen, Zhenyao Cui, Yaozhi Wen, Jiayu An, Jingwei Luo, and Dongrui Wu. Eeg foundation models: Progresses, benchmarking, and open problems. *arXiv preprint arXiv:2601.17883*, 2026b.

Zhining Liu, Ze Yang, Xiao Lin, Ruizhong Qiu, Tianxin Wei, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. Breaking silos: Adaptive model fusion unlocks better time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

Fabien Lotte, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche, and Bruno Arnaldi. A review of classification algorithms for eeg-based brain–computer interfaces. *Journal of neural engineering*, 4(2):R1–R13, 2007.

Weiheng Lu, Chunfeng Song, Jiamin Wu, Pengyu Zhu, Yuchen Zhou, Weijian Mai, Qihao Zheng, and Wanli Ouyang. Unimind: Unleashing the power of llms for unified multi-task brain decoding. *arXiv preprint arXiv:2506.18962*, 2025.

Jingying Ma, Feng Wu, Qika Lin, Yucheng Xing, Chenyu Liu, Ziyu Jia, and Mengling Feng. Codebrain: Bridging decoupled tokenizer and multi-scale architecture for eeg foundation model. In *The Fourteenth International Conference on Learning Representations*, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Abhijit Mishra, Shreya Shukla, Jose Torres, Jacek Gwizdka, and Shounak Roychowdhury. Thought2text: text generation from eeg signal using large language models (llms). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3747–3759, 2025.

Navid Mohammadi Foumani, Geoffrey Mackellar, Soheila Ghane, Saad Irtza, Nam Nguyen, and Mahsa Salehi. Eeg2rep: enhancing self-supervised eeg representation through informative masked inputs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5544–5555, 2024.

Amir Moslemi, Anna Briskina, Zubeka Dang, and Jason Li. A survey on knowledge distillation: Recent advancements. *Machine Learning with Applications*, 18:100605, 2024. ISSN 2666-8270. doi: https://doi.org/10.1016/j.mlwa.2024.100605.

Wajid Mumtaz. MDD Patients and Healthy Controls EEG Data (New), November 2016.

Mathias Perslev, Sune Darkner, Lykke Kempfner, Miki Nikolic, Poul Jørgen Jennum, and Christian Igel. U-sleep: resilient high-frequency sleep staging. *NPJ digital medicine*, 4(1):72, 2021.

Jathurshan Pradeepkumar, Xihao Piao, Zheng Chen, and Jimeng Sun. Tokenizing single-channel eeg with time-frequency motif learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Simanto Saha and Mathias Baumert. Intra- and Inter-subject Variability in EEG-Based Sensorimotor Brain Computer Interface: A Review. *Frontiers in Computational Neuroscience*, 13:87, January 2020. ISSN 1662-5188. doi: 10.3389/fncom.2019.00087. URL https://www.frontiersin.org/article/10.3389/fncom.2019.00087/full.

Qichao Shentu, Beibu Li, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jaeyoung Shin, Alexander von Lühmann, Do-Won Kim, Jan Mehnert, Han-Jeong Hwang, and Klaus-Robert Müller. Simultaneous acquisition of eeg and nirs during cognitive tasks for an open access dataset. *Scientific Data*, 5, 2018.

Afshin Shoeibi, Marjane Khodatars, Navid Ghassemi, Mahboobeh Jafari, Parisa Moridian, Roohallah Alizadehsani, Maryam Panahiazar, Fahime Khozeimeh, Assef Zare, Hossein Hosseini-Nejad, et al. Epileptic seizures detection using deep learning techniques: a review. *International journal of environmental research and public health*, 18(11):5780, 2021.

Yonghao Song, Qingqing Zheng, Bingchuan Liu, and Xiaorong Gao. Eeg conformer: Convolutional transformer for eeg decoding and visualization. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:710–719, 2022.

Kostas M Tsiouris, Vasileios C Pezoulas, Michalis Zervakis, Spiros Konitsiotis, Dimitrios D Koutsouris, and Dimitrios I Fotiadis. A long short-term memory deep learning network for the prediction of epileptic seizures using eeg signals. *Computers in biology and medicine*, 99:24–37, 2018.

Lars van der Laan and Ahmed M Alaa. Self-calibrating conformal prediction. *Advances in Neural Information Processing Systems*, 37:107138–107170, 2024.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. Knowledge fusion of large language models. *ArXiv*, abs/2401.10491, 2024.

Guangyu Wang, Wenchao Liu, Yuhong He, Cong Xu, Lin Ma, and Haifeng Li. Eegpt: Pretrained transformer for universal and reliable representation of eeg signals. *Advances in Neural Information Processing Systems*, 37:39249–39280, 2024a.

Jiaqi Wang, Zhenxi Song, Zhengyu Ma, Xipeng Qiu, Min Zhang, and Zhiguo Zhang. Enhancing eeg-to-text decoding through transferable representations from pre-trained contrastive eeg-text masked autoencoder. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7278–7292, 2024b.

Jiquan Wang, Sha Zhao, Zhiling Luo, Yangxuan Zhou, Haiteng Jiang, Shijian Li, Tao Li, and Gang Pan. Cbramod: A criss-cross brain foundation model for eeg decoding. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Jiquan Wang, Sha Zhao, Zhiling Luo, Yangxuan Zhou, Shijian Li, and Gang Pan. Eegmamba: An eeg foundation model with mamba. *Neural Networks*, pp. 107816, 2025b.

Jiquan Wang, Sha Zhao, Yangxuan Zhou, Yiming Kang, Shijian Li, and Gang Pan. Deeperbrain: A neuro-grounded eeg foundation model towards universal bci. *arXiv preprint arXiv:2601.06134*, 2026.

Yifei Wang, Yuyang Wu, Zeming Wei, Stefanie Jegelka, and Yisen Wang. A theoretical understanding of self-correction through in-context alignment. *Advances in Neural Information Processing Systems*, 37:89869–89912, 2024c.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Yu-Chang Wu, Shen-Huan Lyu, Haopu Shang, Xiangyu Wang, and Chao Qian. Confidence-aware contrastive learning for selective classification. *arXiv preprint arXiv:2406.04745*, 2024.

Lichao Xu, Minpeng Xu, Yufeng Ke, Xingwei An, Shuang Liu, and Dong Ming. Cross-dataset variability problem in eeg decoding with deep learning. *Frontiers in human neuroscience*, 14:103, 2020.

Chaoqi Yang, M Westover, and Jimeng Sun. Biot: Biosignal transformer for cross-data learning in the wild. *Advances in Neural Information Processing Systems*, 36:78240–78260, 2023.

Shihao Yang, Xiying Huang, Danilo Bernardo, Jun-En Ding, Andrew Michael, Jingmei Yang, Patrick Kwan, Ashish Raj, and Feng Liu. Foundation and large-scale ai models in neuroscience: A comprehensive review. *arXiv preprint arXiv:2510.16658*, 2025.

Jianyi Zhang, Da-Cheng Juan, Cyrus Rashtchian, Chun-Sung Ferng, Heinrich Jiang, and Yiran Chen. Sled: Self logits evolution decoding for improving factuality in large language models. In *Advances in Neural Information Processing Systems*, 2024.

Wei-Long Zheng and Bao-Liang Lu. Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks. *IEEE Transactions on autonomous mental development*, 7(3):162–175, 2015.

Jianheng Zhou, Chenyu Liu, Jinan Zhou, Yi Ding, Yang Liu, Haoran Luo, Ziyu Jia, and Xinliang Zhou. Introducing multimodal paradigm for learning sleep staging psg via general-purpose model. *arXiv preprint arXiv:2509.22810*, 2025a.

Xinliang Zhou, Yuzhe Han, Zhisheng Chen, Chenyu Liu, Yi Ding, Ziyu Jia, and Yang Liu. BiT-MamSleep: Bidirectional Temporal Mamba for EEG Sleep Staging, November 2024. URL http://arxiv.org/abs/2411.01589. arXiv:2411.01589 [eess].

Xinliang Zhou, Chenyu Liu, Zhisheng Chen, Kun Wang, Yi Ding, Ziyu Jia, and Qingsong Wen. Brain foundation models: A survey on advancements in neural signal processing and brain discovery. *IEEE Signal Processing Magazine*, 42(5):22–35, 2025b.

Yuchen Zhou, Jiamin Wu, Zichen Ren, Zhouheng Yao, Weiheng Lu, Kunyu Peng, Qihao Zheng, Chunfeng Song, Wanli Ouyang, and Chao Gou. Csbrain: A cross-scale spatiotemporal brain foundation model for eeg decoding. In *Advances in Neural Information Processing Systems*, 2025c.

APPENDIX CONTENTS

A Detailed Related Work . . . 17
A.1 Scalp EEG Foundation Models . . . 17
A.2 Internal Signals for Guidance, Assessment, and Correction . . . 18
B Baselines and Metrics . . . 19
B.1 Baselines . . . 19
B.1.1 EEG Foundation Model Backbones . . . 19
B.1.2 Baseline Construction . . . 19
B.2 Metrics . . . 19
C Dataset and Experimental Settings . . . 20
C.1 Detailed Dataset Descriptions . . . 20
D Implementation Procedure of GACE . . . 21
D.1 Self-Guidance . . . 21
D.2 Self-Assessment . . . 21
D.3 Self-Correction . . . 23
E Computation of Decision Signals . . . 23
E.1 Computation of Decision Evidence . . . 23
E.2 Computation of Decision Trajectory . . . 24
E.3 Online Training Statistics for Self-Guidance . . . 25
F Self-Assessment and Self-Correction Details . . . 26
F.1 Risk Modeling Details for Self-Assessment . . . 26
F.2 Correction Modeling Details for Self-Correction . . . 28
G Hyperparameter Settings . . . 29
H Extended Results on Downstream EEG Tasks . . . 29
H.1 Competitor Results by Task . . . 30
I Ablation Study . . . 32
I.1 Component-Level Ablation . . . 32
I.2 Selection-Strategy Ablation . . . 32
I.3 Operator-Level Ablation . . . 32
I.4 Correction-Strategy Ablation . . . 35
I.5 Joint Calibration-Label Sensitivity . . . 36
J Reliability and Correction Diagnostics . . . 37
J.1 Self-Assessment Risk Ranking under Different Rejection Rates . . . 37
J.2 Stability and Safety Diagnostics . . . 37
K Detailed Analysis of Decision Patterns . . . 39
K.1 Evidence: Individual Spatial Patterns behind Group Averages . . . 39
K.1.1 Mean Spatial Attribution and Outcome Contrasts across Models and Tasks . . . 42
K.1.2 Self-Guidance and Correct–Incorrect Attribution Separation . . . 52
K.2 Cross-Layer Class Competition across Models and Datasets . . . 53
K.3 Self-Guidance Motivation: Variation in the Time to Persistent Correctness . . . 59
K.3.1 Self-Guidance Motivation: Training Responses across Models and Datasets . . . 63
L Decision-Structure Correlation Analysis . . . 70
M Efficiency and Complexity Analysis . . . 71
M.1 Computational Overhead . . . 71
N Discussion . . . 71
O Limitations, Ethics, and Reproducibility . . . 72
O.1 Code, Data, and Reproducibility . . . 72
O.2 Compute Resources . . . 74

A Detailed Related Work

A.1 Scalp EEG Foundation Models

Electroencephalography (EEG) modeling has undergone a substantial paradigm shift, moving from task-specific supervised pipelines toward general-purpose representation learning and, more recently, EEG foundation models (EFMs) pretrained on heterogeneous EEG corpora (Yang et al., 2025; Zhou et al., 2025b; Liu et al., 2026b).

Early EEG analysis was dominated by task-specific supervised methods. Different downstream applications, such as sleep staging (Jia et al., 2020; Perslev et al., 2021), seizure detection (Tsiouris et al., 2018; Shoeibi et al., 2021), emotion recognition (Zheng & Lu, 2015; Liu et al., 2024b), and motor imagery decoding (Lotte et al., 2007), typically required separately designed features, models, and training protocols. Classical pipelines relied heavily on handcrafted features and conventional classifiers (Lotte et al., 2007). With the rise of deep learning, end-to-end architectures became increasingly common, including convolutional neural networks (CNNs) (Lawhern et al., 2018), recurrent neural networks (RNNs) (Alhagry et al., 2017), and Transformer-based models (Song et al., 2022; Ding et al., 2024). Although these methods improved task-level performance, they often remained tied to specific tasks or data distributions, and models trained in one setting rarely generalized robustly across tasks, subjects, acquisition protocols, or electrode configurations (Xu et al., 2020). Their dependence on costly EEG annotations further limited scalability.

To reduce annotation dependence and improve transferability, EEG research has increasingly moved toward self-supervised representation learning. These methods train EEG encoders on unlabeled recordings through pretext objectives such as contrastive learning (Kostas et al., 2021), masked reconstruction (Chien et al., 2022), or masked segment prediction (Mohammadi Foumani et al., 2024). By learning reusable neural representations before downstream adaptation, self-supervised learning provides an important basis for scalable EEG pretraining.

Building on these developments, recent EFMs aim to learn unified and transferable representations from diverse EEG datasets. Existing work has advanced this goal from several directions. At the architectural level, EFMs have explored Transformer-based backbones (Yang et al., 2023; Wang et al., 2024a; 2025a; Zhou et al., 2025c; Chen et al., 2025), state-space architectures (Wang et al., 2025b; Gui et al., 2024b), and modular routing mechanisms such as mixture-of-experts (Fang et al., 2025; Chen et al., 2026b). In parallel, several studies examine how EEG should be represented before pretraining, including patch construction, continuous tokenization, and discrete neural tokens (Jiang et al., 2024; Gui et al., 2024a; Ma et al., 2026; Pradeepkumar et al., 2026; Cui et al., 2026). Beyond architecture and tokenization, the field has also moved toward scaling pretraining to diverse EEG corpora (El Ouahidi et al., 2025), accommodating heterogeneous electrode layouts (Döner et al., 2025; Dongyeop Han et al., 2025), and incorporating neuroscience-inspired structural priors (Ding et al., 2025; Wang et al., 2026).

A related but distinct line of work studies the interaction between EEG representations and large language models (LLMs). LLMs introduce sequence-level semantic reasoning capabilities (Babu et al., 2025), and recent studies have explored EEG-to-language decoding (Mishra et al., 2025; Wang et al., 2024b; Gijsen & Ritter, 2025), cross-modal generation from neural activity (Liu et al., 2024a; Deng et al., 2025), and vision-language-guided EEG interpretation (Zhou et al., 2025a; Chen et al., 2026a). More recent formulations connect EEG foundation models with language models through neural tokenization for multi-task decoding (Jiang et al., 2025b), alignment between EEG representations and language embeddings (Lu et al., 2025; Jiang et al., 2025a), and decoder-centric modeling (Cui et al., 2024; Liu et al., 2026a). These studies broaden the interface between EEG signals and semantic systems, but they primarily focus on cross-modal alignment or language-based task interfaces rather than the internal decision structure of EFMs themselves.

Overall, existing EFM research has demonstrated that large-scale EEG pretraining can encode transferable spatiotemporal representations and task-relevant neural priors. However, EFM knowledge is most often evaluated through final downstream performance after fine-tuning. In this view, the EFM is treated mainly as a pretrained feature extractor or prediction backbone, while the internal structure formed during prediction remains relatively underexplored. For example, the spatial evidence supporting a prediction and the layer-wise evolution of class support are rarely modeled as explicit

reliability signals. Understanding and using these internal decision structures remains an important open direction for improving the reliability of downstream EFM decisions.

A.2 Internal Signals for Guidance, Assessment, and Correction

Beyond building larger pretrained backbones, recent foundation-model research has increasingly reused model-internal information as an optimization or adaptation resource. This idea appears in knowledge distillation and self-distillation, where teacher predictions or intermediate behaviors guide another model or training instance (Hinton et al., 2015; Gou et al., 2021; Moslemi et al., 2024; Allen-Zhu & Li); in model merging and knowledge fusion, where information stored in trained weights is reused without retraining from scratch (Wortsman et al., 2022; Jin et al.; Wan et al., 2024); and in data selection or adaptive optimization, where training is steered by estimated sample utility or task difficulty (Gu et al.; Liu et al., 2023). Related ideas also appear in self-guided generation and robust adaptation, where internal signals or online statistics are used to regulate model behavior under changing conditions (Epstein et al., 2023; Gong et al., 2023; Bar et al., 2024). In time-series modeling, adaptive fusion and robust anomaly detection further reflect the broader need to exploit model responses under heterogeneous temporal distributions (Liu et al., 2025; Shentu et al., 2025). These works provide useful methodological precedents, but they usually target language, vision, or generic time-series settings rather than EEG-specific decision structures.

Another related line studies whether a model can estimate the reliability of its own prediction. In conventional classification and foundation-model settings, this problem is often formulated through confidence scoring, selective classification, calibration, conformal prediction, or risk estimation based on output probabilities and uncertainty statistics (Wu et al., 2024; Gui et al., 2024c; van der Laan & Alaa, 2024). These approaches are broadly applicable when the final prediction distribution is the main observable object, and they typically rely on quantities such as maximum probability, entropy, margin, or calibrated conformity scores. For EFMs, however, the predicted class distribution is only one part of the observable decision structure. EEG predictions are also supported by spatially organized neural evidence, so reliability assessment can benefit from signals that describe where the evidence arises, not only how confident the final classifier appears.

Model correction has been studied most actively in large language models. Because LLM outputs are natural-language sequences, generated answers can be critiqued, revised, retrieved against external evidence, or fed back into the same model as text (Madaan et al., 2023; Asai et al., 2024). Recent work further analyzes both the limits and mechanisms of such self-correction behavior (Huang et al., 2023; Wang et al., 2024c). This paradigm is highly dependent on the language interface: the model can inspect and rewrite its own textual output. EFMs have a different input-output structure. Their inputs are EEG signals, while their outputs are logits or labels over task classes; a predicted label cannot be naturally reintroduced as a semantically rich EEG input. Correction for EFMs therefore requires modality-specific internal cues, such as how class support is formed across layers, rather than direct textual self-feedback.

In large language models, prior work investigates self-correction through feedback, retrieval, and internal prediction signals, alongside analyses of its capabilities and limitations (Madaan et al., 2023; Asai et al., 2024; Huang et al., 2023; Wang et al., 2024c; Zhang et al., 2024).

Overall, these related directions offer important conceptual ingredients but not a directly transferable framework for EFMs. Training-time guidance, post-hoc reliability assessment, and output correction are typically studied as separate problems, and their relationship is rarely considered within a single decision process. Moreover, existing self-judgment and self-correction mechanisms are not designed around EEG inputs, neurophysiological evidence, or layer-wise decision trajectories. This leaves open how spatial evidence and cross-layer logit patterns can inform assessment and correction, with training-time feedback supporting the representations from which these cues are obtained.

B Baselines and Metrics

B.1 Baselines

B.1.1 EEG Foundation Model Backbones

We benchmark GACE on three representative EEG foundation model backbones:

- **CodeBrain** (Ma et al., 2026): a two-stage EEG foundation model that combines a decoupled neural tokenizer with multi-scale sequence modeling to capture temporal and spectral EEG patterns.
- **CBraMod** (Wang et al., 2025a): a criss-cross Transformer-based EEG foundation model that separately models spatial and temporal dependencies in EEG patches.
- **CSBrain** (Zhou et al., 2025c): a cross-scale spatiotemporal EEG foundation model designed to capture multi-scale neural structures for generalized EEG decoding.

B.1.2 Baseline Construction

For each dataset and backbone, the baseline is the original pretrained EFM fine-tuned on the downstream training split. The baseline and GACE use the same preprocessing, data split, random seed, and evaluation protocol. GACE starts from the same pretrained checkpoint and keeps the backbone architecture unchanged; it only adds training-time Self-Guidance, post-hoc Self-Assessment, and conservative Self-Correction. Thus, the reported differences isolate the effect of the proposed reliability-oriented procedure rather than changes in backbone capacity or data usage.

Online training baseline. Active Bias SGD-WPV maintains each sample's history of true-class probabilities and uses its variability to form an uncertainty weight on cross-entropy. The comparison in Appendix H.1 reports test B-ACC across seeds 0, 1, and 2 for Self-Guidance and Active Bias on the corresponding task splits. Active Bias uses a five-epoch burn-in before applying its weights; checkpoints are selected using the calibration criterion of each training workflow.

Additional risk and correction baselines. TrustScore uses the frozen backbone's 200-dimensional calibration embeddings, removes the least dense 10% of each class by tenth-neighbor radius, and compares second-neighbor distances to the predicted and nearest alternative classes. The negative score ranks errors. Mahalanobis risk uses calibration class means and a pooled full covariance with a fixed diagonal ridge of 0.001 times the mean diagonal; the minimum squared class distance ranks errors. A separate nearest-Mahalanobis-class correction rule changes only risk-selected predictions. Implementation and task-wise outcomes appear in Appendix H.1; the single-layer Mahalanobis adaptation is not a full multilayer out-of-distribution detector.

B.2 Metrics

Following recent EEG foundation model benchmarks, we report three complementary metrics for all downstream classification tasks: Balanced Accuracy (B-ACC), weighted F1-score (F1), and Cohen's Kappa (Kappa). All metrics are computed on the designated test split, and larger values indicate better performance.

- **Balanced Accuracy (B-ACC):** This metric averages class-wise recall and is robust to class imbalance. For K classes, it is computed as

$$\text{B-ACC} = \frac{1}{K}\sum_{k=1}^{K}\frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k}, \tag{15}$$

  where TP_k and FN_k denote the true positives and false negatives of class k, respectively.
- **Weighted F1-score (F1):** Weighted F1 summarizes the class-wise F1 scores using the empirical class support as weights:

$$\text{F1} = \sum_{k=1}^{K}\frac{n_k}{N}\frac{2\,\text{Precision}_k\,\text{Recall}_k}{\text{Precision}_k + \text{Recall}_k}, \tag{16}$$

where n_k is the number of test samples in class k, N is the total number of test samples, $\text{Precision}_k = \text{TP}_k/(\text{TP}_k + \text{FP}_k)$, and $\text{Recall}_k = \text{TP}_k/(\text{TP}_k + \text{FN}_k)$.

- **Cohen's Kappa (Kappa):** Cohen's Kappa measures prediction agreement with the ground truth after correcting for chance agreement:

$$\text{Kappa} = \frac{p_o - p_e}{1 - p_e}, \tag{17}$$

where p_o is the observed agreement and p_e is the expected agreement estimated from the marginal label distributions of the confusion matrix.

Reported multi-seed summaries use seeds $0, 1, 2$. For each dataset and backbone, the baseline and GACE are evaluated under matched splits and the same three seeds. Means and sample standard deviations are computed over these three runs (standard-deviation denominator $3 - 1$); when a table omits the standard deviation, its mean retains the same three-seed scope. Explicitly identified single-run or selected-sample diagnostics retain their stated scope. The improvement Δ in the main tables is computed by subtracting the corresponding fine-tuned backbone baseline from GACE under the same dataset, backbone, and metric:

$$\Delta_{\text{metric}} = \text{metric}(\text{GACE}) - \text{metric}(\text{Baseline}). \tag{18}$$

Dataset descriptions are provided in Appendix C, and the complete computational procedure of GACE is summarized in Appendix D.

C Dataset and Experimental Settings

Table 4 summarizes the downstream EEG datasets used in the main benchmark and supplementary evaluation. The selection spans heterogeneous task families, so the comparison is not concentrated on a single EEG decoding scenario.

Table 4: Summary of downstream EEG tasks and datasets.

EEG Downstream Task	Dataset	#Samples	Train	Val	Test	Shape	Labels
Motor imagery	BCI-IV-2a	5,088	1–5	6–7	8–9	22 channels × 4s	4-class
	Cho2017-MI	9,880	1–31, 33–35	36–42	43–45, 47–48, 50–52	64 channels × 3s	2-class
Emotion recognition	SEED	4,050	1–10	11–12	13–15	62 channels × 4s	2-class
	DEAP	7,680	1–22	23–26	27–32	32 channels ×10s	2-class
	SEED-V	117,744	1–5	6–10	11–15	60 channels × 1s	5-class
Sleep staging	ISRUC-S3	8,800	1–8	9	10	6 channels × 30s	5-class
Speech decoding	Speech-BCIC2020-3	6,000	1–60	61–70	71–80	64 channels × 3s	5-class
MDD classification	Mumtaz2016	7,143	1–43	44–52	53–62	19 channels × 5s	2-class
Attention recognition	ATTENTION	4,680	1–16	17–21	22–26	28 channels × 4s	2-class

C.1 Detailed Dataset Descriptions

BCI-IV-2a. BCI-IV-2a is a motor imagery benchmark containing EEG recordings from nine subjects performing four imagined movement classes: left hand, right hand, both feet, and tongue (Brunner et al., 2008). Following the common subject-disjoint protocol, the data are organized into 4-second windows with 22 EEG channels. Subjects 1–5, 6–7, and 8–9 are used for training, validation, and testing, respectively.

Cho2017-MI. Cho2017-MI is used as an additional motor imagery benchmark to evaluate whether the method generalizes beyond BCI-IV-2a (Cho et al., 2017). We use 64-channel EEG segments of 3 seconds and formulate the task as binary left- versus right-hand motor imagery classification. Subjects 1–31 and 33–35 are assigned to training, 36–42 to validation, and 43–45, 47–48, and 50–52 to testing (34/7/8 subjects, respectively).

SEED. SEED is an EEG emotion recognition dataset collected under affective video stimulation and is originally organized for three-class emotion classification (Zheng & Lu, 2015). Following the

binary emotion-recognition protocol used in ECHO (Liu et al., 2026a), we map the labels to two classes. We use 62-channel EEG segments of 4 seconds and assign subjects 1–10, 11–12, and 13–15 to training, validation, and testing, respectively.

DEAP. DEAP is a multimodal affective computing dataset containing EEG and peripheral physiological recordings elicited by music-video stimuli (Koelstra et al., 2012). We use a subject-wise split: s01–s22 for training, s23–s26 for validation, and s27–s32 for testing, yielding 5,280/960/1,440 samples for train/val/test. Each sample has shape 32×2500 (10s at 250Hz). Labels are binary (0/1) based on valence thresholding.

ISRUC-S3. ISRUC-S3 is a sleep staging subset from the ISRUC-Sleep dataset (Khalighi et al., 2016). It contains healthy-subject sleep recordings annotated according to standard sleep stages. We use six EEG derivations and segment the recordings into 30-second epochs for five-class sleep stage classification. Subjects 1–8, 9, and 10 are assigned to training, validation, and testing, respectively.

Speech-BCIC2020-3. Speech-BCIC2020-3 is an imagined speech dataset released for the 2020 International BCI Competition (Jeong et al., 2022). It contains 64-channel EEG recordings from subjects imagining the pronunciation of five words or phrases. Each trial is treated as a 3-second EEG segment, and the split follows the competition-style trial partition: trials 1–60 for training, 61–70 for validation, and 71–80 for testing.

Mumtaz2016. Mumtaz2016 is a major depressive disorder classification dataset containing EEG recordings from MDD patients and healthy controls (Mumtaz, 2016). We use 19-channel EEG segments of 5 seconds and formulate the task as binary classification. The subject-disjoint split assigns subjects 1–43 to training, 44–52 to validation, and 53–62 to testing.

ATTENTION. ATTENTION is derived from a simultaneous EEG-NIRS study involving a discrimination/selection response paradigm (Shin et al., 2018). The EEG task is formulated as binary recognition of rest versus attention-task segments (labels 0 and 1, respectively). We use 28 EEG channels, 4-second segments, and a subject-disjoint split with subjects 1–16 for training, 17–21 for validation, and 22–26 for testing.

D Implementation Procedure of GACE

GACE keeps the EFM backbone unchanged and executes three sequential modules around its internal decision structure: Self-Guidance for downstream fine-tuning, Self-Assessment for post-hoc risk estimation, and Self-Correction for conservative revision of high-risk predictions. The full execution flow is summarized in Algorithm 1.

D.1 Self-Guidance

Fine-tuning phase. Self-Guidance is applied during downstream fine-tuning on the training set. For each mini-batch sample, the EFM produces logits, from which GACE maintains online summaries of true-class margin and confidence fluctuation. These statistics describe the model's evolving response to that labeled sample and provide feedback for adjusting its relative contribution to the fine-tuning loss. This training step aims to support task-relevant representation learning and provide a basis for clearer Decision Evidence and Decision Trajectory. After this phase, we obtain a self-guided EFM and freeze it as the backbone for the subsequent post-hoc modules.

D.2 Self-Assessment

Risk-head training phase. The Risk Head is trained on the calibration set. The frozen EFM first predicts each calibration sample and extracts the post-hoc decision signals used for risk modeling, including the predicted class probabilities, final logits, and Decision Evidence. Since calibration samples have ground-truth labels, each prediction can be marked as correct or wrong. These correctness labels supervise the Risk Head to estimate whether a prediction is likely to be an error from its post-hoc decision signals. Internal K-fold validation on the calibration set selects its checkpoint epoch and provides out-of-fold risk scores for calibration-sample selection.

Algorithm 1 Guidance, Assessment, and Correction for EEG Foundation Models

Input: Training set $\mathcal{D}_{\mathrm{tr}}$, calibration set $\mathcal{D}_{\mathrm{cal}}$, test set $\mathcal{D}_{\mathrm{te}}$, pretrained EFM f_θ, training-response extractors $\Psi_{\mathrm{mar}}, \Psi_{\mathrm{cf}}$, Risk Head g, Correction Head h, risk-selection budget q.
Output: Self-guided EFM $f_{\theta^\star}$, risk scores $r(\mathbf{X})$, and final predictions $\tilde{y}$ on $\mathcal{D}_{\mathrm{te}}$.

Stage 0: Data Split & Backbone Initialization

1. Partition downstream data into $\mathcal{D}_{\mathrm{tr}}$, $\mathcal{D}_{\mathrm{cal}}$, and $\mathcal{D}_{\mathrm{te}}$.
2. Initialize the downstream model with pretrained EFM weights θ.

Stage 1: Self-Guidance during Fine-tuning (Section 3.1)

repeat
    1. Sample mini-batch $\{(\mathbf{X}, y)\}$ from $\mathcal{D}_{\mathrm{tr}}$.
    2. Compute training-time logits $\boldsymbol{\ell}_t(\mathbf{X})$.
    3. Update the online training statistics $\mathbf{D}(\mathbf{X}; \Psi_{\mathrm{mar}})$ and $\mathbf{D}(\mathbf{X}; \Psi_{\mathrm{cf}})$ (Eq. (5) and Eq. (7)).
    4. Convert these statistics into sample weights $w_{\mathrm{mar}}(\mathbf{X})$ and $w_{\mathrm{cf}}(\mathbf{X})$ (Eq. (8)).
    5. Update θ using the self-guided fine-tuning objective (Eq. (9)).
until convergence or max epochs reached
6. Freeze the resulting self-guided EFM as $f_{\theta^\star}$.

Stage 2: Self-Assessment with Risk Modeling (Section 3.2)

for each calibration sample $(\mathbf{X}, y) \in \mathcal{D}_{\mathrm{cal}}$ **do**
    1. Compute class probabilities $\mathbf{R}(\mathbf{X})$ and Decision Evidence $\mathbf{E}(\mathbf{X})$ with $f_{\theta^\star}$ (Eq. (3)).
    2. Construct risk pair $(\mathbf{v}^{\mathrm{risk}}(\mathbf{X}), \tau^{\mathrm{risk}}(\mathbf{X}))$, where τ^{risk} marks whether $\hat{y} \neq y$ (Eq. (11)).
end for
3. Select the checkpoint epoch by internal K-fold validation on $\mathcal{D}_{\mathrm{cal}}$, then train Risk Head g on $\mathcal{D}_{\mathrm{cal}}$ with the risk optimization objective (Eq. (53) – (54)).
4. Obtain out-of-fold risk scores for $\mathcal{D}_{\mathrm{cal}}$; apply the full-calibration g to $\mathcal{D}_{\mathrm{te}}$.
5. Select the highest-risk 12% within each split to form $\mathcal{H}_{\mathrm{cal}}$ and $\mathcal{H}_{\mathrm{te}}$ (Eq. (10)).

Stage 3: Self-Correction for High-risk Predictions (Section 3.3)

for each calibration sample $(\mathbf{X}, y) \in \mathcal{H}_{\mathrm{cal}}$ **do**
    1. Extract Decision Trajectory $\mathbf{T}(\mathbf{X})$ from layer-wise logits (Eq. (4)).
    2. Construct correction input $\mathbf{v}^{\mathrm{corr}}(\mathbf{X})$ and target $\tau^{\mathrm{corr}}(\mathbf{X})$ (Eq. (12) and Eq. (13)).
end for
3. Select the checkpoint epoch by internal K-fold validation on $\mathcal{D}_{\mathrm{cal}}$, then train Correction Head h on $\mathcal{H}_{\mathrm{cal}}$ with the correction objective (Eq. (63)).
for each high-risk test sample $\mathbf{X} \in \mathcal{H}_{\mathrm{te}}$ **do**
    4. Score candidate actions $\mathcal{A} = \{\mathrm{keep}, 1, \ldots, K\}$ with h.
    5. Change the original prediction only if the candidate differs from it and both probability gates pass (Eq. (65)).
end for

6. **return** Final predictions $\tilde{y}$, keeping all low-risk samples unchanged.

Risk scoring phase. The trained Risk Head is then applied to the test set. Each test sample is processed by the frozen EFM to obtain the original prediction and the same decision signals used during risk-head training. The Risk Head assigns a risk score to each prediction, and predictions with high scores are selected as high-risk samples. Low-risk samples keep the original EFM prediction, while high-risk samples are passed to Self-Correction.

D.3 Self-Correction

Correction-head training phase. The Correction Head is trained on the highest-risk 12% of calibration samples selected by Self-Assessment from the out-of-fold risk scores. Its checkpoint epoch is selected by internal K-fold validation on the calibration set. For the selected samples, GACE further extracts the Decision Trajectory, which summarizes class-wise alignment between intermediate and final logits. The calibration labels define the correction target: if the original prediction is correct, the target action is keep; if the original prediction is wrong, the target action is the ground-truth class. The Correction Head therefore learns how to choose between preserving the original prediction and revising it using the learned trajectory and decision-evidence features.

Conservative correction phase. Only high-risk test samples selected by Self-Assessment enter this phase. For each such sample, the Correction Head scores a set of candidate actions, including keep and all possible class labels. The final prediction changes only if the most probable action among the class-label actions differs from the original prediction, the summed probability of the class-label actions passes a threshold, and that action's probability exceeds the original-class action probability by a specified margin. Otherwise, the original predicted label is preserved. The keep probability determines this class-action mass through $1 - \pi_{\mathrm{keep}}$; the candidate is not required to exceed the keep probability individually. In this way, Self-Correction does not reclassify all samples, but only performs limited revision when the estimated risk and learned correction policy pass their respective selection and change rules.

E Computation of Decision Signals

E.1 Computation of Decision Evidence

Decision Evidence is computed as a channel-level Integrated Gradients attribution of the maximum-logit response. Given an input $\mathbf{X} \in \mathbb{R}^{C\times T}$, where C is the number of EEG channels and T is the number of sampling points, the EFM first produces logits $\boldsymbol{\ell}(\mathbf{X}) \in \mathbb{R}^K$ over K candidate classes and the class probabilities $\mathbf{R}(\mathbf{X}) = \mathrm{Softmax}(\boldsymbol{\ell}(\mathbf{X}))$. The final predicted label is $\hat{y} = \arg\max_k [\mathbf{R}(\mathbf{X})]_k$. For attribution, we evaluate $g(\mathbf{U}) = \max_k \ell_k(\mathbf{U})$ along the interpolation path; its maximizing class need not remain $\hat{y}$.

Integrated-gradient path. We use the zero baseline $\mathbf{X}_0 = \mathbf{0}$ and construct M uniformly spaced interpolation points from the baseline to the real input, where M is the number of integration steps:

$$\alpha_m = \frac{m}{M}, \qquad \mathbf{X}^{(m)} = \mathbf{X}_0 + \alpha_m(\mathbf{X} - \mathbf{X}_0), \qquad m = 1, \ldots, M. \tag{19}$$

At each path point, let $k_m = \arg\max_j \ell_j(\mathbf{X}^{(m)})$. We use $M = 16$ right-endpoint steps in the experiments. For each channel-time coordinate (c, t), the Integrated Gradients attribution is approximated by

$$\mathrm{IG}_{c,t}(\mathbf{X}) = (X_{c,t} - X_{0,c,t}) \frac{1}{M} \sum_{m=1}^{M} \frac{\partial \ell_{k_m}(\mathbf{X}^{(m)})}{\partial X_{c,t}^{(m)}}. \tag{20}$$

The gradient therefore follows the largest logit at each interpolation point, matching the operator used for both Decision Evidence features and the spatial analyses in Appendix K.1. Ground-truth labels are not used as attribution targets.

Channel aggregation. We use absolute attribution magnitude to avoid cancellation between positive and negative signs:

$$A_{c,t}(\mathbf{X}) = \left|\mathrm{IG}_{c,t}(\mathbf{X})\right|. \tag{21}$$

The raw channel evidence score is obtained by summing the absolute attribution over time:

$$a_c(\mathbf{X}) = \sum_{t=1}^{T} A_{c,t}(\mathbf{X}), \qquad c = 1, \ldots, C. \tag{22}$$

Equivalently, when the backbone represents the temporal axis as P patches with within-patch length D ($T = PD$), the same computation is applied to coordinates (c, p, d), and the channel score is

$$a_c(\mathbf{X}) = \sum_{p=1}^{P} \sum_{d=1}^{D} \left| \mathrm{IG}_{c,p,d}(\mathbf{X}) \right|. \tag{23}$$

where $p = 1, \ldots, P$ indexes temporal patches and $d = 1, \ldots, D$ indexes positions within each patch. This gives the raw attribution vector

$$\mathbf{a}(\mathbf{X}) = [a_1(\mathbf{X}), \ldots, a_C(\mathbf{X})]^\top, \qquad a_c(\mathbf{X}) \geq 0. \tag{24}$$

Evidence normalization. Finally, the raw channel scores are normalized into the Decision Evidence distribution used in Eq. (3), with $\epsilon > 0$ denoting a small numerical constant:

$$\bar{a}_c(\mathbf{X}) = \frac{a_c(\mathbf{X})}{\sum_{j=1}^{C} a_j(\mathbf{X}) + \epsilon}, \qquad \mathbf{E}(\mathbf{X}) = [\bar{a}_1(\mathbf{X}), \ldots, \bar{a}_C(\mathbf{X})]^\top. \tag{25}$$

Thus, the complete computation maps $\mathbf{X}$ to a maximum-logit-path attribution map, aggregates its absolute values into the raw channel vector $\mathbf{a}(\mathbf{X})$, and normalizes it into $\mathbf{E}(\mathbf{X})$. The resulting vector describes relative spatial attribution magnitude rather than signed support for a single class or a new class prediction.

E.2 Computation of Decision Trajectory

Decision Trajectory is computed from the logit-evolution pattern across EFM layers. Given an input $\mathbf{X}$, the EFM produces the final logits $\boldsymbol{\ell}(\mathbf{X}) \in \mathbb{R}^K$ and the layer-wise representations $\mathbf{Z}^{(l)}(\mathbf{X})$, where $l = 1, \ldots, L$, L is the number of encoder layers, and K is the number of candidate classes. We first map each layer representation to the same logit space using the shared classification head:

$$\boldsymbol{\ell}^{(l)}(\mathbf{X}) = \mathcal{H}(\mathbf{Z}^{(l)}(\mathbf{X}); \phi), \qquad l = 1, \ldots, L. \tag{26}$$

Here, $\mathcal{H}(\cdot; \phi)$ is the classification head with parameters ϕ. If the final layer is included among the cached trajectory layers, its self-comparison to the final logits has zero alignment and therefore receives zero trajectory weight.

Layer-wise probability and evolution direction. For each layer, we compute a temperature-scaled probability vector

$$\mathbf{p}^{(l)}(\mathbf{X}) = \mathrm{Softmax}\left(\frac{\boldsymbol{\ell}^{(l)}(\mathbf{X})}{\rho} \right), \qquad \mathbf{p}^{(l)}(\mathbf{X}) \in [0, 1]^K, \tag{27}$$

where $\rho > 0$ is the evolution temperature. For class c, let $\mathbf{e}_c \in \mathbb{R}^K$ denote the one-hot vector of that class. The class-direction vector at layer l is

$$\mathbf{v}_{l,c}(\mathbf{X}) = \mathbf{p}^{(l)}(\mathbf{X}) - \mathbf{e}_c, \qquad c = 1, \ldots, K. \tag{28}$$

We define the intermediate-minus-final logit difference as

$$\boldsymbol{\delta}_l(\mathbf{X}) = \boldsymbol{\ell}^{(l)}(\mathbf{X}) - \boldsymbol{\ell}(\mathbf{X}). \tag{29}$$

Here, $\mathbf{v}_{l,c}$ is formed from the layer-wise probability vector and the one-hot vector for class c, while $\boldsymbol{\delta}_l$ points from the final logits toward the layer-l logits.

Alignment-based trajectory score. For each layer-class pair (l, c), we compute a non-negative alignment score:

$$u_{l,c}(\mathbf{X}) = \left[\max\left(\frac{\langle \boldsymbol{\delta}_l(\mathbf{X}), \mathbf{v}_{l,c}(\mathbf{X})\rangle}{\|\boldsymbol{\delta}_l(\mathbf{X})\|_2 \|\mathbf{v}_{l,c}(\mathbf{X})\|_2 + \epsilon}, 0\right)\right]^2, \tag{30}$$

where $\langle \cdot, \cdot \rangle$ is the inner product, $\| \cdot \|_2$ is the Euclidean norm, and $\epsilon > 0$ is a numerical constant. The squared positive cosine score retains positive alignment between each class-indexed vector and the intermediate-minus-final logit difference. It is an alignment statistic, not a direct measure of a class preference lost before the final layer.

The normalized layer-wise trajectory score is obtained from the alignment scores within each layer:

$$p^{(l)}_{\log}(c \mid \mathbf{X}) = \frac{u_{l,c}(\mathbf{X})}{\sum_{j=1}^{K} u_{l,j}(\mathbf{X}) + \epsilon}. \tag{31}$$

We then compute the total alignment mass of layer l,

$$q_l(\mathbf{X}) = \sum_{c=1}^{K} u_{l,c}(\mathbf{X}), \tag{32}$$

and convert it into the layer aggregation weight

$$w_l(\mathbf{X}) = \frac{q_l(\mathbf{X})}{\sum_{r=1}^{L} q_r(\mathbf{X}) + \epsilon}, \qquad l = 1, \ldots, L. \tag{33}$$

Thus, a layer receives a larger weight when its intermediate-minus-final logit difference has greater total positive alignment across classes.

Trajectory aggregation. The final Decision Trajectory aggregates the layer-wise trajectory scores into a class-wise alignment vector:

$$p_{\log}(c \mid \mathbf{X}) = \sum_{l=1}^{L} w_l(\mathbf{X}) p^{(l)}_{\log}(c \mid \mathbf{X}), \qquad \mathbf{T}(\mathbf{X}) = [p_{\log}(1 \mid \mathbf{X}), \ldots, p_{\log}(K \mid \mathbf{X})]^\top. \tag{34}$$

Here, $p_{\log}(c \mid \mathbf{X})$ is an aggregated alignment score for class c, and $\mathbf{T}(\mathbf{X}) \in \mathbb{R}^K$ is the cross-layer alignment vector. The scalar $1 - \max_k T_k(\mathbf{X})$ summarizes how dispersed these scores are. Thus, $\mathbf{T}(\mathbf{X})$ encodes intermediate-versus-final logit differences as a correction-head feature; its largest component need not be the true class.

E.3 Online Training Statistics for Self-Guidance

Self-Guidance computes sample-wise training feedback during downstream fine-tuning. For a labeled training sample $(\mathbf{X}, y)$, let t denote a training stage at which the sample is observed, and let $\boldsymbol{\ell}_t(\mathbf{X}) \in \mathbb{R}^K$ be the logits produced by the current EFM over K candidate classes. Given a training-response extractor $\Psi(\cdot)$, we maintain an exponential moving state for this sample:

$$\mathbf{D}_t(\mathbf{X}; \Psi) = \beta \mathbf{D}_{t-1}(\mathbf{X}; \Psi) + (1 - \beta)\Psi(\boldsymbol{\ell}_t(\mathbf{X})), \qquad \beta \in [0, 1), \tag{35}$$

where β is the EMA decay coefficient. When the sample is first observed, we initialize $\mathbf{D}_0(\mathbf{X}; \Psi) = \Psi(\boldsymbol{\ell}_0(\mathbf{X}))$. After the latest observed training stage τ, the online statistic used in the main text is $\mathbf{D}(\mathbf{X}; \Psi) = \mathbf{D}_\tau(\mathbf{X}; \Psi)$, which instantiates the $\mathrm{EMA}_{t \le \tau}[\cdot]$ notation in Eq. (5).

Training-response extractors. For self-guidance, we use the two extractors defined in Section 3.1. The first extractor measures true-class margin support:

$$\Psi_{\mathrm{mar}}(\boldsymbol{\ell}_t(\mathbf{X})) = \ell_{t,y}(\mathbf{X}) - \max_{k \neq y} \ell_{t,k}(\mathbf{X}), \tag{36}$$

where $\ell_{t,k}(\mathbf{X})$ is the logit of class k, and y is the ground-truth class of $\mathbf{X}$. The second extractor measures fluctuation of true-class confidence:

$$\Psi_{\mathrm{cf}}(\boldsymbol{\ell}_t(\mathbf{X})) = \big(p_{t,y}(\mathbf{X}) - \mu_{t,y}(\mathbf{X})\big)^2, \qquad p_{t,y}(\mathbf{X}) = \mathrm{Softmax}(\boldsymbol{\ell}_t(\mathbf{X}))_y. \tag{37}$$

Here, $p_{t,y}(\mathbf{X})$ is the model probability assigned to the true class at stage t, and $\mu_{t,y}(\mathbf{X})$ is its running mean. Using the same EMA convention as Eq. (35), this mean is updated by

$$\mu_{t,y}(\mathbf{X}) = \beta \mu_{t-1,y}(\mathbf{X}) + (1 - \beta) p_{t,y}(\mathbf{X}). \tag{38}$$

Thus, Ψ_{mar} captures whether the true class is separated from its strongest competitor, while Ψ_{cf} captures whether the true-class confidence is stable over training.

Robust normalization. Before converting the training statistics into weights, we normalize their values across a normalization pool $\mathcal{P}$, such as the current mini-batch or a cached subset of training samples. For a scalar training statistic u_i from sample $i \in \mathcal{P}$, the robust normalization operator used in Eq. (8) is

$$\mathrm{RNorm}(u_i) = \frac{u_i - \mathrm{Med}(\{u_j\})}{\mathrm{MAD}(\{u_j\}) + \epsilon}, \qquad \mathrm{MAD}(\{u_j\}) = \mathrm{Med}\left(\{|u_j - \mathrm{Med}(\{u_j\})|\}\right). \tag{39}$$

Here, $\{u_j\}_{j\in\mathcal{P}}$ denotes the training-statistic values in the normalization pool, Med is the median, MAD is the median absolute deviation, and $\epsilon > 0$ is a numerical constant.

Feedback-based sample weights. The normalized training statistics are converted into two multiplicative online weights:

$$\begin{aligned} w_{\mathrm{mar}}(\mathbf{X}) &= (1 - \lambda_{\mathrm{mar}}) + \lambda_{\mathrm{mar}}\sigma\left(\mathrm{RNorm}\left(\mathbf{D}(\mathbf{X}; \Psi_{\mathrm{mar}})\right)\right), \\ w_{\mathrm{cf}}(\mathbf{X}) &= \exp\left(-\lambda_{\mathrm{cf}} \max\left(\mathrm{RNorm}\left(\mathbf{D}(\mathbf{X}; \Psi_{\mathrm{cf}})\right), 0\right)\right), \end{aligned} \tag{40}$$

where $\sigma(\cdot)$ is the sigmoid function, and λ_{mar} and λ_{cf} control the strength of relative margin-based emphasis and confidence-fluctuation suppression, respectively. Finally, these weights are injected into the self-guided objective:

$$\mathcal{L}_{\mathrm{sg}} = \frac{1}{|B|} \sum_{(\mathbf{X},y)\in B} w_{\mathrm{mar}}(\mathbf{X}) w_{\mathrm{cf}}(\mathbf{X}) \cdot \ell_{\mathrm{ce}}(\mathbf{X}, y), \tag{41}$$

where B is the mini-batch and ℓ_{ce} is the standard cross-entropy loss. This gives the complete computation from training-time logits to online feedback statistics and then to sample-level self-guidance weights.

F Self-Assessment and Self-Correction Details

F.1 Risk Modeling Details for Self-Assessment

Proof of Lemma 3.2. For each test sample $\mathbf{X}_i$, let

$$\tau_i^{\mathrm{risk}} = \mathbb{I}[\hat{y}_i \neq y_i] \tag{42}$$

denote whether the current predicted label is incorrect, and let

$$\eta_i = P(\tau_i^{\mathrm{risk}} = 1 \mid \mathbf{v}^{\mathrm{risk}}(\mathbf{X}_i)) \tag{43}$$

be its conditional error probability given the post-hoc risk input. For any selected subset S with $|S| = m$, the expected number of captured errors conditioned on the risk inputs is

$$\mathbb{E}\left[\sum_{i\in S} \tau_i^{\mathrm{risk}} \mid \{\mathbf{v}^{\mathrm{risk}}(\mathbf{X}_i)\}_{i=1}^N\right] = \sum_{i\in S} \eta_i. \tag{44}$$

If S is not the top-m set under η_i, then there exist $i \in S$ and $j \notin S$ such that $\eta_j > \eta_i$. Replacing i with j keeps the budget fixed and increases Eq. (44) by $\eta_j - \eta_i > 0$. Repeating this exchange argument yields the set of the m largest conditional risks, proving Eq. (10).

The BCE risk head is consistent with the same statistic. For a fixed $\mathbf{v}^{\mathrm{risk}}$, let $\eta = P(\tau^{\mathrm{risk}} = 1 \mid \mathbf{v}^{\mathrm{risk}})$. The conditional binary cross-entropy for prediction $r \in (0, 1)$ is

$$\mathcal{R}(r \mid \mathbf{v}^{\mathrm{risk}}) = -\eta \log r - (1 - \eta) \log(1 - r). \tag{45}$$

Setting its derivative to zero gives

$$-\frac{\eta}{r} + \frac{1 - \eta}{1 - r} = 0 \quad \Rightarrow \quad r = \eta. \tag{46}$$

Thus, with sufficient calibration data and model capacity, minimizing BCE estimates the conditional error probability, and ranking by this estimate is optimal for fixed-budget error selection. □

Post-hoc Risk Pair Construction. Self-assessment is trained on a calibration split $\mathcal{D}_{\text{cal}}$. For a calibration sample $(\mathbf{X}, y)$, the frozen self-guided EFM outputs Decision Evidence $\mathbf{E}(\mathbf{X})$, final logits $\boldsymbol{\ell}(\mathbf{X})$, class probabilities $\mathbf{R}(\mathbf{X})$, and predicted label $\hat{y} = \arg\max_k R_k(\mathbf{X})$. We convert this sample into the post-hoc risk pair used in Eq. (11):

$$\left(\mathbf{v}^{\text{risk}}(\mathbf{X}), \tau^{\text{risk}}(\mathbf{X})\right) = \left([\mathbf{E}(\mathbf{X}); \boldsymbol{\ell}(\mathbf{X}); \Gamma_{\text{R}}(\mathbf{R}(\mathbf{X}))], \mathbb{I}[\hat{y} \neq y]\right). \tag{47}$$

Here, $\mathbf{v}^{\text{risk}}$ is the input to the risk head, and $\tau^{\text{risk}} \in \{0, 1\}$ is the supervised target indicating whether the original EFM prediction is wrong. The three input blocks have complementary roles:

- **Decision Evidence** $\mathbf{E}(\mathbf{X})$ describes the channel-level attribution associated with the decision, capturing the concentration and location of the model's spatial reliance.
- **Logits** $\boldsymbol{\ell}(\mathbf{X})$ retain the pre-softmax class responses, preserving decision-scale information that may be compressed by posterior probabilities.
- **Output-probability statistics** $\Gamma_{\text{R}}(\mathbf{R}(\mathbf{X}))$ summarize the posterior confidence and uncertainty of the final class probability distribution.

The operator Γ_{R} is instantiated with three statistics: peak probability, probability margin, and normalized entropy. Let

$$p_{\max}(\mathbf{X}) = \max_k R_k(\mathbf{X}), \qquad p_{(2)}(\mathbf{X}) = \text{the second largest entry of } \mathbf{R}(\mathbf{X}), \tag{48}$$

where $p_{\max}$ is the peak probability and $p_{(2)}$ is the strongest competing posterior probability. The probability margin and normalized entropy are

$$m_{\text{prob}}(\mathbf{X}) = p_{\max}(\mathbf{X}) - p_{(2)}(\mathbf{X}), \qquad \widetilde{H}(\mathbf{R}(\mathbf{X})) = \frac{-\sum_{k=1}^{K} R_k(\mathbf{X}) \log R_k(\mathbf{X})}{\log K}. \tag{49}$$

Then the summary vector is

$$\Gamma_{\text{R}}(\mathbf{R}(\mathbf{X})) = [p_{\max}(\mathbf{X}); m_{\text{prob}}(\mathbf{X}); \widetilde{H}(\mathbf{R}(\mathbf{X}))]. \tag{50}$$

Peak probability measures top-class confidence; probability margin measures separation from the nearest competing class; normalized entropy measures posterior dispersion over all K classes. Smaller margin or higher entropy indicates a less decisive prediction.

The risk label can also be written as a deterministic function of the predicted class probabilities and the ground-truth label:

$$\tau^{\text{risk}}(\mathbf{X}) = \tau^{\text{risk}}(\mathbf{R}(\mathbf{X}), y) = \mathbb{I}[\arg\max_k R_k(\mathbf{X}) \neq y]. \tag{51}$$

Risk Head and Optimization. Let $\mathbf{z}$ be the MLP representation of the standardized risk input. The head produces a learned residual $g_r(\mathbf{z})$ and auxiliary class logits $\mathbf{a} = g_a(\mathbf{z})$. With $c = 1 - p_{\max}(\mathbf{X})$, its risk score is

$$\begin{aligned} s_{\text{res}}(\mathbf{X}) &= \text{softplus}(\alpha)\,\text{logit}(\text{clip}(c)) + \beta + \kappa \tanh(g_r(\mathbf{z})), \\ s(\mathbf{X}) &= (1 - \lambda_{\text{mix}}) s_{\text{res}}(\mathbf{X}) + \lambda_{\text{mix}}\,\text{logit}\left(1 - \text{Softmax}(\mathbf{a})_{\hat{y}}\right), \qquad r(\mathbf{X}) = \sigma(s(\mathbf{X})). \end{aligned} \tag{52}$$

Here α and β are learned, κ bounds the residual, and λ_{mix} controls the auxiliary class-support contribution. The fitting term for the original prediction's error indicator is binary cross-entropy:

$$\mathcal{L}_{\text{bce}} = -\tau^{\text{risk}}(\mathbf{X}) \log r(\mathbf{X}) - \left(1 - \tau^{\text{risk}}(\mathbf{X})\right) \log\left(1 - r(\mathbf{X})\right). \tag{53}$$

The reported training objective is

$$\mathcal{L}_{\text{risk}} = \mathbb{E}_{(\mathbf{X},y)\sim\mathcal{D}_{\text{cal}}}\left[\mathcal{L}_{\text{bce}}(\mathbf{X}) + \lambda_{\text{aux}}\,\text{CE}(\mathbf{a}, y)\right], \tag{54}$$

where $\lambda_{\text{aux}} = 0$ disables the auxiliary class loss. The selected mixing and auxiliary-loss coefficients are reported in Appendix G. At inference time, $r(\mathbf{X})$ is computed for each test sample and samples are sorted in descending risk. Under a fixed selection budget ρ, the high-risk set is

$$\mathcal{S}_\rho = \text{Top}_{\lceil \rho N \rceil}\left\{r(\mathbf{X}_i)\right\}_{i=1}^{N}, \tag{55}$$

which is then passed to Self-Correction, while predictions outside $\mathcal{S}_\rho$ are kept unchanged.

F.2 Correction Modeling Details for Self-Correction

Post-hoc Correction Pair Construction. Self-correction is trained on the highest-risk 12% calibration subset $\mathcal{S}_{\text{cal}}$ selected by Self-Assessment. For each sample $(\mathbf{X}, y) \in \mathcal{S}_{\text{cal}}$, the frozen EFM provides the same post-hoc signals used for risk modeling, together with the Decision Trajectory $\mathbf{T}(\mathbf{X})$. The correction pair in Eq. (13) is

$$\left(\mathbf{v}^{\text{corr}}(\mathbf{X}), \tau^{\text{corr}}(\mathbf{X})\right) = \left([\mathbf{E}(\mathbf{X}); \boldsymbol{\ell}(\mathbf{X}); \Gamma_{\text{R}}(\mathbf{R}(\mathbf{X})); \Gamma_{\text{T}}(\mathbf{T}(\mathbf{X}))], \tau^{\text{corr}}(\mathbf{X})\right). \tag{56}$$

The input $\mathbf{v}^{\text{corr}}$ contains four blocks:

- **Decision Evidence** $\mathbf{E}(\mathbf{X})$ provides the channel-level attribution pattern associated with the original decision.
- **Logits** $\boldsymbol{\ell}(\mathbf{X})$ provide the final pre-softmax class responses of the EFM.
- **Output-probability statistics** $\Gamma_{\text{R}}(\mathbf{R}(\mathbf{X}))$ reuse the peak, margin, and entropy statistics defined in Eq. (50), preserving posterior confidence and uncertainty.
- **Trajectory statistics** $\Gamma_{\text{T}}(\mathbf{T}(\mathbf{X}))$ summarize class-wise cross-layer alignment from the Decision Trajectory.

The trajectory summary is computed as

$$\Gamma_{\text{T}}(\mathbf{T}(\mathbf{X})) = [\mathbf{T}(\mathbf{X}); 1 - \max_k T_k(\mathbf{X}); \text{onehot}(\widetilde{y}_{\text{T}})], \tag{57}$$

where $\mathbf{T}(\mathbf{X}) \in [0,1]^K$ is the normalized trajectory-score vector, $1 - \max_k T_k(\mathbf{X})$ is a score-dispersion summary, and

$$\widetilde{y}_{\text{T}} = \arg\max_{k \neq \hat{y}} T_k(\mathbf{X}) \tag{58}$$

is the highest-scoring non-original class under this alignment statistic. The one-hot vector $\text{onehot}(\widetilde{y}_{\text{T}})$ marks this alternative as an input feature, while the supervised correction head decides whether any class change is warranted. Thus, Γ_{T} contains the full alignment vector, a dispersion summary, and an alternative-class identity cue.

The correction target is a deterministic keep-or-change action:

$$\tau^{\text{corr}}(\mathbf{X}) = \tau^{\text{corr}}(\mathbf{R}(\mathbf{X}), y) = \begin{cases} \text{keep}, & \arg\max_k R_k(\mathbf{X}) = y, \\ y, & \arg\max_k R_k(\mathbf{X}) \neq y. \end{cases} \tag{59}$$

The action space is $\mathcal{A} = \{\text{keep}, 1, \ldots, K\}$. If the original prediction is correct, the target action is keep; otherwise, the target action is the ground-truth class y. Therefore, self-correction learns both whether to change the original predicted label and which class to change it to.

Optimization. The correction head $h(\cdot)$ is trained only on $\mathcal{S}_{\text{cal}}$, while the EFM backbone remains fixed. Given the correction input, it produces $K+1$ action scores and action probabilities:

$$\mathbf{s}_{\text{corr}}(\mathbf{X}) = h(\mathbf{v}^{\text{corr}}(\mathbf{X})) \in \mathbb{R}^{K+1}, \qquad \boldsymbol{\pi}(\mathbf{X}) = \text{Softmax}(\mathbf{s}_{\text{corr}}(\mathbf{X})). \tag{60}$$

The action cross-entropy fits the keep-or-change target:

$$\mathcal{L}_{\text{act}} = -\log \pi_{\tau^{\text{corr}}(\mathbf{X})}(\mathbf{X}). \tag{61}$$

To implement the target-action ranking constraint in Eq. (14), we further require the target action to exceed every competing action by a margin:

$$\mathcal{L}^{\text{corr}}_{\text{rank}} = \sum_{a \in \mathcal{A},\, a \neq \tau^{\text{corr}}(\mathbf{X})} \left[m_{\text{corr}} - s_{\tau^{\text{corr}}(\mathbf{X})}(\mathbf{X}) + s_a(\mathbf{X})\right]_+. \tag{62}$$

This term makes the policy conservative in two directions: for originally correct samples, keep is pushed above all change actions; for originally wrong samples, the ground-truth correction is pushed above both keep and the other class changes. The full correction objective is

$$\mathcal{L}_{\text{corr}} = \mathbb{E}_{\mathbf{X} \in \mathcal{S}_{\text{cal}}} \left[\mathcal{L}_{\text{act}}(\mathbf{X}) + \lambda_{\text{corr}} \mathcal{L}^{\text{corr}}_{\text{rank}}(\mathbf{X})\right], \tag{63}$$

where λ_{corr} controls the strength of the ranking term.

At inference time, Self-Correction is applied only to the high-risk test subset $\mathcal{S}_\rho$. For each selected sample, the best change action is

$$k^*(\mathbf{X}) = \arg \max_{k \in \{1,\dots,K\}} \pi_k(\mathbf{X}). \tag{64}$$

A correction is accepted only if the candidate differs from the original predicted class $\hat{y}$ and both probability gates pass:

$$k^*(\mathbf{X}) \neq \hat{y}, \qquad 1 - \pi_{\text{keep}}(\mathbf{X}) \geq c_{\text{p}}, \qquad \pi_{k^*(\mathbf{X})}(\mathbf{X}) - \pi_{\hat{y}}(\mathbf{X}) \geq c_{\text{m}}. \tag{65}$$

Here c_{p} is the minimum summed probability of the class-label actions and c_{m} is the minimum candidate-versus-original-class probability margin. The class-action sum $1 - \pi_{\text{keep}}$ includes the original-class action, which does not change the predicted label by itself. The keep probability is not compared directly with π_{k^*}. If all three conditions hold, the final prediction is changed to $k^*(\mathbf{X})$; otherwise, the original predicted label is preserved. Samples outside $\mathcal{S}_\rho$ never enter the correction head and are kept unchanged.

G Hyperparameter Settings

Selected-run risk configurations. The reported post-hoc risk inputs summarize posterior confidence, predicted-class identity, and channel-attribution information. Features are constructed without ground-truth or error labels, standardized using calibration-split statistics, and clipped to the same finite range at inference.

In Table 5, λ_{mix} controls the mixture of a confidence-based residual risk score and an auxiliary class-support score, λ_{aux} weights the auxiliary class cross-entropy, and κ bounds the learned residual. Inference uses an exponential moving average of the trained parameters. The trained score is used for ranking and is not assumed calibrated.

Table 5: Selected Self-Assessment risk-head mixing and auxiliary-loss coefficients. Each cell gives $\lambda_{\text{mix}}/\lambda_{\text{aux}}$ for the selected run. The residual cap is $\kappa = 1$.

Dataset	CodeBrain	CSBrain	CBraMod
ATTENTION	0.20/0.85	0.10/0.25	0.50/0.50
BCI-IV-2a	1.00/1.00	1.00/1.00	0.50/0.50
Cho2017-MI	0.10/0.10	0.25/1.00	0.75/1.00
ISRUC-S3	0.00/0.10	1.00/0.50	0.10/0.00
Mumtaz2016	0.50/0.50	0.50/0.25	0.50/0.50
Speech-BCIC2020-3	0.50/0.50	0.50/0.50	0.50/0.50
SEED	1.00/0.15	1.00/1.00	1.00/1.00

Table 6 lists the selected minimum class-action-mass threshold c_{p} for each backbone and dataset, truncated to one decimal for display. The original-class probability-margin threshold is $c_{\text{m}} = 0.03$ in every run, and all three seeds of a backbone–dataset pair share the same gates. Gate sensitivity is reported separately in Table 21.

Table 6: Selected Self-Correction class-action-mass thresholds c_{p}.

Backbone	**ATTENTION**	**BCI-IV-2a**	**Cho2017-MI**	**ISRUC-S3**	**Mumtaz2016**	**Speech-BCIC2020-3**	**SEED-2C**
CodeBrain	0.4	0.7	0.5	1.0[†]	0.9	0.8	0.8
CSBrain	0.5	0.8	0.8	0.8	0.4	0.9	0.7
CBraMod	0.3	0.7	0.5	0.9	0.4	0.9	0.9

[†]Correction disabled in the selected configuration.

H Extended Results on Downstream EEG Tasks

This section reports extended comparisons for the six-dataset main benchmark (ATTENTION, BCI-IV-2a, SEED, ISRUC-S3, Mumtaz2016, and Speech-BCIC2020-3) and additional supplementary

settings, including Cho2017-MI. Some supplementary datasets have partial backbone coverage and are reported only in the appendix. Table 7 reports Balanced Accuracy (B-ACC), Cohen's Kappa (Kappa), and weighted F1-score (F1) for each available (Backbone, Dataset) pair.

Table 7: Extended results on downstream EEG tasks in a Baseline/GACE format.

Dataset	Backbone	Baseline B-ACC	GACE B-ACC	Baseline Kappa	GACE Kappa	Baseline F1	GACE F1
ATTENTION	CodeBrain	0.6541 ±0.0195	0.6658 ±0.0113	0.3082 ±0.0390	0.3317 ±0.0225	0.6498 ±0.0254	0.6633 ±0.0099
	CSBrain	0.6481 ±0.0063	0.6866 ±0.0069	0.2963 ±0.0125	0.3733 ±0.0139	0.6452 ±0.0086	0.6862 ±0.0076
	CBraMod	0.7109 ±0.0046	0.7374 ±0.0143	0.4218 ±0.0091	0.4749 ±0.0285	0.7089 ±0.0044	0.7370 ±0.0146
BCI-IV-2a	CodeBrain	0.4048 ±0.0152	0.4705 ±0.0211	0.2064 ±0.0203	0.2940 ±0.0282	0.3467 ±0.0409	0.4403 ±0.0326
	CSBrain	0.4899 ±0.0420	0.5489 ±0.0265	0.3198 ±0.0560	0.3985 ±0.0353	0.4675 ±0.0524	0.5375 ±0.0285
	CBraMod	0.4282 ±0.0105	0.5200 ±0.0257	0.2377 ±0.0140	0.3600 ±0.0343	0.3892 ±0.0154	0.4996 ±0.0332
Cho2017-MI	CodeBrain	0.6956 ±0.0060	0.7194 ±0.0074	0.3912 ±0.0119	0.4387 ±0.0147	0.6956 ±0.0059	0.7192 ±0.0073
	CSBrain	0.7104 ±0.0096	0.7281 ±0.0168	0.4208 ±0.0192	0.4563 ±0.0336	0.7080 ±0.0094	0.7266 ±0.0158
	CBraMod	0.5469 ±0.0174	0.5775 ±0.0108	0.0938 ±0.0348	0.1550 ±0.0217	0.5253 ±0.0461	0.5726 ±0.0181
DEAP	CodeBrain	0.5165 ±0.0174	0.5439 ±0.0109	0.0309 ±0.0342	0.0787 ±0.0253	0.3943 ±0.1164	0.4777 ±0.0709
	CSBrain	0.5118 ±0.0208	0.5143 ±0.0200	0.0237 ±0.0424	0.0289 ±0.0407	0.5410 ±0.0197	0.5423 ±0.0195
	CBraMod	0.5294 ±0.0054	0.5347 ±0.0086	0.0616 ±0.0102	0.0738 ±0.0169	0.5615 ±0.0037	0.5674 ±0.0083
ISRUC-S3	CodeBrain	0.7172 ±0.0126	0.7374 ±0.0177	0.5896 ±0.0181	0.6168 ±0.0284	0.6566 ±0.0145	0.6804 ±0.0267
	CSBrain	0.7013 ±0.0135	0.7235 ±0.0297	0.5649 ±0.0099	0.6132 ±0.0274	0.6384 ±0.0012	0.6898 ±0.0176
	CBraMod	0.7115 ±0.0320	0.7235 ±0.0457	0.5945 ±0.0376	0.6241 ±0.0531	0.6706 ±0.0273	0.7005 ±0.0405
Mumtaz2016	CodeBrain	0.8492 ±0.0137	0.9592 ±0.0140	0.7479 ±0.0191	0.8866 ±0.0202	0.8963 ±0.0075	0.9512 ±0.0085
	CSBrain	0.8654 ±0.0038	0.9425 ±0.0197	0.7596 ±0.0253	0.8418 ±0.0716	0.8999 ±0.0117	0.9305 ±0.0325
	CBraMod	0.8342 ±0.0077	0.9239 ±0.0381	0.7065 ±0.0256	0.7821 ±0.1112	0.8787 ±0.0111	0.9014 ±0.0525
SEED	CodeBrain	0.6025 ±0.0365	0.6325 ±0.0056	0.2049 ±0.0730	0.2650 ±0.0111	0.5866 ±0.0510	0.6293 ±0.0094
	CSBrain	0.6498 ±0.0125	0.6626 ±0.0087	0.2996 ±0.0250	0.3251 ±0.0173	0.6484 ±0.0137	0.6618 ±0.0097
	CBraMod	0.6391 ±0.0058	0.6745 ±0.0166	0.2782 ±0.0117	0.3490 ±0.0332	0.6316 ±0.0087	0.6733 ±0.0177
SEED-V	CBraMod	0.3926 ±0.0061	0.4107 ±0.0070	0.2430 ±0.0082	0.2655 ±0.0087	0.3953 ±0.0055	0.4176 ±0.0058
Speech-BCIC	CodeBrain	0.4836 ±0.0094	0.4844 ±0.0267	0.3544 ±0.0117	0.3556 ±0.0334	0.4824 ±0.0084	0.4837 ±0.0276
	CSBrain	0.4716 ±0.0020	0.4840 ±0.0074	0.3394 ±0.0025	0.3550 ±0.0093	0.4716 ±0.0016	0.4833 ±0.0075
	CBraMod	0.4209 ±0.0121	0.4338 ±0.0191	0.2761 ±0.0151	0.2922 ±0.0238	0.4197 ±0.0118	0.4334 ±0.0191

H.1 Competitor Results by Task

Tables 8–10 keep training, risk identification, and correction outcomes on their own scales. Each result uses the method and task coverage stated in its caption.

Table 8: Online training comparison: test B-ACC (%), three-seed mean±SD. G is Self-Guidance and A is Active Bias SGD-WPV; $\Delta = \mathrm{A} - \mathrm{G}$ is in percentage points.

Dataset	Backbone	Self-Guidance (G)	Active Bias (A)	Δ (pp)
ATTENTION	CodeBrain	**65.74**±1.08	62.65±0.81	-3.09
	CSBrain	**67.49**±1.02	66.63±0.55	-0.86
	CBraMod	**72.41**±2.58	69.18±1.23	-3.23
BCI-IV-2a	CodeBrain	**44.47**±2.89	42.65±1.97	-1.82
	CSBrain	**53.65**±3.41	53.07±0.43	-0.58
	CBraMod	49.42±2.56	**49.94**±3.18	+0.52
Cho2017-MI	CodeBrain	**71.46**±0.31	70.73±1.15	-0.73
	CSBrain	**72.12**±1.07	68.83±1.77	-3.29
	CBraMod	**56.31**±0.92	52.60±0.85	-3.71
ISRUC-S3	CodeBrain	73.74±1.77	**73.92**±0.53	+0.18
	CSBrain	**69.15**±2.06	67.32±1.77	-1.83
	CBraMod	72.13±4.32	**73.60**±3.66	+1.47
Mumtaz2016	CodeBrain	**86.91**±1.25	84.70±1.75	-2.21
	CSBrain	87.24±3.93	**87.49**±0.52	+0.25
	CBraMod	**83.87**±0.97	82.36±0.69	-1.51
SEED-2C	CodeBrain	**62.26**±0.93	61.65±0.94	-0.62
	CSBrain	**65.64**±0.58	61.11±1.42	-4.53
	CBraMod	**66.67**±1.13	63.70±0.49	-2.96
Speech-BCIC2020-3	CodeBrain	**48.40**±2.74	47.24±0.66	-1.16
	CSBrain	**48.04**±0.85	46.04±1.32	-2.00
	CBraMod	**43.20**±1.62	**43.20**±1.16	+0.00

Table 9: Matched risk ranking on current self-guided predictions: RP@12% (%), three-seed task means. R denotes Self-Assessment (the learned risk head), T TrustScore, and M Mahalanobis distance; higher is better.

Dataset	CodeBrain			CSBrain			CBraMod		
	R	T	M	R	T	M	R	T	M
ATTENTION	**52.14**	37.09	33.68	**52.14**	35.73	28.03	**54.19**	33.50	18.12
BCI-IV-2a	**78.66**	52.28	30.94	**70.26**	41.97	34.53	**72.18**	57.55	31.41
Cho2017-MI	**49.48**	23.78	34.38	**52.60**	17.01	26.22	**52.95**	47.05	44.10
SEED-2C	**51.70**	41.84	**51.70**	48.98	32.99	**53.06**	45.92	38.44	**54.08**
ISRUC-S3	**64.89**	52.84	27.30	**79.08**	73.40	21.99	**60.99**	40.43	34.75
Mumtaz2016	**63.85**	53.76	7.51	**56.10**	51.64	22.07	64.55	**74.18**	11.50
Speech-BCIC2020-3	**68.52**	57.04	51.48	**76.30**	54.07	60.37	**74.44**	56.67	61.11

Table 10: Task-wise correction gains with a frozen risk subset for each run: Self-Correction (G, implemented by the learned correction head) and nearest Mahalanobis class (M). Values are three-seed mean changes in B-ACC (percentage points) relative to each method's corresponding self-guided prediction.

Dataset	CodeBrain		CSBrain		CBraMod	
	G	M	G	M	G	M
ATTENTION	**+0.84**	+0.72	**+1.17**	+1.15	**+1.33**	+0.95
BCI-IV-2a	**+2.58**	+0.58	**+1.24**	-0.78	**+2.58**	-0.32
Cho2017-MI	+0.48	**+0.52**	**+0.69**	+0.02	**+1.44**	+0.48
ISRUC-S3	**+0.00**	-0.16	**+3.20**	+3.19	**+0.22**	-0.06
Mumtaz2016	**+9.01**	+8.75	+7.01	**+7.19**	+8.52	**+8.72**
SEED-2C	**+0.99**	+0.86	**+0.62**	-0.16	**+0.78**	+0.45
Speech-BCIC2020-3	**+0.04**	-0.71	**+0.36**	-0.31	**+0.18**	-1.02

I Ablation Study

This section complements the main ablation study in Section 4.3 with a broader set of diagnostic experiments. We organize the analysis into four types of ablations: component-level ablations for the three modules of GACE, selection-strategy ablations for the high-risk subset used by Self-Correction, operator-level ablations for the decision signals inside each module, and correction-strategy ablations comparing learned and heuristic correction policies.

I.1 Component-Level Ablation

This section reports component-level ablations for the three core parts of GACE: *Self-Guidance*, *Self-Assessment*, and *Self-Correction*. For Self-Guidance and Self-Correction, we report three task metrics (B-ACC, Kappa, and F1) to provide a complete view of the performance trend.

Self-Guidance. The first component-level ablation examines whether Self-Guidance can improve backbone fine-tuning independently (Table 11), and whether Self-Assessment improves the selection of samples for Self-Correction. As shown in Table 3, Self-Guidance already improves the baseline without using later modules. For example, B-ACC increases from 0.8492 to 0.8691 on Mumtaz2016 and from 0.4048 to 0.4447 on BCI-IV-2a, indicating that online training feedback provides a useful signal for Self-Guidance. The full model obtains the best results on both datasets and across all metrics. With matched correction-head retraining, random selection gives a small additional B-ACC gain on Mumtaz2016 but reduces B-ACC on BCI-IV-2a, and remains below the learned selector on both tasks. The broader three-backbone selection comparison appears in Table 14.

Self-Assessment. The second component-level ablation evaluates whether Self-Assessment can identify predictions that are genuinely likely to be wrong. We report reject precision at a 12% rejection rate (RP@12%), where the risk head selects the top 12% highest-risk predictions and the metric measures how many selected samples are actual errors. As shown in Table 12, the learned risk estimator consistently enriches erroneous predictions across datasets and backbones. The effect is particularly clear on BCI-IV-2a, ISRUC-S3, and Speech-BCIC2020-3, where RP@12% is usually above 0.65. The lower scores on SEED indicate that the risk ranking is more challenging for some emotion-recognition settings, but the overall trend supports Self-Assessment as the necessary bridge between Self-Guidance and Self-Correction.

Self-Correction. The third component-level ablation evaluates whether Self-Correction can further improve the self-guided predictions selected by Self-Assessment. Table 13 compares the model before and after correction. The largest correction gains appear on Mumtaz2016. For example, B-ACC improves from 0.8691 to 0.9592 on Mumtaz2016 with CodeBrain. On ISRUC-S3, CSBrain improves from 0.6915 to 0.7235, whereas CodeBrain remains at 0.7374 before and after correction; its gain over the baseline comes from Self-Guidance. On motor-imagery datasets, the gains are smaller but generally positive, while Speech-BCIC2020-3 remains nearly unchanged and SEED shows modest positive gains across backbones. This pattern is consistent with the conservative design of Self-Correction: it improves predictions when high-risk samples contain actionable residual decision structure, while avoiding broad uncontrolled changes when the correction evidence is weak.

I.2 Selection-Strategy Ablation

To compare selection rules, Random draws a reproducible 12% sample and Low-confidence takes the lowest maximum baseline softmax probabilities. Each rule selects calibration and test samples for correction-head training and evaluation.

Table 14 reports the results. Across the 21 backbone–dataset pairs, Full exceeds Random in three-seed mean correction gain in all 21. Random selection often harms balanced accuracy; Low-confidence helps on a few tasks but does not reproduce the selected Full gains consistently.

I.3 Operator-Level Ablation

We retrain Self-Assessment or Self-Correction after removing attribution features derived from Decision Evidence $\mathbf{E}(\mathbf{X})$, the trajectory statistics $\Gamma_{\mathrm{T}}(\mathbf{T}(\mathbf{X}))$ defined in the correction input, or both

Table 11: Self-Guidance ablation across all datasets in a before/after format. "Before" is the baseline EFM and "After" is the self-guided variant. Values are mean±std.

Dataset	Backbone	Before B-ACC	After B-ACC	Before Kappa	After Kappa	Before F1	After F1
ATTENTION	CodeBrain	0.6541 ±0.0195	0.6574 ±0.0108	0.3082 ±0.0390	0.3148 ±0.0216	0.6498 ±0.0254	0.6571 ±0.0111
	CSBrain	0.6481 ±0.0063	0.6749 ±0.0102	0.2963 ±0.0125	0.3498 ±0.0204	0.6452 ±0.0086	0.6744 ±0.0106
	CBraMod	0.7109 ±0.0046	0.7241 ±0.0258	0.4218 ±0.0091	0.4481 ±0.0517	0.7089 ±0.0044	0.7221 ±0.0275
BCI-IV-2a	CodeBrain	0.4048 ±0.0152	0.4447 ±0.0289	0.2064 ±0.0203	0.2596 ±0.0385	0.3467 ±0.0409	0.4062 ±0.0397
	CSBrain	0.4899 ±0.0420	0.5365 ±0.0341	0.3198 ±0.0560	0.3819 ±0.0454	0.4675 ±0.0524	0.5227 ±0.0369
	CBraMod	0.4282 ±0.0105	0.4942 ±0.0256	0.2377 ±0.0140	0.3256 ±0.0342	0.3892 ±0.0154	0.4692 ±0.0335
Cho2017-MI	CodeBrain	0.6956 ±0.0060	0.7146 ±0.0031	0.3912 ±0.0119	0.4292 ±0.0063	0.6956 ±0.0059	0.7141 ±0.0032
	CSBrain	0.7104 ±0.0096	0.7212 ±0.0107	0.4208 ±0.0192	0.4425 ±0.0213	0.7080 ±0.0094	0.7199 ±0.0096
	CBraMod	0.5469 ±0.0174	0.5631 ±0.0092	0.0938 ±0.0348	0.1262 ±0.0184	0.5253 ±0.0461	0.5576 ±0.0151
ISRUC-S3	CodeBrain	0.7172 ±0.0126	0.7374 ±0.0177	0.5896 ±0.0181	0.6168 ±0.0284	0.6566 ±0.0145	0.6804 ±0.0267
	CSBrain	0.7013 ±0.0135	0.6915 ±0.0206	0.5649 ±0.0099	0.5576 ±0.0068	0.6384 ±0.0012	0.6356 ±0.0105
	CBraMod	0.7115 ±0.0320	0.7213 ±0.0432	0.5945 ±0.0376	0.6205 ±0.0485	0.6706 ±0.0273	0.6974 ±0.0362
Mumtaz2016	CodeBrain	0.8492 ±0.0137	0.8691 ±0.0125	0.7479 ±0.0191	0.7699 ±0.0220	0.8963 ±0.0075	0.9045 ±0.0091
	CSBrain	0.8654 ±0.0038	0.8724 ±0.0393	0.7596 ±0.0253	0.7609 ±0.0636	0.8999 ±0.0117	0.8999 ±0.0260
	CBraMod	0.8342 ±0.0077	0.8387 ±0.0097	0.7065 ±0.0256	0.6702 ±0.0245	0.8787 ±0.0111	0.8591 ±0.0142
Speech-BCIC2020-3	CodeBrain	0.4836 ±0.0094	0.4840 ±0.0274	0.3544 ±0.0117	0.3550 ±0.0342	0.4824 ±0.0084	0.4832 ±0.0283
	CSBrain	0.4716 ±0.0020	0.4804 ±0.0085	0.3394 ±0.0025	0.3506 ±0.0106	0.4716 ±0.0016	0.4795 ±0.0087
	CBraMod	0.4209 ±0.0121	0.4320 ±0.0162	0.2761 ±0.0151	0.2900 ±0.0202	0.4197 ±0.0118	0.4318 ±0.0163
SEED	CodeBrain	0.6025 ±0.0365	0.6226 ±0.0093	0.2049 ±0.0730	0.2453 ±0.0185	0.5866 ±0.0510	0.6197 ±0.0106
	CSBrain	0.6498 ±0.0125	0.6564 ±0.0058	0.2996 ±0.0250	0.3128 ±0.0117	0.6484 ±0.0137	0.6551 ±0.0059
	CBraMod	0.6391 ±0.0058	0.6667 ±0.0113	0.2782 ±0.0117	0.3333 ±0.0226	0.6316 ±0.0087	0.6613 ±0.0151

Table 12: Self-Assessment reject precision at a 12% rejection rate (RP@12%) across datasets and backbones. Higher values indicate that the selected high-risk subset contains a larger fraction of actual errors.

Dataset	CodeBrain	CSBrain	CBraMod
ATTENTION	0.5214 ± 0.0129	0.5214 ± 0.0618	0.5419 ± 0.0924
BCI-IV-2a	0.7866 ± 0.0905	0.7026 ± 0.0911	0.7218 ± 0.0220
Cho2017-MI	0.4948 ± 0.0365	0.5260 ± 0.0529	0.5295 ± 0.0377
ISRUC-S3	0.6489 ± 0.0592	0.7908 ± 0.0917	0.6099 ± 0.0747
Mumtaz2016	0.6385 ± 0.0712	0.5610 ± 0.1262	0.6455 ± 0.0753
Speech-BCIC2020-3	0.6852 ± 0.0932	0.7630 ± 0.0421	0.7444 ± 0.0401
SEED	0.5170 ± 0.0118	0.4898 ± 0.0102	0.4592 ± 0.0270

Table 13: Self-Correction ablation across all datasets in a before/after format. "Before" is the self-guided output and "After" is the self-corrected output. Values are mean±std.

Dataset	Backbone	Before B-ACC	After B-ACC	Before Kappa	After Kappa	Before F1	After F1
ATTENTION	CodeBrain	0.6574 ±0.0108	0.6658 ±0.0113	0.3148 ±0.0216	0.3317 ±0.0225	0.6571 ±0.0111	0.6633 ±0.0099
	CSBrain	0.6749 ±0.0102	0.6866 ±0.0069	0.3498 ±0.0204	0.3733 ±0.0139	0.6744 ±0.0106	0.6862 ±0.0076
	CBraMod	0.7241 ±0.0258	0.7374 ±0.0143	0.4481 ±0.0517	0.4749 ±0.0285	0.7221 ±0.0275	0.7370 ±0.0146
BCI-IV-2a	CodeBrain	0.4447 ±0.0289	0.4705 ±0.0211	0.2596 ±0.0385	0.2940 ±0.0282	0.4062 ±0.0397	0.4403 ±0.0326
	CSBrain	0.5365 ±0.0341	0.5489 ±0.0265	0.3819 ±0.0454	0.3985 ±0.0353	0.5227 ±0.0369	0.5375 ±0.0285
	CBraMod	0.4942 ±0.0256	0.5200 ±0.0257	0.3256 ±0.0342	0.3600 ±0.0343	0.4692 ±0.0335	0.4996 ±0.0332
Cho2017-MI	CodeBrain	0.7146 ±0.0031	0.7194 ±0.0074	0.4292 ±0.0063	0.4387 ±0.0147	0.7141 ±0.0032	0.7192 ±0.0073
	CSBrain	0.7212 ±0.0107	0.7281 ±0.0168	0.4425 ±0.0213	0.4563 ±0.0336	0.7199 ±0.0096	0.7266 ±0.0158
	CBraMod	0.5631 ±0.0092	0.5775 ±0.0108	0.1262 ±0.0184	0.1550 ±0.0217	0.5576 ±0.0151	0.5726 ±0.0181
ISRUC-S3	CodeBrain	0.7374 ±0.0177	0.7374 ±0.0177	0.6168 ±0.0284	0.6168 ±0.0284	0.6804 ±0.0267	0.6804 ±0.0267
	CSBrain	0.6915 ±0.0206	0.7235 ±0.0297	0.5576 ±0.0068	0.6132 ±0.0274	0.6356 ±0.0105	0.6898 ±0.0176
	CBraMod	0.7213 ±0.0432	0.7235 ±0.0457	0.6205 ±0.0485	0.6241 ±0.0531	0.6974 ±0.0362	0.7005 ±0.0405
Mumtaz2016	CodeBrain	0.8691 ±0.0125	0.9592 ±0.0140	0.7699 ±0.0220	0.8866 ±0.0202	0.9045 ±0.0091	0.9512 ±0.0085
	CSBrain	0.8724 ±0.0393	0.9425 ±0.0197	0.7609 ±0.0636	0.8418 ±0.0716	0.8999 ±0.0260	0.9305 ±0.0325
	CBraMod	0.8387 ±0.0097	0.9239 ±0.0381	0.6702 ±0.0245	0.7821 ±0.1112	0.8591 ±0.0142	0.9014 ±0.0525
Speech-BCIC2020-3	CodeBrain	0.4840 ±0.0274	0.4844 ±0.0267	0.3550 ±0.0342	0.3556 ±0.0334	0.4832 ±0.0283	0.4837 ±0.0276
	CSBrain	0.4804 ±0.0085	0.4840 ±0.0074	0.3506 ±0.0106	0.3550 ±0.0093	0.4795 ±0.0087	0.4833 ±0.0075
	CBraMod	0.4320 ±0.0162	0.4338 ±0.0191	0.2900 ±0.0202	0.2922 ±0.0238	0.4318 ±0.0163	0.4334 ±0.0191
SEED	CodeBrain	0.6226 ±0.0093	0.6325 ±0.0056	0.2453 ±0.0185	0.2650 ±0.0111	0.6197 ±0.0106	0.6293 ±0.0094
	CSBrain	0.6564 ±0.0058	0.6626 ±0.0087	0.3128 ±0.0117	0.3251 ±0.0173	0.6551 ±0.0059	0.6618 ±0.0097
	CBraMod	0.6667 ±0.0113	0.6745 ±0.0166	0.3333 ±0.0226	0.3490 ±0.0332	0.6613 ±0.0151	0.6733 ±0.0177

Table 14: Selection-strategy ablation (three-seed mean ΔB-ACC in percentage points). Full uses the current RR12 setting; Random and Low-conf use trained correction heads.

Model	Dataset	Full	Random	Low-conf
CodeBrain	ATTENTION	+0.844	-1.296	+0.103
	BCI-IV-2a	+2.575	-0.752	+0.405
	Cho2017-MI	+0.479	-0.729	-0.188
	ISRUC-S3	0.000	-3.202	-1.001
	Mumtaz2016	+9.004	+0.788	+1.808
	Speech-BCIC2020-3	+0.044	0.000	+0.044
	SEED-2C	+0.988	-0.741	+0.165
CSBrain	ATTENTION	+1.173	-0.988	-0.267
	BCI-IV-2a	+1.244	-1.505	-1.649
	Cho2017-MI	+0.688	-3.437	-0.104
	ISRUC-S3	+3.198	-2.827	-0.047
	Mumtaz2016	+7.018	-1.372	+3.159
	Speech-BCIC2020-3	+0.356	-0.400	0.000
	SEED-2C	+0.617	0.000	-0.206
CBraMod	ATTENTION	+1.337	-2.757	-0.082
	BCI-IV-2a	+2.575	-0.203	+0.984
	Cho2017-MI	+1.437	-1.396	-0.083
	ISRUC-S3	+0.219	-0.277	-0.037
	Mumtaz2016	+8.517	+0.279	+1.296
	Speech-BCIC2020-3	+0.178	-0.489	0.000
	SEED-2C	+0.782	+0.206	-0.082

groups together. Table 15 reports the matched results across seven datasets. Within each dataset, the same split, seed, and other training settings are retained; correction variants also share the same selected risk subset.

Table 15: Operator-input ablation across seven datasets. Each row averages three backbones and three seeds. Full risk is RP@12% (%); full correction is ΔB-ACC (percentage points). The $\mathbf{E}$ ablation removes evidence-derived attribution features; the $\Gamma_{\mathrm{T}}(\mathbf{T})$ ablation removes trajectory statistics; the joint ablation removes both. Ablation columns show the change from the matched Full result; negative values indicate lower performance.

Dataset	Self-Assessment		Self-Correction			
	Full RP@12%	w/o $\mathbf{E}$ features	Full ΔB-ACC	w/o $\mathbf{E}$ features	w/o $\Gamma_{\mathrm{T}}(\mathbf{T})$	w/o both
ATTENTION	52.82	−3.99	+1.12	−0.47	−0.34	−0.67
BCI-IV-2a	73.70	−2.96	+2.13	−1.04	−0.87	−0.68
Cho2017-MI	51.68	−2.66	+0.87	−0.49	−0.31	−0.59
ISRUC-S3	67.38	−0.95	+1.14	−0.17	−0.20	−0.20
Mumtaz2016	61.50	−0.24	+8.18	−1.64	−0.46	−2.78
Speech-BCIC2020-3	73.09	−0.74	+0.19	−0.15	−0.10	−0.43
SEED-2C	48.87	−6.46	+0.80	−0.60	−0.33	−0.58
Mean	61.29	−2.57	+2.06	−0.65	−0.37	−0.85

I.4 Correction-Strategy Ablation

We compare Self-Correction, implemented by the selected correction MLP, with four fixed, training-free rules on exactly the same self-guided logits, complete multi-layer outputs, and Self-Assessment-selected 12% test subset. Second Softmax always uses the runner-up final class. Trajectory Top selects the highest alignment-score class distinct from the original prediction, keeping the original class when support vanishes. Probability Margin flips to the runner-up only when the top-two softmax

margin is below 0.05. Entropy Trigger flips to the runner-up when unnormalized predictive entropy exceeds 0.5 nats. The two thresholds are inherited fixed rules and are not fitted to these results. Each reported value is the equal-weight mean of three seed-level changes in balanced accuracy.

Table 16 reports all current backbone–dataset pairs. These rules are intentionally simple behavioral controls; they are not implementations of separately named published correction algorithms. The learned head and every rule share the same baseline and risk mask within each run.

Table 16: Matched Self-Correction and fixed-rule comparisons. Entries are three-seed mean ΔB-ACC in proportion units (multiply by 100 for percentage points). All methods use the same self-guided predictions and selected RR12 subset in each run.

Model	Dataset	Second Softmax	Trajectory Top	Prob. Margin	Entropy Trigger	Self-Correction
CodeBrain	ATTENTION	+0.0051	+0.0051	+0.0043	+0.0064	**+0.0084**
	BCI-IV-2a	+0.0208	+0.0116	+0.0038	+0.0205	**+0.0258**
	Cho2017-MI	-0.0013	-0.0013	-0.0031	-0.0013	**+0.0048**
	ISRUC-S3	-0.0072	-0.0198	-0.0008	-0.0057	**0.0000**
	Mumtaz2016	+0.0789	+0.0789	+0.0018	+0.0051	**+0.0900**
	Speech-BCIC2020-3	-0.0067	-0.0133	-0.0022	-0.0067	**+0.0004**
	SEED-2C	+0.0041	+0.0041	+0.0025	+0.0082	**+0.0099**
	Average	+0.0134	+0.0093	+0.0009	+0.0038	**+0.0199**
CSBrain	ATTENTION	+0.0051	+0.0051	-0.0004	+0.0051	**+0.0117**
	BCI-IV-2a	+0.0145	+0.0139	+0.0020	**+0.0177**	+0.0124
	Cho2017-MI	+0.0063	+0.0063	+0.0002	+0.0067	**+0.0069**
	ISRUC-S3	+0.0291	+0.0275	+0.0015	+0.0165	**+0.0320**
	Mumtaz2016	+0.0653	+0.0653	+0.0054	+0.0431	**+0.0702**
	Speech-BCIC2020-3	**+0.0058**	**+0.0058**	+0.0013	**+0.0058**	+0.0036
	SEED-2C	-0.0025	-0.0025	0.0000	+0.0004	**+0.0062**
	Average	+0.0177	+0.0173	+0.0014	+0.0136	**+0.0204**
CBraMod	ATTENTION	+0.0101	+0.0101	-0.0006	+0.0053	**+0.0134**
	BCI-IV-2a	+0.0145	+0.0156	-0.0003	+0.0084	**+0.0258**
	Cho2017-MI	+0.0071	+0.0071	+0.0013	+0.0013	**+0.0144**
	ISRUC-S3	+0.0019	-0.0026	+0.0021	**+0.0036**	+0.0022
	Mumtaz2016	+0.0827	+0.0827	+0.0009	+0.0117	**+0.0852**
	Speech-BCIC2020-3	+0.0027	0.0000	**+0.0049**	+0.0022	+0.0018
	SEED-2C	-0.0099	-0.0099	0.0000	-0.0111	**+0.0078**
	Average	+0.0156	+0.0147	+0.0012	+0.0031	**+0.0215**

I.5 Joint Calibration-Label Sensitivity

In Tables 17 and 18, the 10–100% columns denote the fraction of labeled calibration examples available to both Self-Assessment and Self-Correction. The Self-Correction head trains only on labeled examples within the top-risk 12% calibration subset. RP@12% is the error precision among the top-risk 12% test predictions (%); correction ΔB-ACC is the self-corrected B-ACC minus the corresponding self-guided B-ACC, in percentage points. Values are mean±SD over three seeds; the 100% column uses the full-label mainline configuration.

Table 17: Joint calibration-label sensitivity on BCI-IV-2a and Cho2017-MI: RP@12%.

Dataset	Backbone	10%	25%	50%	75%	100%
BCI-IV-2a	CodeBrain	68.82 ± 4.63	68.59 ± 4.22	71.94 ± 8.84	77.94 ± 8.34	78.66 ± 9.05
	CSBrain	67.39 ± 4.79	69.78 ± 1.25	65.23 ± 14.07	69.30 ± 11.18	70.26 ± 9.11
	CBraMod	64.99 ± 6.20	65.47 ± 5.43	72.18 ± 2.91	71.94 ± 2.59	72.18 ± 2.20
Cho2017-MI	CodeBrain	47.92 ± 2.90	47.05 ± 2.46	48.96 ± 3.65	48.44 ± 4.26	49.48 ± 3.65
	CSBrain	46.53 ± 4.24	47.92 ± 0.90	48.96 ± 4.07	50.35 ± 4.95	52.60 ± 5.29
	CBraMod	47.74 ± 3.05	47.40 ± 0.52	51.04 ± 5.02	50.52 ± 5.21	52.95 ± 3.77

Table 18: Joint calibration-label sensitivity on BCI-IV-2a and Cho2017-MI: correction ΔB-ACC.

Dataset	Backbone	10%	25%	50%	75%	100%
BCI-IV-2a	CodeBrain	$+0.23 \pm 0.18$	$+0.00 \pm 0.00$	$+0.03 \pm 0.05$	$+1.68 \pm 1.13$	$+2.58 \pm 1.78$
	CSBrain	$+0.46 \pm 0.87$	$+0.06 \pm 0.10$	$+0.98 \pm 1.10$	$+1.16 \pm 1.44$	$+1.24 \pm 1.41$
	CBraMod	$+0.00 \pm 0.00$	$+0.78 \pm 1.74$	$+2.17 \pm 0.53$	$+2.00 \pm 0.83$	$+2.58 \pm 0.10$
Cho2017-MI	CodeBrain	$+0.23 \pm 0.10$	$+0.02 \pm 0.04$	$+0.62 \pm 0.53$	$+0.37 \pm 1.37$	$+0.48 \pm 0.43$
	CSBrain	$+0.08 \pm 0.20$	$+0.19 \pm 0.17$	$+0.06 \pm 0.29$	$+0.27 \pm 0.56$	$+0.69 \pm 1.17$
	CBraMod	$+0.10 \pm 0.04$	$+0.17 \pm 0.04$	$+0.71 \pm 1.16$	$+0.60 \pm 0.75$	$+1.44 \pm 0.68$

J Reliability and Correction Diagnostics

This appendix reports additional diagnostics for the post-hoc components. We focus on three practical questions: whether the Self-Assessment risk model ranks unreliable decisions near the top, whether the Self-Correction head changes predictions safely, and how much computational overhead the post-hoc pipeline adds.

J.1 Self-Assessment Risk Ranking under Different Rejection Rates

Table 19 evaluates the Self-Assessment risk head on Cho2017-MI using reject precision. A rejection rate (RR) of $q\%$ selects the top $q\%$ highest-risk test samples, and reject precision measures the fraction of selected samples that are actual errors. The precision varies with rejection rate and backbone. At the 12% operating point, the three-seed means are 0.495 for CodeBrain, 0.526 for CSBrain, and 0.530 for CBraMod. The selected sets are enriched for errors, while admitting more samples does not always change precision monotonically.

Table 19: Self-Assessment reject precision on Cho2017-MI under different rejection rates. Values are mean±std over three seeds.

Backbone	RR 3%	RR 5%	RR 7%	RR 9%	RR 12%	RR 16%	RR 20%	RR 30%
CodeBrain	0.514±0.084	0.479±0.052	0.470±0.041	0.488±0.036	0.495±0.036	0.482±0.022	0.478±0.028	0.461±0.019
CSBrain	0.549±0.115	0.500±0.090	0.515±0.075	0.539±0.058	0.526±0.053	0.495±0.020	0.487±0.017	0.456±0.010
CBraMod	0.521±0.110	0.546±0.040	0.539±0.019	0.539±0.031	0.530±0.038	0.514±0.030	0.504±0.017	0.483±0.010

J.2 Stability and Safety Diagnostics

This section summarizes the observed flips under the selected Self-Correction configurations. Table 20 separates repaired predictions, harmful flips, and unchanged risk-selected predictions. A high keep rate means that few selected predictions were changed; it does not by itself establish that the changed predictions were correct.

Table 21 reports trimmed local sensitivity of the correction gates. For each seed, we hold the correction head and risk subset fixed and evaluate a 5×5 grid around the selected gates, with c_{p} offsets $\{-0.10, -0.05, 0, 0.05, 0.10\}$ and c_{m} offsets $\{-0.02, -0.01, 0, 0.01, 0.02\}$. We remove the five highest and five lowest test B-ACC values, compute the population standard deviation of the remaining 15 values, and average equally across three seeds and three backbones for each dataset.

Table 20: Self-Correction safety for the selected three-seed runs. $W \to C$ and $C \to W$ are mean counts per seed. Keep Rate is the fraction of risk-selected samples that retain their original prediction. Flip Precision is the mean wrong-to-correct fraction among actual flips for seeds with at least one flip; † marks a row with one undefined seed. ΔB-ACC is in percentage points; — indicates no flips in any seed.

Dataset	Backbone	$W \to C$	$C \to W$	Keep Rate	Flip Precision	ΔB-ACC (pp)
Mumtaz2016	CodeBrain	87.3	35.7	13.38%	0.709	+9.004
	CSBrain	69.0	36.7	25.59%	0.674	+7.018
	CBraMod	72.7	26.0	30.52%	0.735	+8.517
Cho2017-MI	CodeBrain	51.0	43.3	50.87%	0.539	+0.479
	CSBrain	68.0	57.0	34.90%	0.530	+0.688
	CBraMod	58.7	35.7	50.87%	0.619	+1.438
BCI-IV-2a	CodeBrain	51.3	21.7	15.11%	0.433	+2.575
	CSBrain	34.7	20.3	41.97%	0.434	+1.244
	CBraMod	40.7	11.0	44.84%	0.533	+2.575
ISRUC-S3	CodeBrain	0.0	0.0	100.00%	—	+0.000
	CSBrain	52.3	15.3	10.28%	0.618	+3.198
	CBraMod	2.7	0.3	96.81%	0.938†	+0.219
ATTENTION	CodeBrain	87.7	74.0	17.09%	0.543	+0.844
	CSBrain	60.0	41.0	48.21%	0.590	+1.173
	CBraMod	103.0	81.3	5.47%	0.557	+1.337
Speech-BCIC2020-3	CodeBrain	0.3	0.0	99.26%	0.500†	+0.044
	CSBrain	11.0	8.3	55.56%	0.275	+0.356
	CBraMod	3.0	1.7	87.04%	0.151	+0.178
SEED	CodeBrain	16.3	8.3	74.83%	0.756	+0.988
	CSBrain	12.3	7.3	79.93%	0.815	+0.617
	CBraMod	29.3	23.0	46.60%	0.569	+0.782

Table 21: Trimmed local gate sensitivity by dataset (percentage points).

Dataset	Trimmed Sens. Std (pp)
ATTENTION	0.088
BCI-IV-2a	0.364
Cho2017-MI	0.157
ISRUC-S3	0.309
Mumtaz2016	0.494
Speech-BCIC2020-3	0.196
SEED-2C	0.230
Mean	0.263

K Detailed Analysis of Decision Patterns

We examine the spatial, layer-wise, and training-time responses underlying EFM predictions. The analyses address three questions: whether individual spatial attribution patterns recur across training seeds, whether intermediate readouts give the true class a larger margin over its strongest competitor than the final erroneous readout, and how the time needed to maintain correct predictions varies across training samples. These observations motivate the information used by Self-Assessment, Self-Correction, and Self-Guidance, respectively.

Data and scope. The CodeBrain/Cho2017-MI case studies use seeds $0, 1, 2$ (Table 22). The test set contains 1,600 unique trials from eight held-out subjects, with 800 per true class; identities and labels are aligned across seeds. Spatial and layer-wise analyses use fitted self-guided checkpoints before correction, while training-time analyses use separate, uniformly weighted cross-entropy (CE) runs. These are complementary observations at specified model states, not a paired comparison of training procedures. Repeated observations of the same EEG trials across seeds are not additional independent samples. The three atlases extend these views to three models and six datasets at seed 0; extended training-response records cover all three seeds (Section K.3.1).

Table 22: Observation populations for the CodeBrain/Cho2017 decision-pattern case studies. Each seed uses the same input split.

View	Population	Model state and observation
Evidence	All 1,600 test trials; true class and prediction outcome reported separately	Self-guided checkpoint before correction; 64-channel attribution per trial
Trajectory	All final errors: 452/456/462 for seeds 0/1/2	Same fitted checkpoints; eight layer-wise probability readouts
Training responses	All 6,880 training trials per seed, evaluated after each of 50 epochs	Separate ordinary-CE fine-tuning runs; true-class probability and prediction history

K.1 Evidence: Individual Spatial Patterns behind Group Averages

Attribution protocol. The Cho2017 visualizations reuse the frozen attribution cache. Its target is $g_s(\mathbf{X}) = \max_k \ell_{s,k}(\mathbf{X})$, evaluated at each of 16 right-endpoint integration steps from a zero baseline. The maximizing class can change along this path. Absolute input-multiplied integrated contributions are summed over time and patches, then normalized over the 64 channels for each trial. This is the Decision Evidence operator defined in Appendix E.1; it does not use a fixed true-class target. We denote the resulting unit-sum channel vector by $\mathbf{a}_i^{(s)}$. Maps use the standard 10–20 template channel positions and linear interpolation. They describe model attribution on the scalp, not cortical source estimates.

Matching the same trial across seeds. To examine individual patterns beyond a group average, we display the same trial under three independently trained seeds in Figure 5. For each of the 1,600 trials, we compute

$$r_i^{(s,s')} = \mathrm{Corr}_{c=1}^{64}\left(a_{i,c}^{(s)}, a_{i,c}^{(s')}\right), \qquad r_i^{\min} = \min_{0\le s<s'\le 2} r_i^{(s,s')}. \tag{66}$$

Pearson correlation centers each vector over channels, comparing relative spatial variation around its mean of $1/64$. Taking the minimum requires similarity in all three seed pairs. Within each true-class $\times$ unanimous-correctness group, we select the trial with the largest $r_i^{\min}$, breaking ties by test index. The four groups contain 467 left/correct, 104 left/incorrect, 429 right/correct, and 120 right/incorrect trials. The other 480 trials have mixed correctness across seeds; they remain in the full-test ranking but are not in these display pools.

What the individual cases show. These trials exhibit recurring spatial emphasis across seeds, including cases that all three runs misclassify. Repeated spatial attribution can therefore occur at the individual level without implying a correct decision. The examples are deliberately selected for similarity: over the complete test set, the median $r_i^{\min}$ is 0.424, with an interquartile range of

Table 23: Selection record for Figure 5. Correctness is identical across all three seeds for each displayed trial. Ranks are descending among all 1,600 test trials.

Trial	Subject	True class	Outcome	$r_i^{\min}$	Global rank
0726	45	Left	Correct	0.750	2
1183	45	Left	Incorrect	0.730	9
0170	45	Right	Correct	0.722	10
1270	48	Right	Incorrect	0.665	37

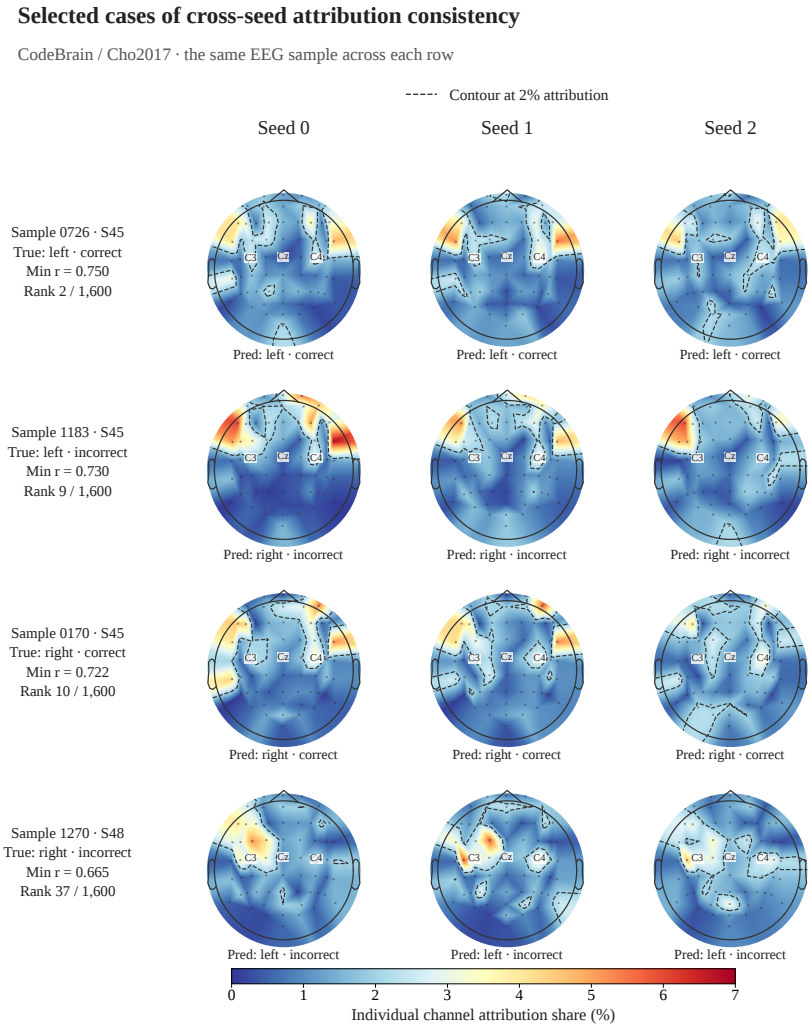

Figure 5: **Selected individual attribution patterns across seeds.** Each row fixes one EEG trial; columns show seeds 0, 1, and 2. Predictions appear below each map. All maps share a 0–7% attribution-share scale and dashed 2% isolines; these contours are display thresholds. Trials are selected by the largest minimum pairwise correlation within each class/outcome group (Table 23), rather than sampled at random. No trials or seeds are averaged.

0.339–0.510. The four cases establish the presence of recurring individual patterns, while the full-test distribution describes their variability. They do not imply that every individual resembles its group mean or that all test trials are equally stable.

Where correct and incorrect groups differ. We next compare spatial location directly. Within each true class y and seed s, let $\mathcal{I}^{\mathrm{err}}_{y,s}$ and $\mathcal{I}^{\mathrm{cor}}_{y,s}$ contain the incorrectly and correctly predicted trials. The plotted channel difference is

$$d_{y,s,c} = 100 \left(\frac{\sum_{i \in \mathcal{I}^{\mathrm{err}}_{y,s}} a^{(s)}_{i,c}}{|\mathcal{I}^{\mathrm{err}}_{y,s}|} - \frac{\sum_{i \in \mathcal{I}^{\mathrm{cor}}_{y,s}} a^{(s)}_{i,c}}{|\mathcal{I}^{\mathrm{cor}}_{y,s}|} \right). \tag{67}$$

Figure 6 reports this difference in percentage points of normalized attribution share. For left-hand trials, the C4 share is lower in the incorrect group by 0.959, 0.857, and 0.118 percentage points for seeds 0, 1, and 2. For right-hand trials, the C3 share is higher in the incorrect group by 0.584, 1.611, and 0.528 points. Other locations vary across runs: for example, the right-hand C4 difference changes sign across seeds. The observed contrast concerns the spatial distribution of relative attribution, with both recurring and run-dependent changes.

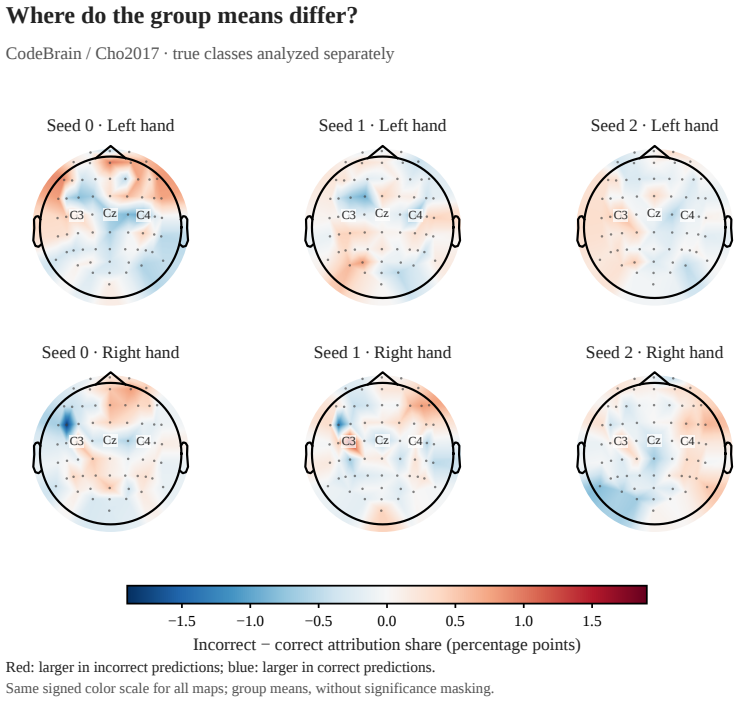

Figure 6: **Incorrect-minus-correct spatial attribution.** Differences are computed separately by true class and seed on the complete test set. All six maps share a symmetric scale in attribution-share percentage points; red denotes a larger mean share among incorrect predictions. Correct/incorrect group sizes for left and right are 566/234 and 582/218 (seed 0), 631/169 and 513/287 (seed 1), and 573/227 and 565/235 (seed 2). C3, Cz, and C4 are landmarks. These are outcome-conditioned group-mean differences, without sample pairing or significance masking.

Connection to Self-Assessment. The individual maps and group differences ask complementary questions: whether a spatial pattern recurs for a fixed input, and where prediction groups differ on average. They motivate retaining channel location when assessing decision reliability. Since each vector is normalized, a larger share at one channel is a relative redistribution rather than a larger absolute attribution magnitude. Risk-prediction utility is evaluated by the Self-Assessment experiments (the reject-precision evaluation in Table 19), rather than inferred from visual separation alone.

K.1.1 Mean Spatial Attribution and Outcome Contrasts across Models and Tasks

Figures 7–18 use seed-0 self-guided checkpoints before correction. All test trials are averaged separately by true class and correctness after per-trial channel normalization. Mean maps label the three largest channel shares. Each dataset's separate difference figure shows correct minus incorrect means: red indicates higher correct-group shares, blue higher incorrect-group shares. Differences are in percentage points except for ISRUC-S3, which uses units of uniform channel share (one sixth). Labels mark the positive and negative extrema. Each figure type uses one shared scale per dataset. Dense montages use template positions and linear interpolation; ISRUC retains six named derivations without interpolation. Missing groups remain undefined. These descriptive group contrasts do not establish individual consistency or statistical significance.

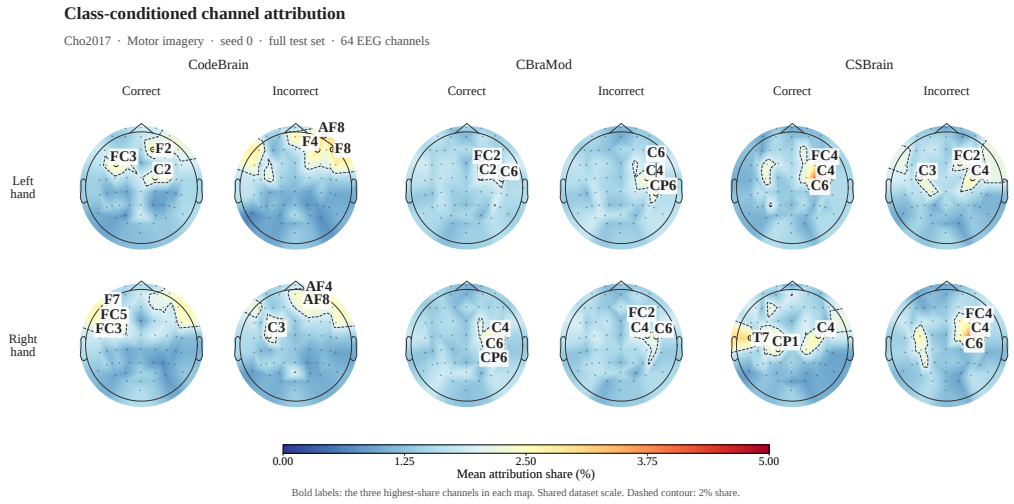

Figure 7: **Cho2017: motor imagery.** All test trials per model, separated by left/right-hand imagery and prediction correctness. The common mean-share scale is 0–5%; dashed contours mark 2%. Labels identify each map's three highest-share channels.

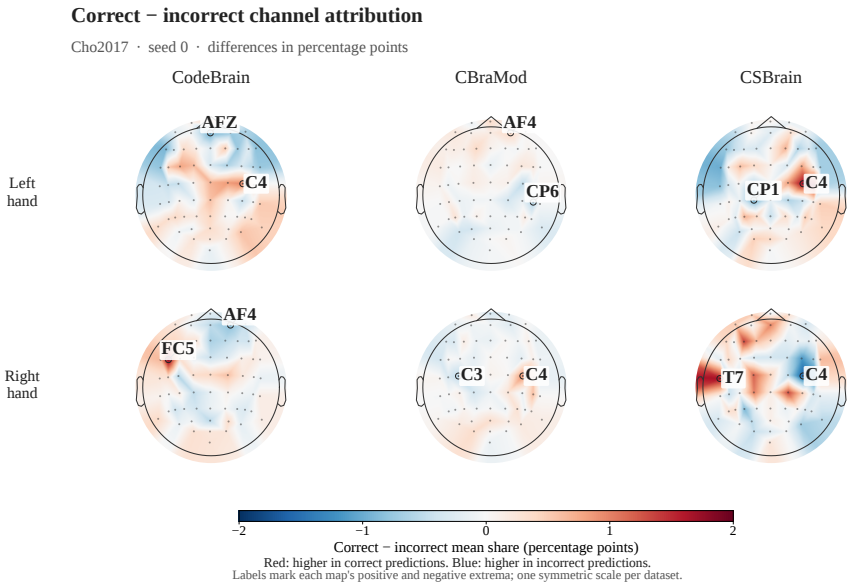

Figure 8: **Cho2017: correct-minus-incorrect attribution.** Differences use the same model and true-class groups as Figure 7, on a shared ± 2 percentage-point scale. Red denotes larger correct-group shares; blue denotes larger incorrect-group shares. The extreme channels are shown.

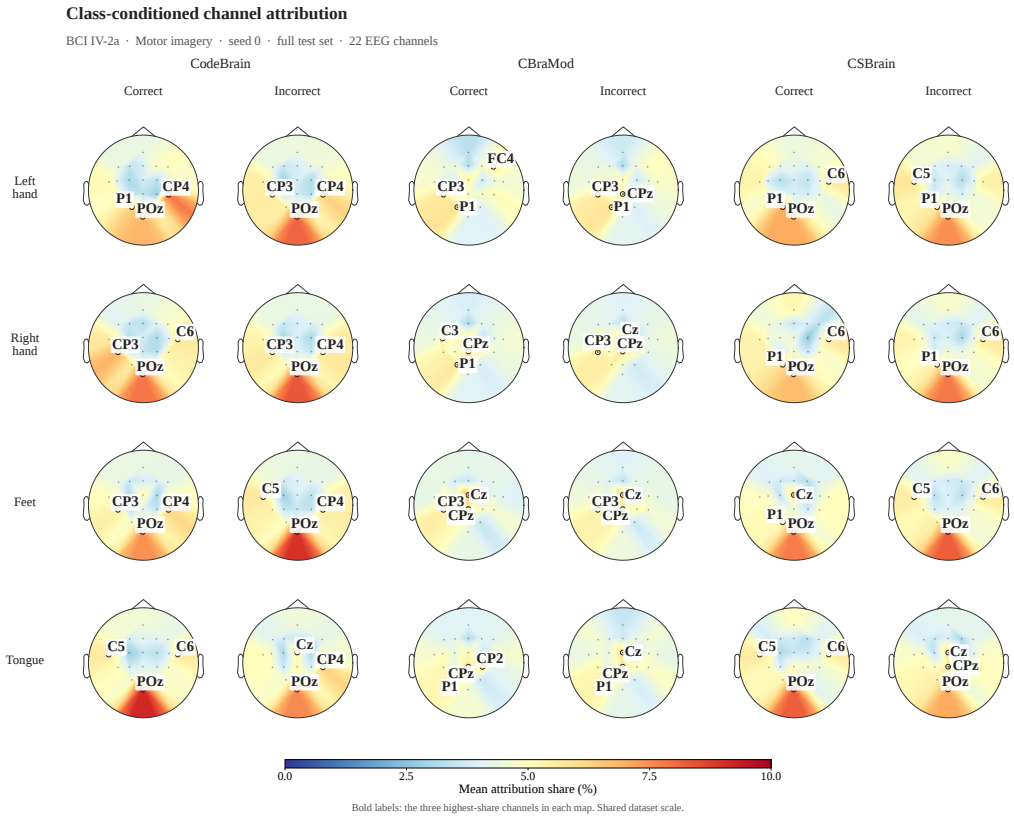

Figure 9: **BCI IV-2a: four-class motor imagery.** All test trials per model, with left hand, right hand, feet, and tongue averaged separately by prediction correctness. The common mean-share scale is 0–10%. Labels identify each map's three highest-share channels.

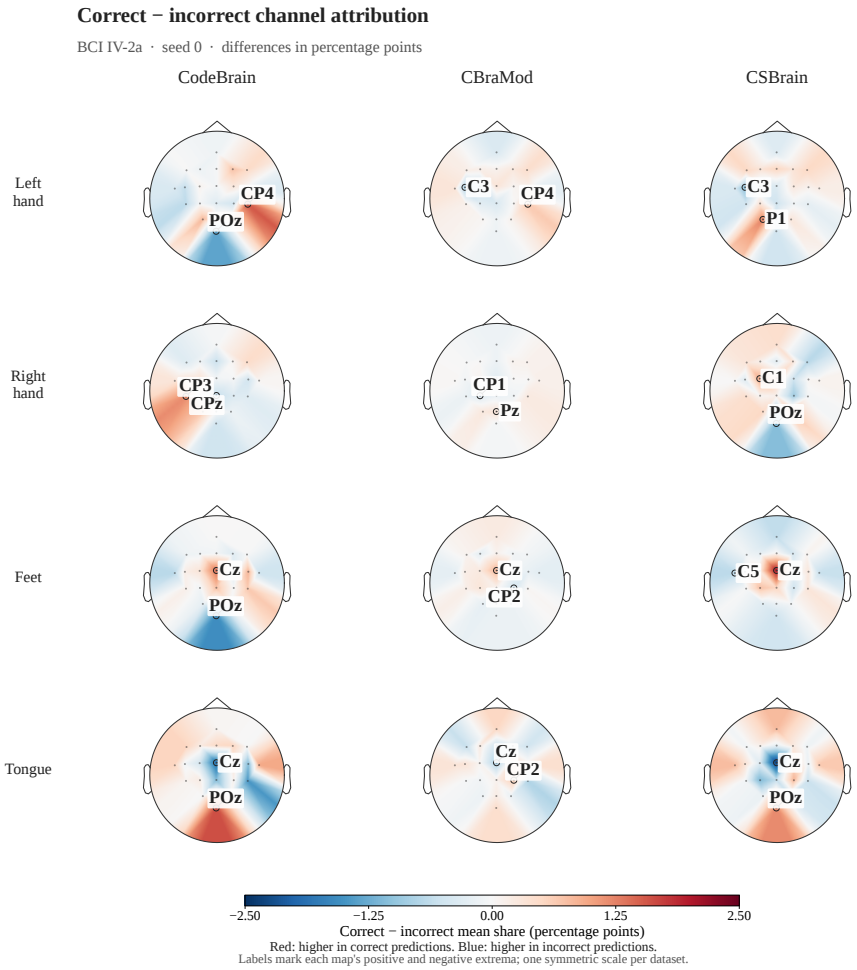

Figure 10: **BCI IV-2a: correct-minus-incorrect attribution.** Differences use the same model and true-class groups as Figure 9, on a shared ± 2.5 percentage-point scale. Red denotes larger correct-group shares; blue denotes larger incorrect-group shares. The extreme channels are shown.

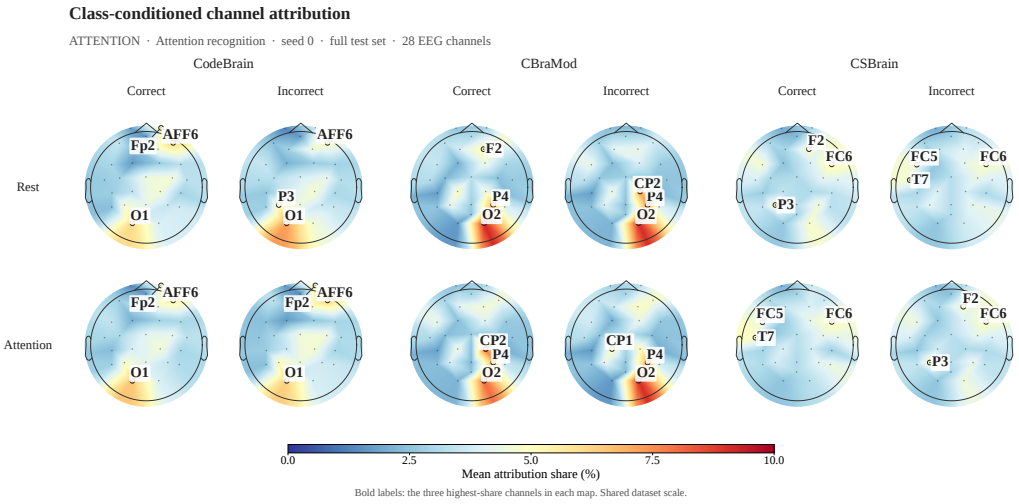

Figure 11: **ATTENTION: attention recognition.** All test trials per model, with rest and attention-task segments averaged separately by prediction correctness. The common mean-share scale is 0–10%. Labels identify each map's three highest-share channels.

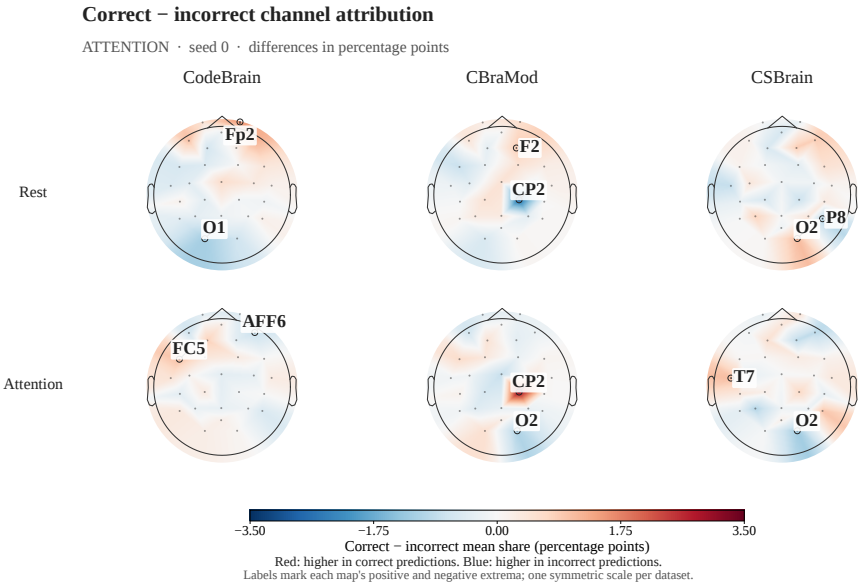

Figure 12: **ATTENTION: correct-minus-incorrect attribution.** Differences use the same model and true-class groups as Figure 11, on a shared ± 3.5 percentage-point scale. Red denotes larger correct-group shares; blue denotes larger incorrect-group shares. The extreme channels are shown.

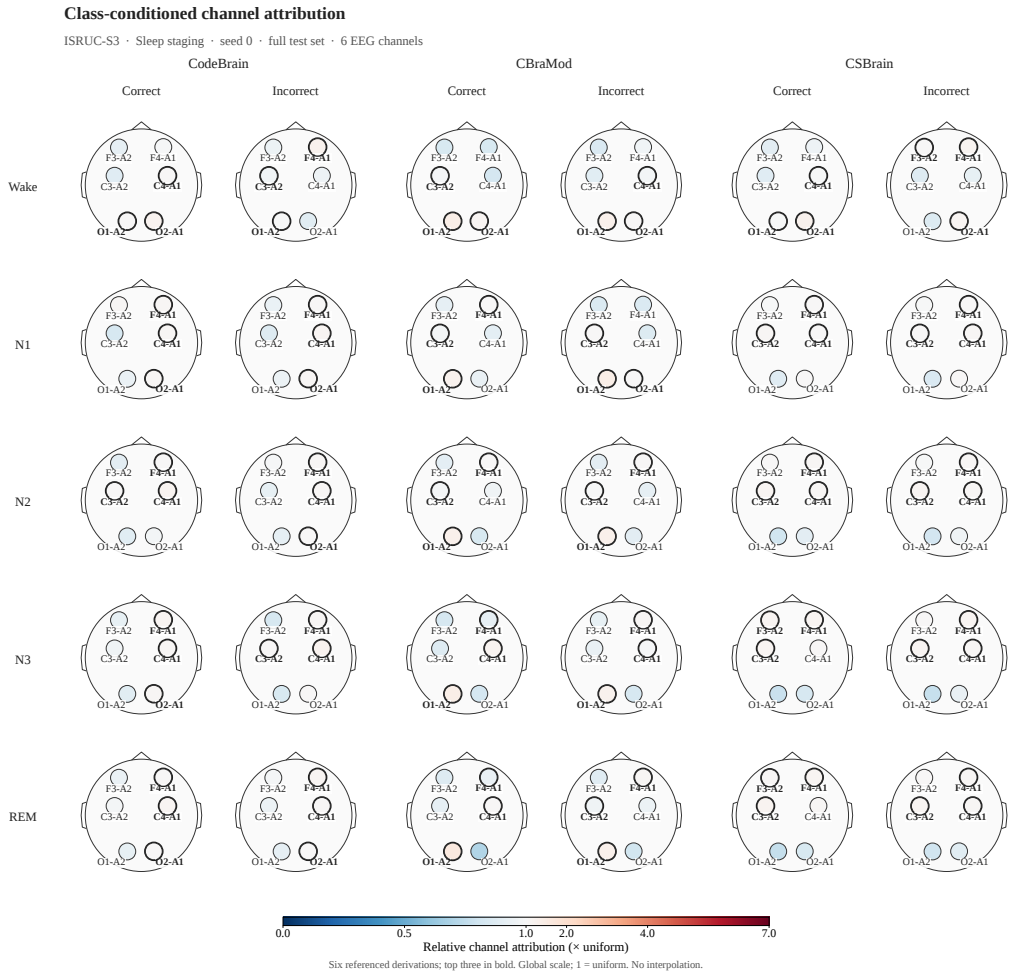

Figure 13: **ISRUC-S3: sleep staging.** All test epochs per model from the predefined S3 test subject. Six referenced derivations are shown without spatial interpolation; attribution is scaled relative to uniform channel share, with 1 denoting one sixth. Every model–class pair has both correct and incorrect predictions. Labels identify each map's three highest-share channels.

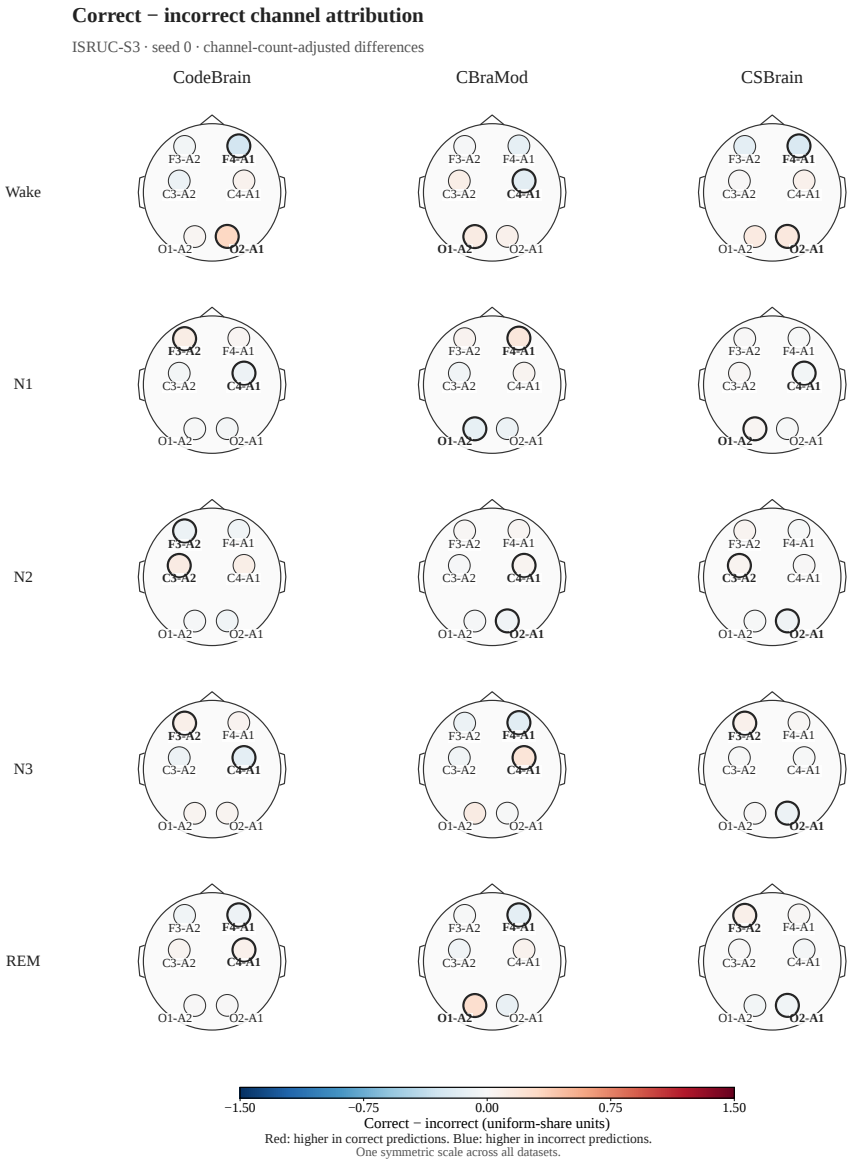

Figure 14: **ISRUC-S3: correct-minus-incorrect attribution.** Differences use the same S3 test epochs and true-class groups as Figure 13. Red denotes larger correct-group shares; blue denotes larger incorrect-group shares. Differences are expressed in units of uniform channel share (one sixth). The six derivations are shown; all 15 model–class contrasts are defined.

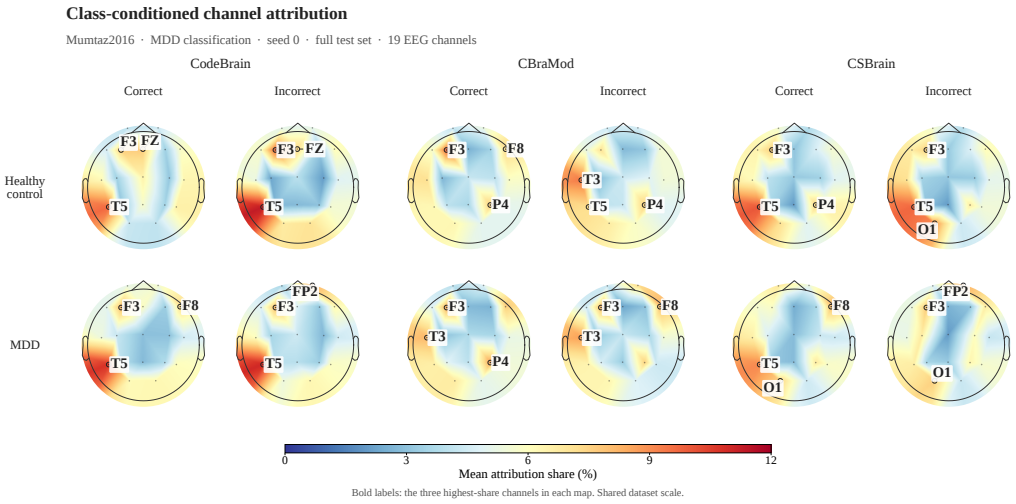

Figure 15: **Mumtaz2016: MDD classification.** All test trials per model, with healthy-control and MDD trials averaged separately by prediction correctness. The common mean-share scale is 0–12%. Labels identify each map's three highest-share channels.

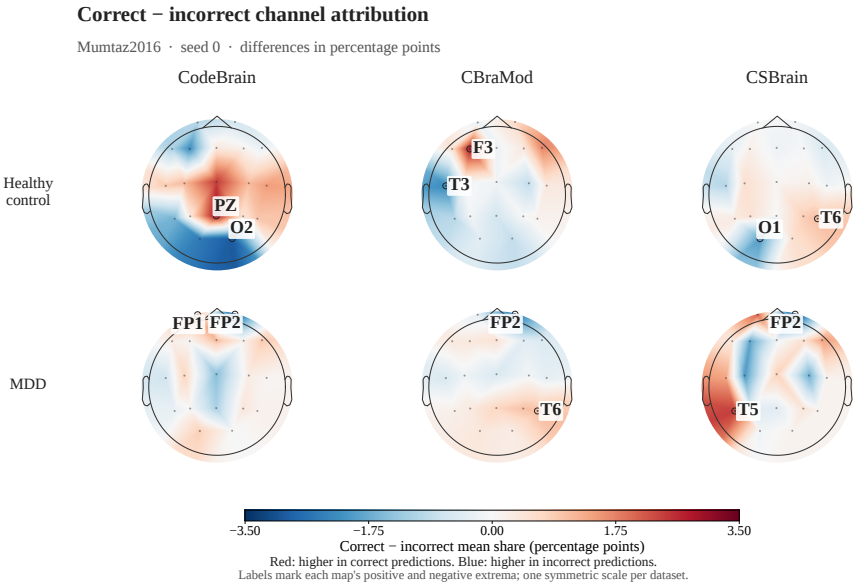

Figure 16: **Mumtaz2016: correct-minus-incorrect attribution.** Differences use the same model and true-class groups as Figure 15, on a shared ± 3.5 percentage-point scale. Red denotes larger correct-group shares; blue denotes larger incorrect-group shares. The extreme channels are shown.

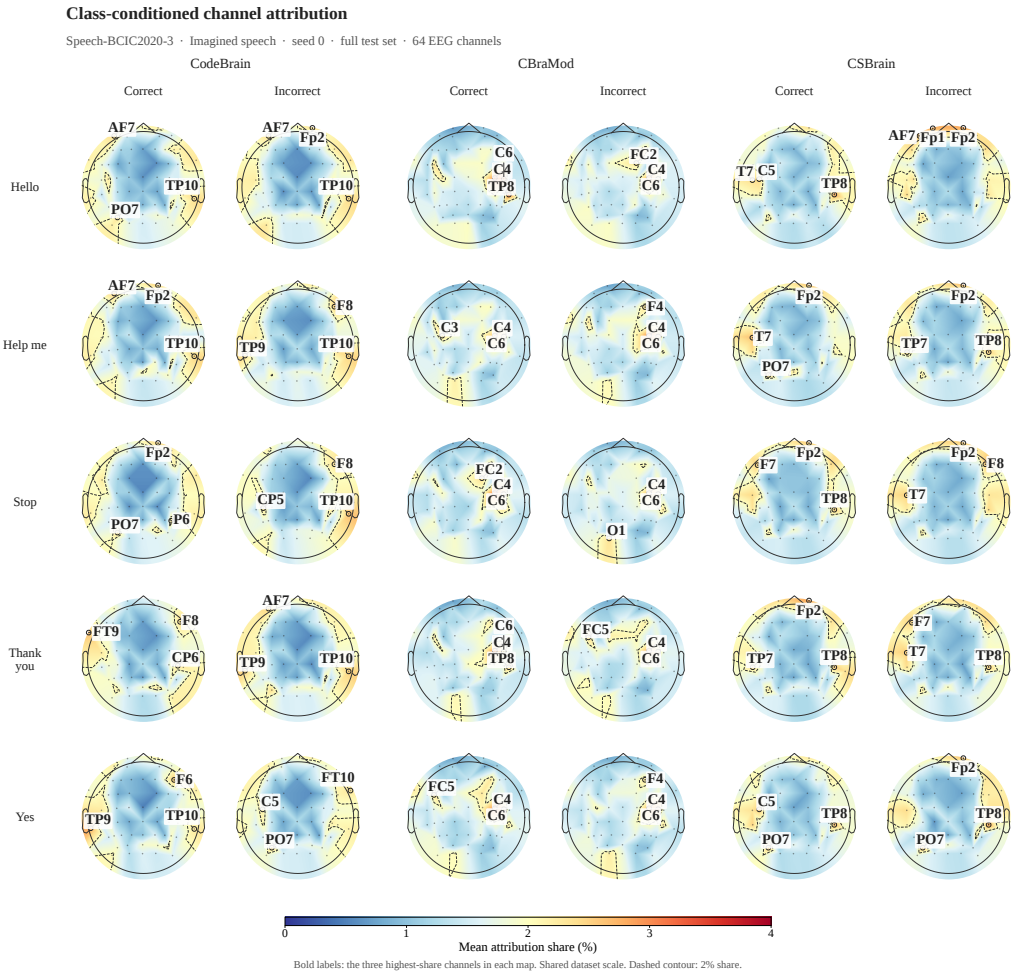

Figure 17: **Speech-BCIC2020-3: imagined speech.** All test trials per model, separated into five true classes and prediction correctness. The 64-channel mapping was recovered from the original files and verified against every processed test signal. The common mean-share scale is 0–4%, with dashed 2% contours. Labels identify each map's three highest-share channels.

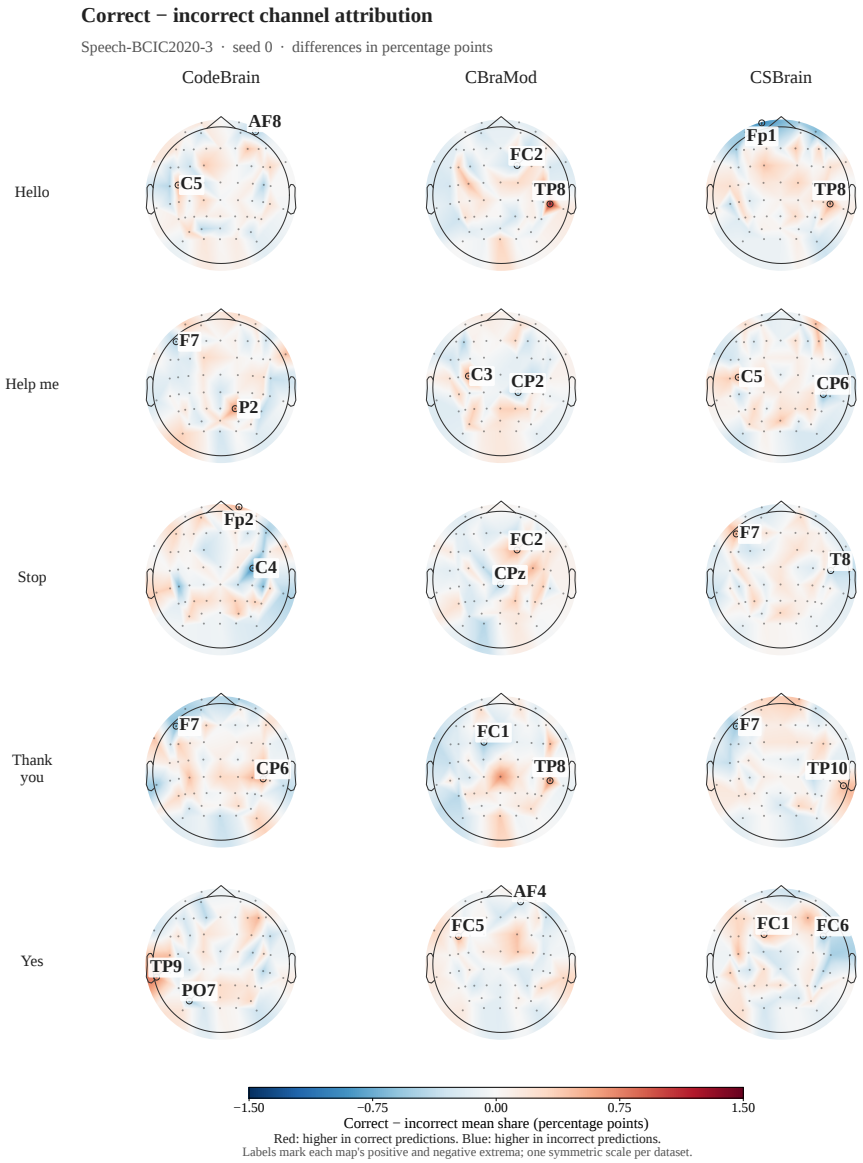

Figure 18: **Speech-BCIC2020-3: correct-minus-incorrect attribution.** Differences use the same model and true-class groups as Figure 17, on a shared ± 1.5 percentage-point scale. Red denotes larger correct-group shares; blue denotes larger incorrect-group shares. The extreme channels are shown.

K.1.2 Self-Guidance and Correct–Incorrect Attribution Separation

We compare ordinary cross-entropy (CE) fine-tuning with online Self-Guidance on the three reported datasets. We evaluate the final checkpoint before correction. Attribution uses the same 16-step, zero-baseline maximum-logit-path protocol as the spatial analyses above, with absolute contributions normalized across channels for each trial.

For method m, seed s, and true class k, let $\boldsymbol{\mu}_{m,s,k}^{\mathrm{correct}}$ and $\boldsymbol{\mu}_{m,s,k}^{\mathrm{incorrect}}$ be the mean normalized attribution vectors of that method's correct and incorrect test predictions. We summarize their spatial separation by the class-averaged total-variation distance

$$D_{m,s} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{2} \left\| \boldsymbol{\mu}_{m,s,k}^{\mathrm{correct}} - \boldsymbol{\mu}_{m,s,k}^{\mathrm{incorrect}} \right\|_1 \tag{68}$$

For each method, let $\overline{D}_m = \frac{1}{3} \sum_{s=0}^{2} D_{m,s}$. The relative change reported in the table is

$$\Delta_{\mathrm{rel}} = \frac{\overline{D}_{\mathrm{SG}} - \overline{D}_{\mathrm{CE}}}{\overline{D}_{\mathrm{CE}}} \times 100\%. \tag{69}$$

where $\overline{D}_{\mathrm{CE}} > 0$ for every reported row. Larger $D \in [0, 1]$ indicates a larger difference between the correct and incorrect spatial means. Each true class receives equal weight. Group membership is determined separately for CE and Self-Guidance, so changes can reflect both attribution and prediction-group composition.

Table 24: **Correct–incorrect attribution separation with and without Self-Guidance.** The two D columns report the mean separation for each dataset. The final column is the percentage change relative to CE, computed before rounding.

Dataset	CE D	Self-Guidance D	Relative change (%)
Cho2017	0.06816	0.08308	+21.89
ATTENTION	0.05276	0.06806	+29.01
Speech-BCIC2020-3	0.05458	0.05847	+7.14

Across the three reported datasets, the mean correct–incorrect attribution separation increases from 0.05850 under CE to 0.06987 under Self-Guidance, a relative increase of 19.44% (Table 24).

K.2 Cross-Layer Class Competition across Models and Datasets

For each final error, we compute $m_l = p^{(l)}(y) - \max_{c \neq y} p^{(l)}(c)$ and plot $100 m_l$ in percentage points. A positive individual margin indicates a correct readout. Samples are grouped by the earliest layer maximizing $m_l - m_L$ above 10^{-8}; otherwise they enter "None." Peak positions follow from this grouping rule. Lines show unsmoothed group means and bands show sample interquartile ranges; a positive mean does not imply that every member is correct. Legend percentages use all final errors. Figures 19–24 retain native depths and share vertical limits across models within each dataset. These retrospective groups differ from the correction head's alignment feature and do not measure correction success. All three seeds and per-sample assignments are archived; the figures show seed 0.

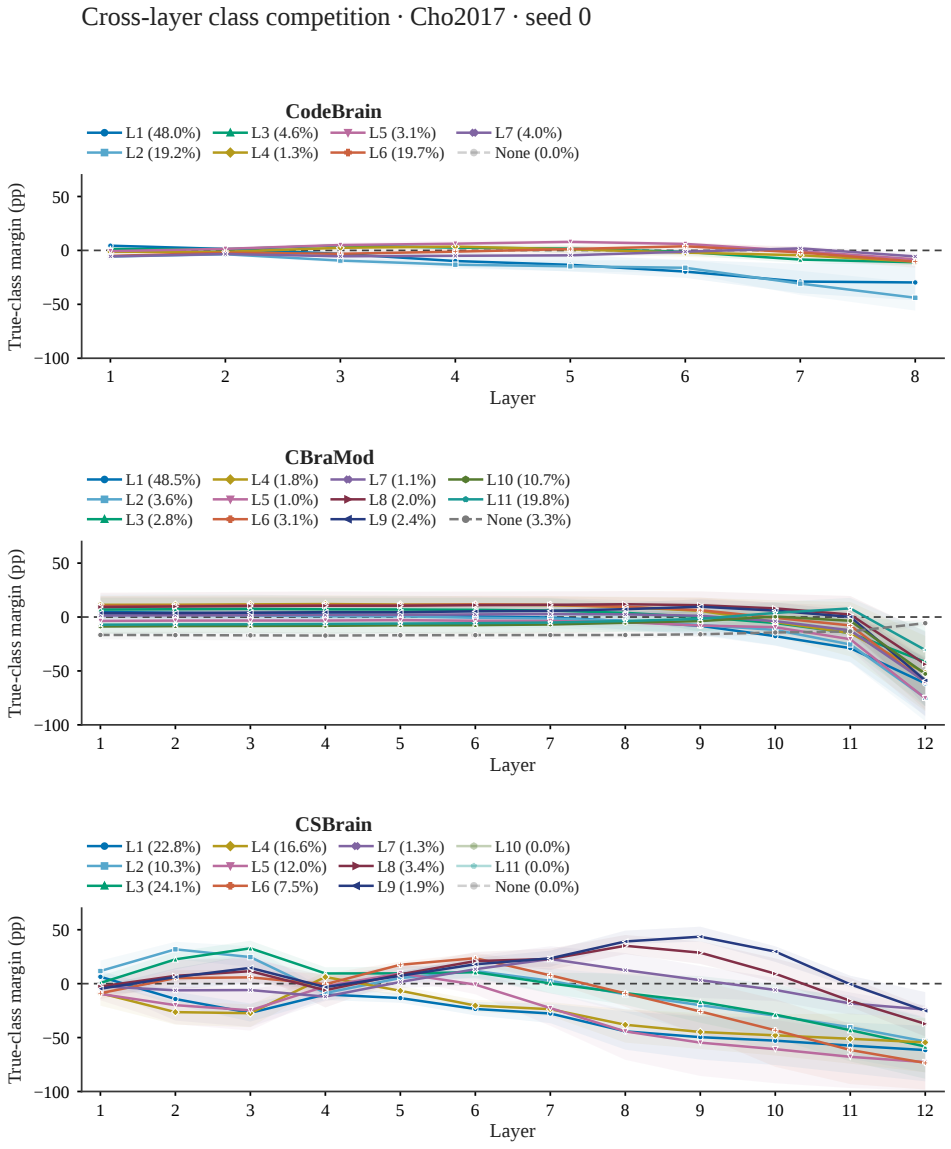

Figure 19: **Cho2017: cross-layer class competition.** Three models, seed 0; means and sample interquartile ranges over all final errors.

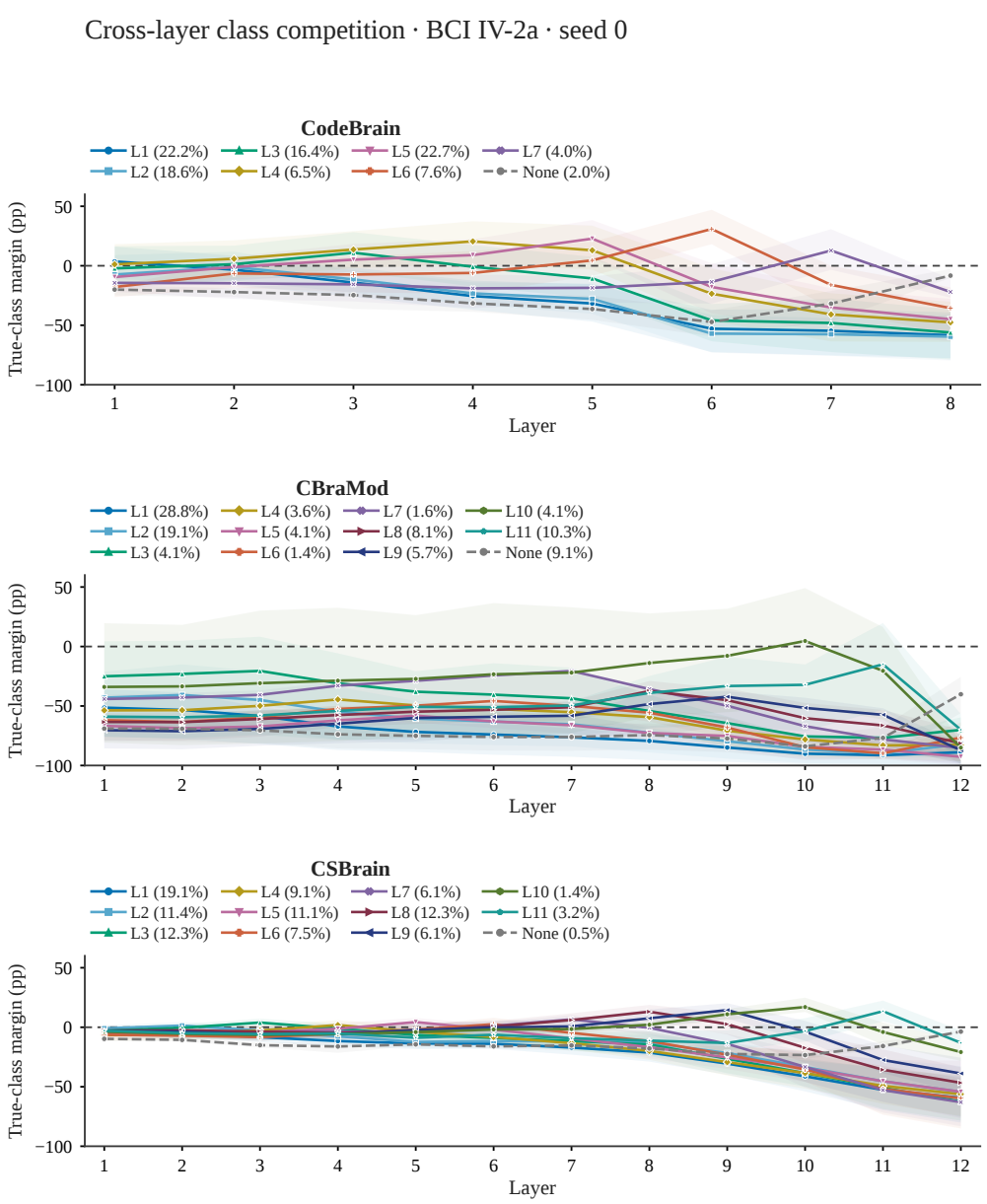

Figure 20: **BCI IV-2a: cross-layer class competition.** Seed 0; all final errors. Margins compare the true class with the strongest competing class at each layer.

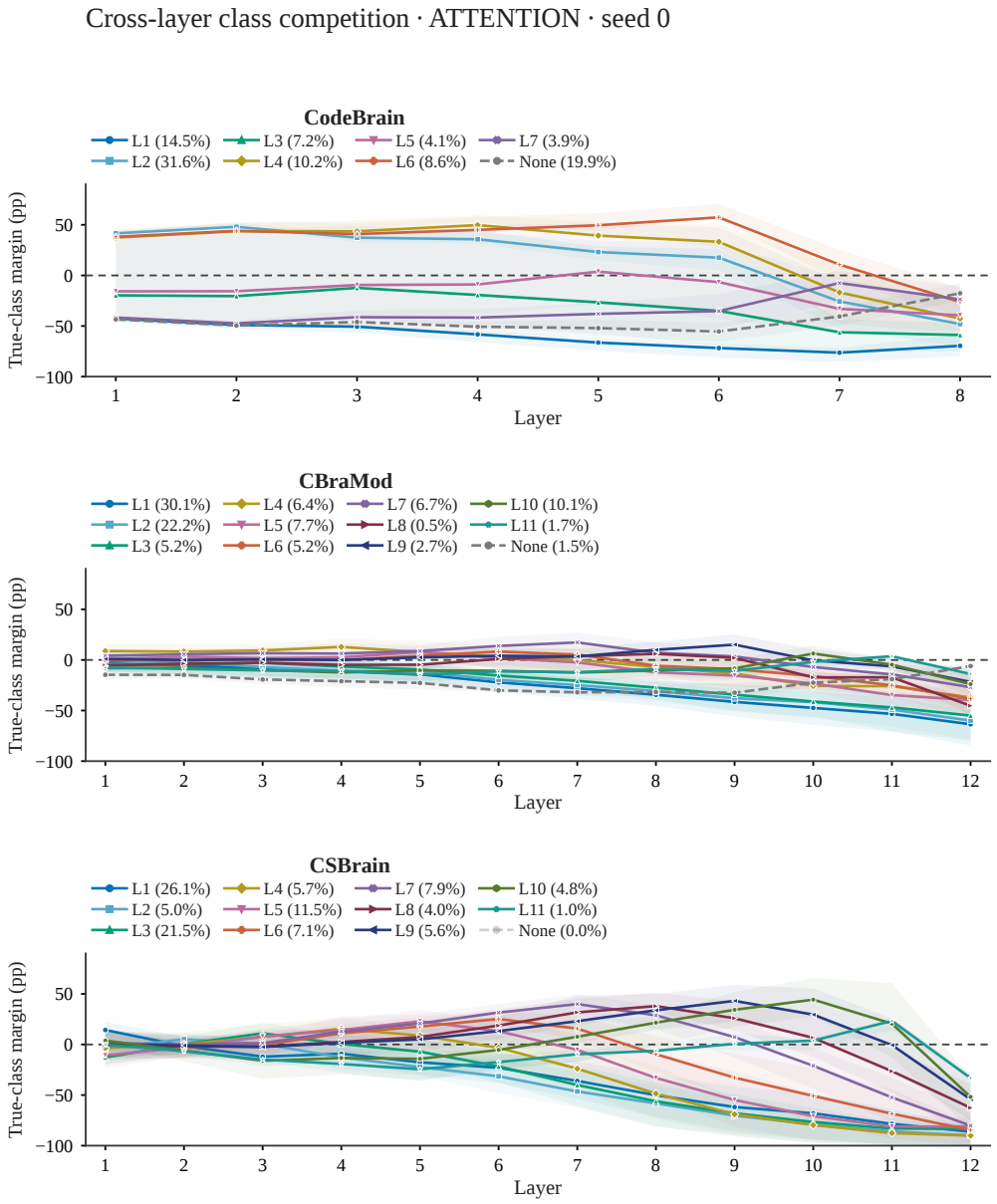

Figure 21: **ATTENTION: cross-layer class competition.** Seed 0; all final errors. Group means and sample-IQR bands follow Section K.2.

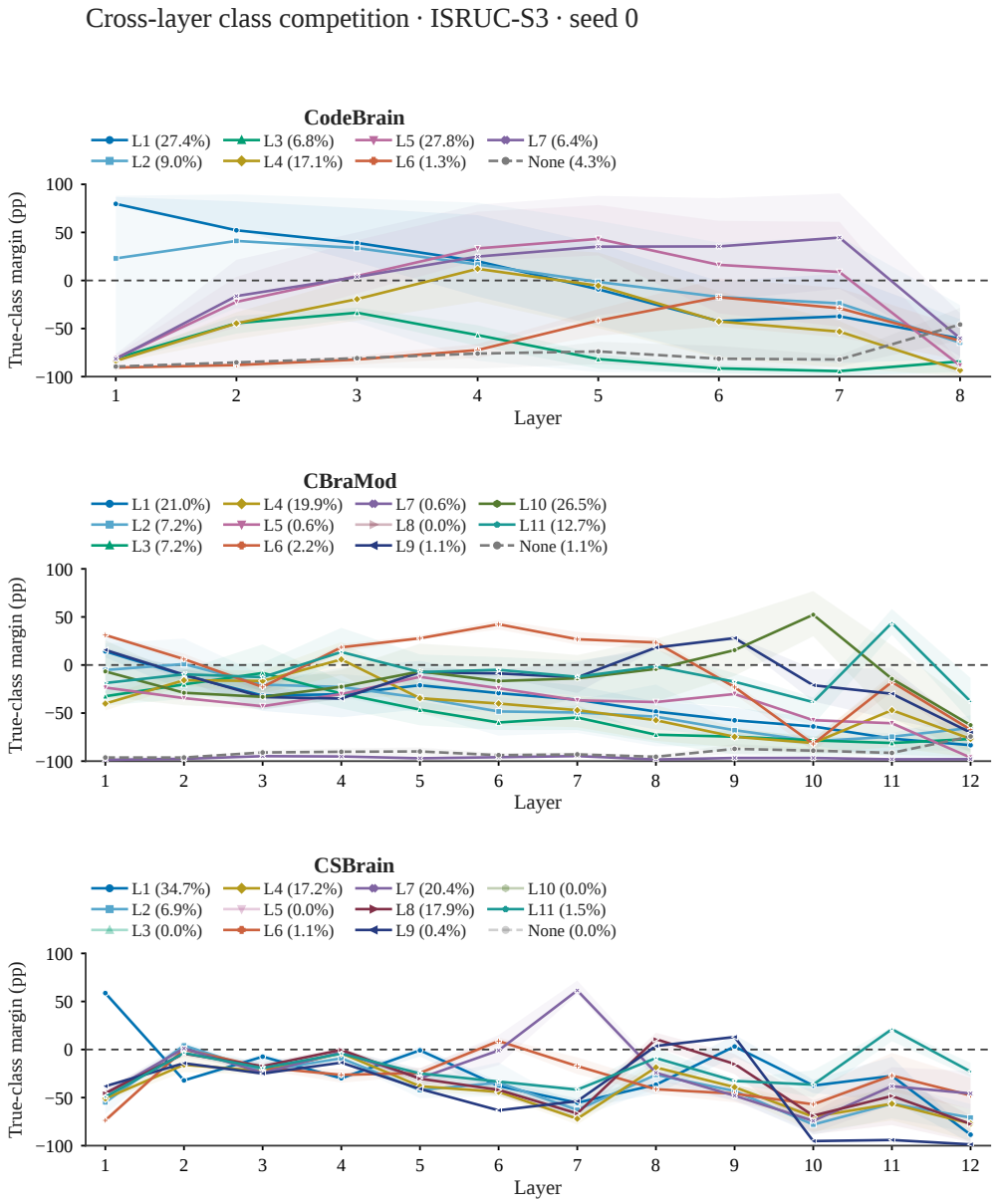

Figure 22: **ISRUC-S3: cross-layer class competition.** Seed 0; all final errors. "None" denotes no earlier margin advantage; vertical limits are shared across models.

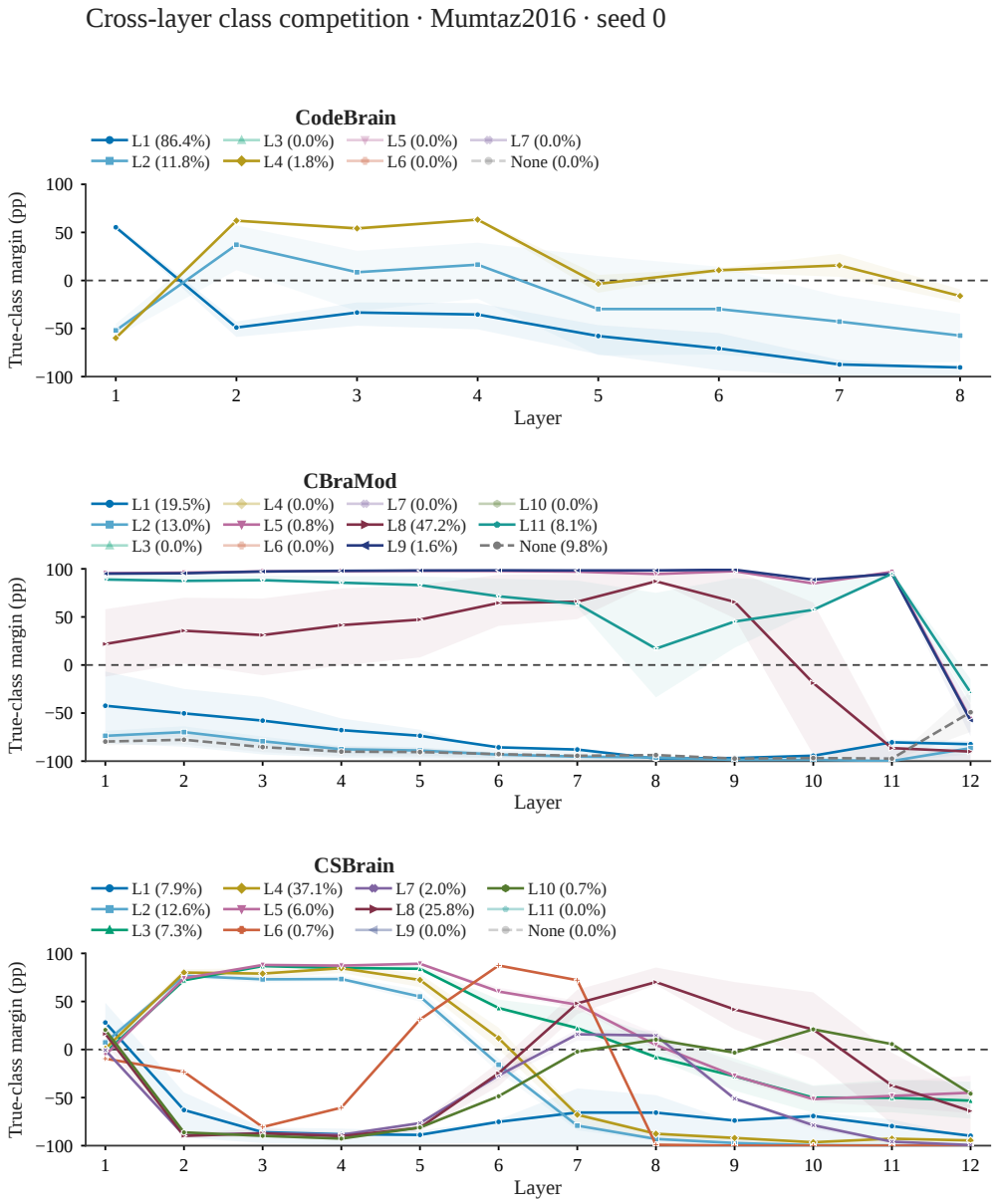

Figure 23: **Mumtaz2016: cross-layer class competition.** Seed 0; all final errors. Lines are group means and bands are sample interquartile ranges.

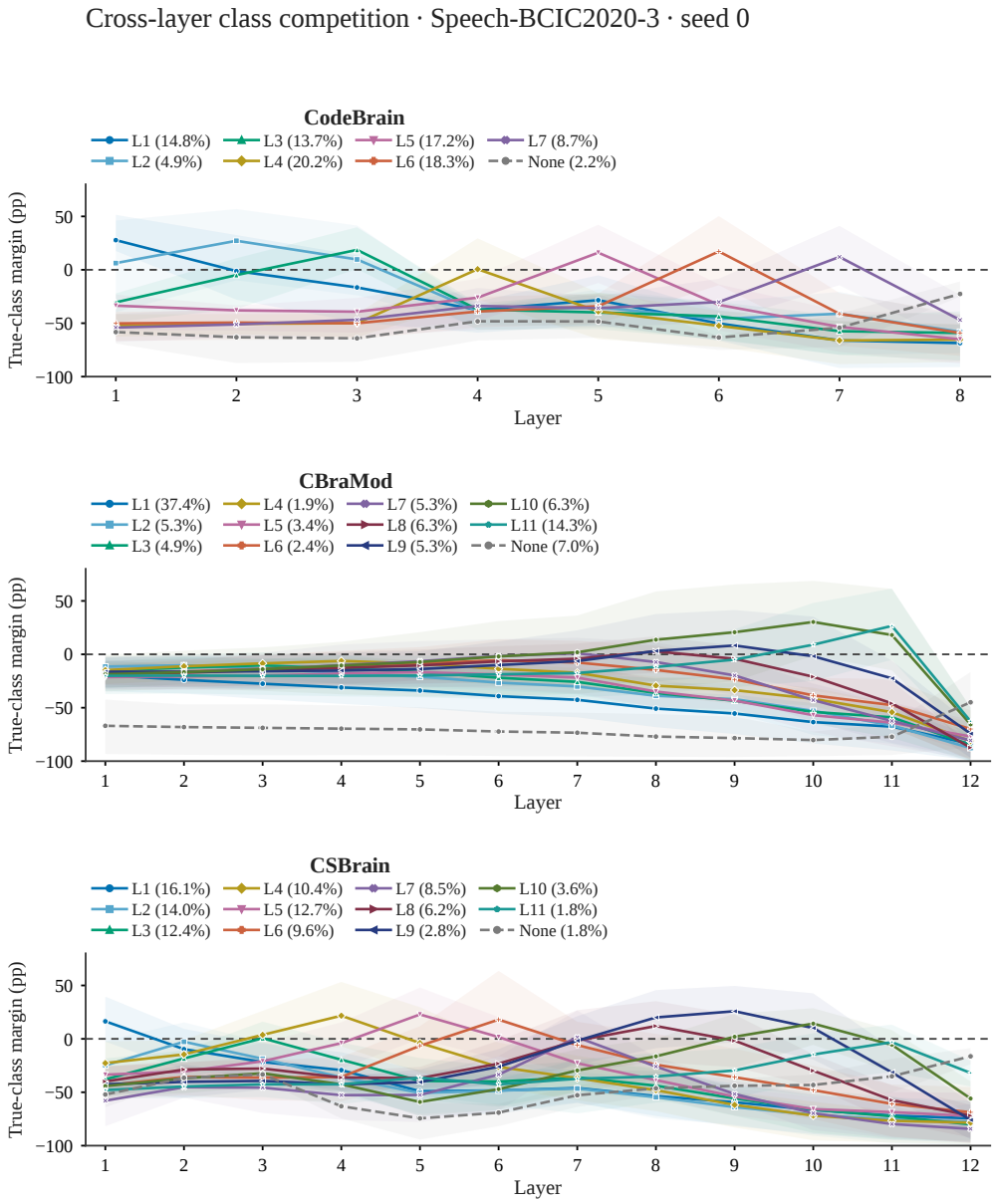

Figure 24: **Speech-BCIC2020-3: cross-layer class competition.** Seed 0; all final errors contribute once to a peak-layer or "None" group.

K.3 Self-Guidance Motivation: Variation in the Time to Persistent Correctness

Observing ordinary fine-tuning. For each seed, we record true-class probabilities and predictions for all 6,880 training trials after each of 50 uniformly weighted CE epochs, with the model fixed during evaluation. These histories describe ordinary fine-tuning before introducing feedback-based weighting, rather than within-batch online weights.

An operational grouping. Let $b_{i,s,t} = \mathbf{1}[\hat{y}_{i,s,t} = y_i]$ for epoch $t \in \{1, \ldots, 50\}$. The length of the terminal correct run is

$$R_{i,s} = 50 - \max\big(\{t : b_{i,s,t} = 0\} \cup \{0\}\big). \tag{70}$$

An onset is confirmed only if $R_{i,s} \geq w$, with $\tau_{i,s} = 51 - R_{i,s}$. For $w = 5$, confirmed onsets at epochs 1–16 are *early* and those at epochs 17–46 are *later*. A wrong final prediction or a correct terminal run shorter than five epochs is *not yet confirmed*. Earlier reversals are allowed. Unconfirmed trials are counted separately, without an artificial confirmed onset at epoch 51. These are retrospective groups, whose sizes are reported in Table 25.

Table 25: Persistent-correctness groups under the five-epoch confirmation rule. Entries are trial count (percentage of 6,880 training trials).

Seed	Early	Later	Not yet confirmed
0	3,801 (55.25%)	2,851 (41.44%)	228 (3.31%)
1	3,256 (47.33%)	3,268 (47.50%)	356 (5.17%)
2	4,285 (62.28%)	2,416 (35.12%)	179 (2.60%)

The distribution behind the grouped curves. Figure 25 shows the full onset distribution. Across runs, 35.12–47.50% of training samples become persistently correct after epoch 16, and 2.60–5.17% remain unconfirmed. This variation includes initially difficult trials learned later; it does not establish three intrinsic sample clusters or equate stable predictions with optimization convergence.

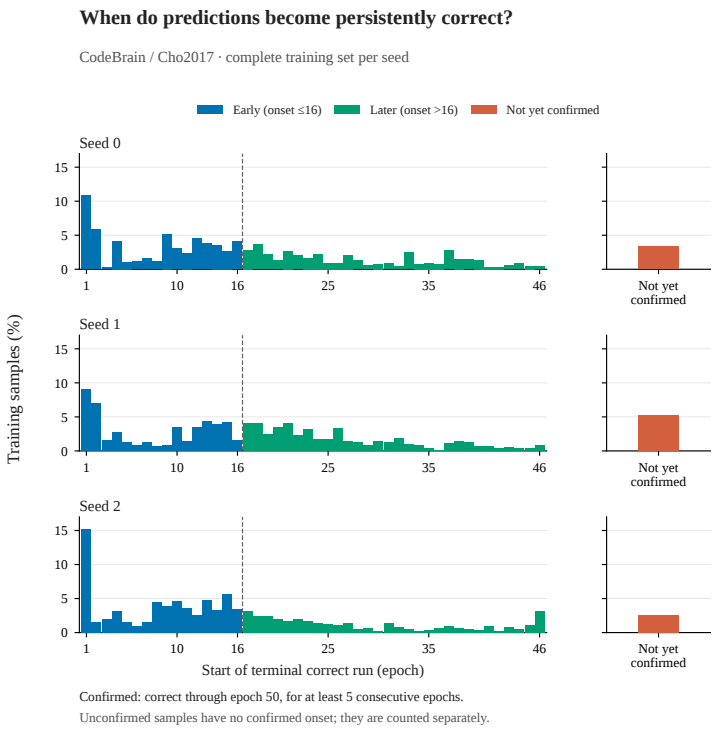

Figure 25: **Distribution of persistent-correctness onset.** Each seed includes all 6,880 training trials. Bars report the fraction with each confirmed onset; early (1–16) and later (17–46) onsets are distinguished. Predictions must remain correct through epoch 50 for at least five epochs. Unconfirmed trials are counted separately. Earlier prediction reversals are permitted. These are descriptive groups computed after ordinary CE training.

Individual histories beneath a group mean. Figure 26 shows every trial's true-class probability across all 50 epochs. Within each seed, rows are ordered by early/later/unconfirmed group, terminal-correct onset, and trial index. The displays retain native epochs without smoothing, shifting time, or subsampling trials. Sorting exposes differences in when support becomes sustained, while preserving within-group variation that can be hidden by a mean curve. Row order is seed-specific: matching row numbers across panels does not match the EEG inputs. Full matrices and row-to-trial mappings are retained with the analysis data.

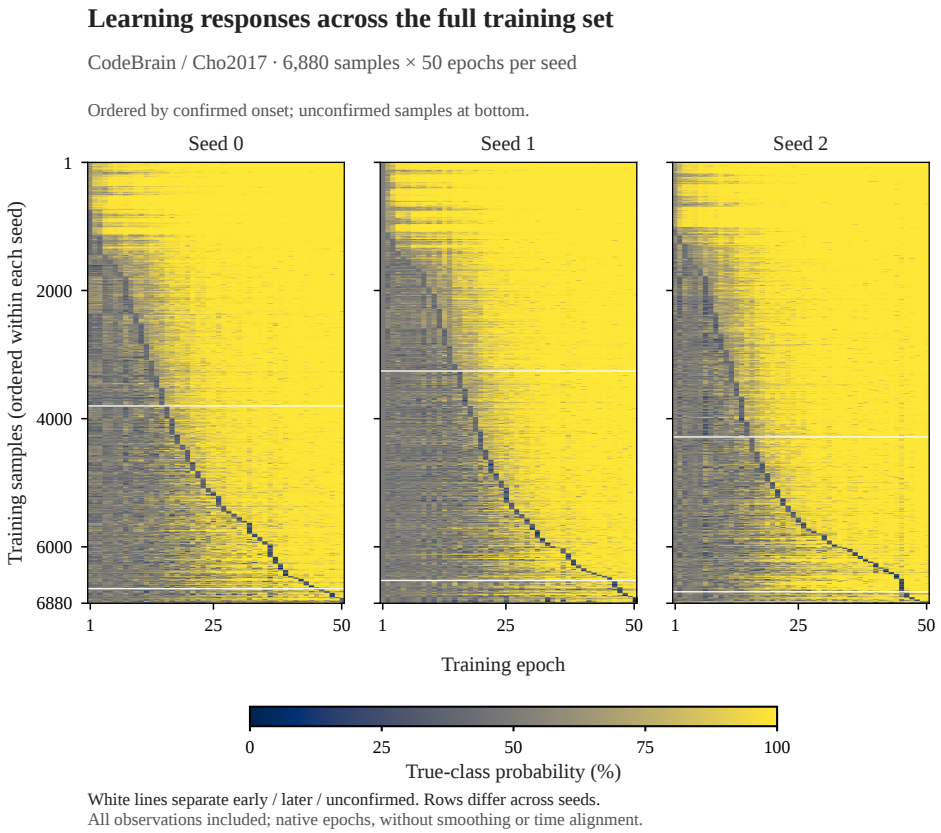

Figure 26: **Complete training probability histories.** Panels show all 6,880 trials by 50 epochs for each seed, using a common 0–100% true-class probability scale. White lines separate early, later, and unconfirmed groups under $w = 5$. Rows are sorted independently for each seed, and all sample–epoch values are included. Dense individual rows may not be distinguishable at printed resolution. These are epoch-end ordinary-CE observations, not applied online weights.

Dependence on the persistence definition. We keep the early boundary at epoch 16 and vary w over 3, 5, and 10 (Figure 27). The unconfirmed fraction is 2.50/3.31/6.02% for seed 0, 2.51/5.17/7.49% for seed 1, and 1.02/2.60/8.28% for seed 2. Longer requirements move short correct terminal runs from the later group into the unconfirmed group. The early group is unchanged by construction: onset no later than epoch 16 already guarantees a terminal correct run of at least 35 epochs. This analysis quantifies the sensitivity of the summary to its definition.

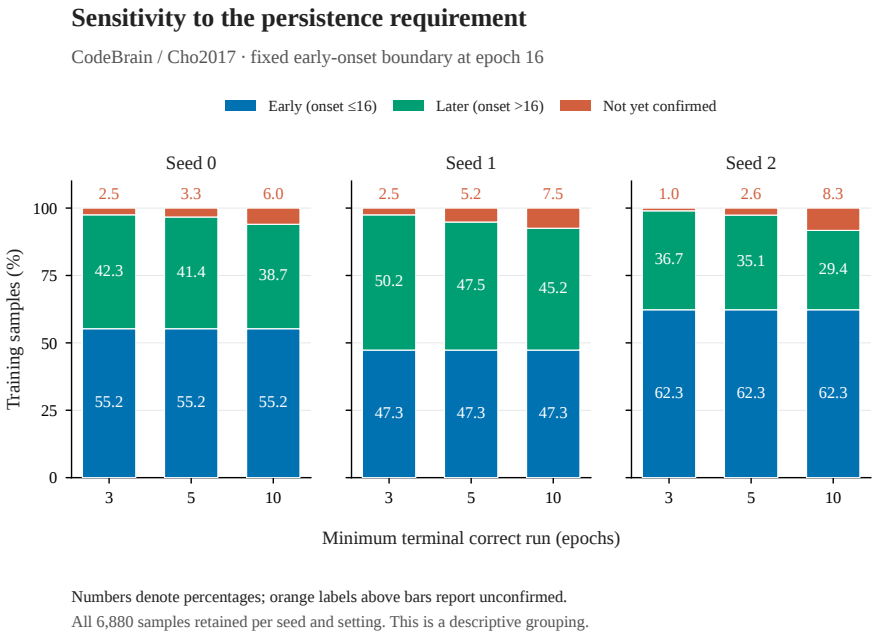

Figure 27: **Sensitivity to the persistence requirement.** Each bar partitions all 6,880 trials using a minimum terminal correct run of three, five, or ten epochs, with the early boundary fixed at epoch 16. Labels give group percentages, including the unconfirmed fraction above each bar. Early-group invariance follows from the definition; changes in later and unconfirmed groups show how much the classification depends on the required observation duration.

From the observation to Self-Guidance. The histories motivate revisiting each sample's training weight as learning proceeds: low early support need not persist, and a temporarily favorable prediction does not establish persistent support. Self-Guidance uses these evolving responses as training feedback, aiming to support task-relevant representations from which Decision Evidence and Decision Trajectory are obtained. Margin support measures separation of the labeled class from its strongest competitor (Eq. (36)); confidence fluctuation measures deviation of true-class confidence from its running mean (Eq. (37)). The two quantities address class support and temporal variation, and enter the weighting rule in Eq. (40). Unlike the retrospective groups, these online statistics use only the training history available at the current update.

The probability histories motivate this feedback-based strategy without identifying a unique weighting formula. An across-sample interquartile band describes variation between trials, whereas confidence fluctuation is computed from a given trial's changes over time. The figures do not themselves measure applied online weights or the effect of deleting weighted samples. Guidance comparisons and separately specified interventions address effectiveness and changes in decision formation; the observations here provide the preceding empirical motivation.

K.3.1 Self-Guidance Motivation: Training Responses across Models and Datasets

Figures 28–33 cover complete training sets under ordinary CE. Cho2017/BCI use $T = 50$ epochs, ATTENTION/Mumtaz/Speech use $T = 30$, and the updated ISRUC-S3 comparison uses $T = 30$ for CodeBrain/CSBrain and $T = 50$ for CBraMod. A correct terminal run of at least five epochs confirms an onset: early by $\lfloor T/3 \rfloor$, later thereafter; remaining trials are unconfirmed. Earlier reversals are allowed. Lines are unsmoothed true-class probability means; bands are sample interquartile ranges. Percentages use all training samples; empty groups have no curve. Seed 0 is fixed for display. The accompanying data summarize all displayed model–dataset runs and three-seed group fractions; the earlier six-dataset records additionally include 3/5/10-epoch persistence sensitivity.

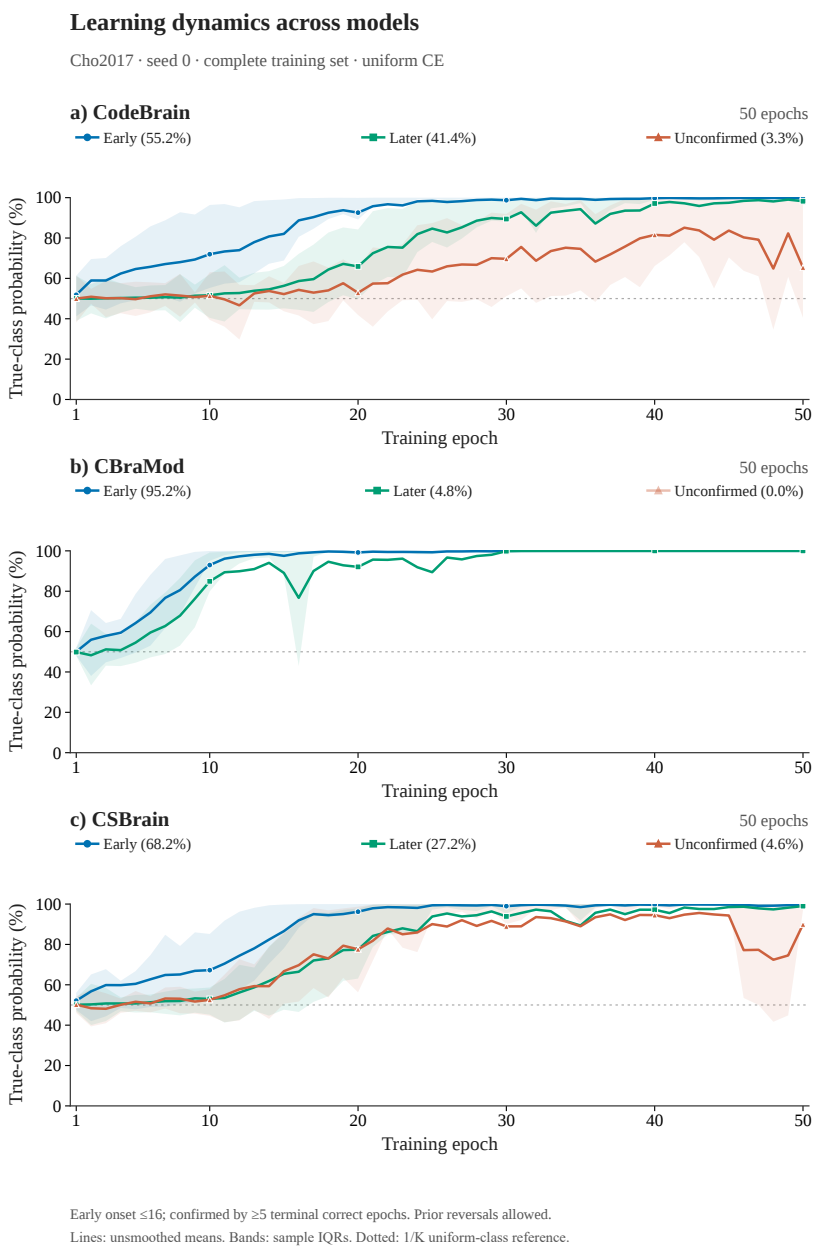

Figure 28: **Cho2017: training responses across three models.** All 6,880 training trials per model, over 50 ordinary-CE epochs. Early onsets are epochs 1–16; later confirmed onsets are 17–46. Shading represents sample dispersion, not uncertainty across seeds. The dotted line is the uniform-class reference $1/2$.

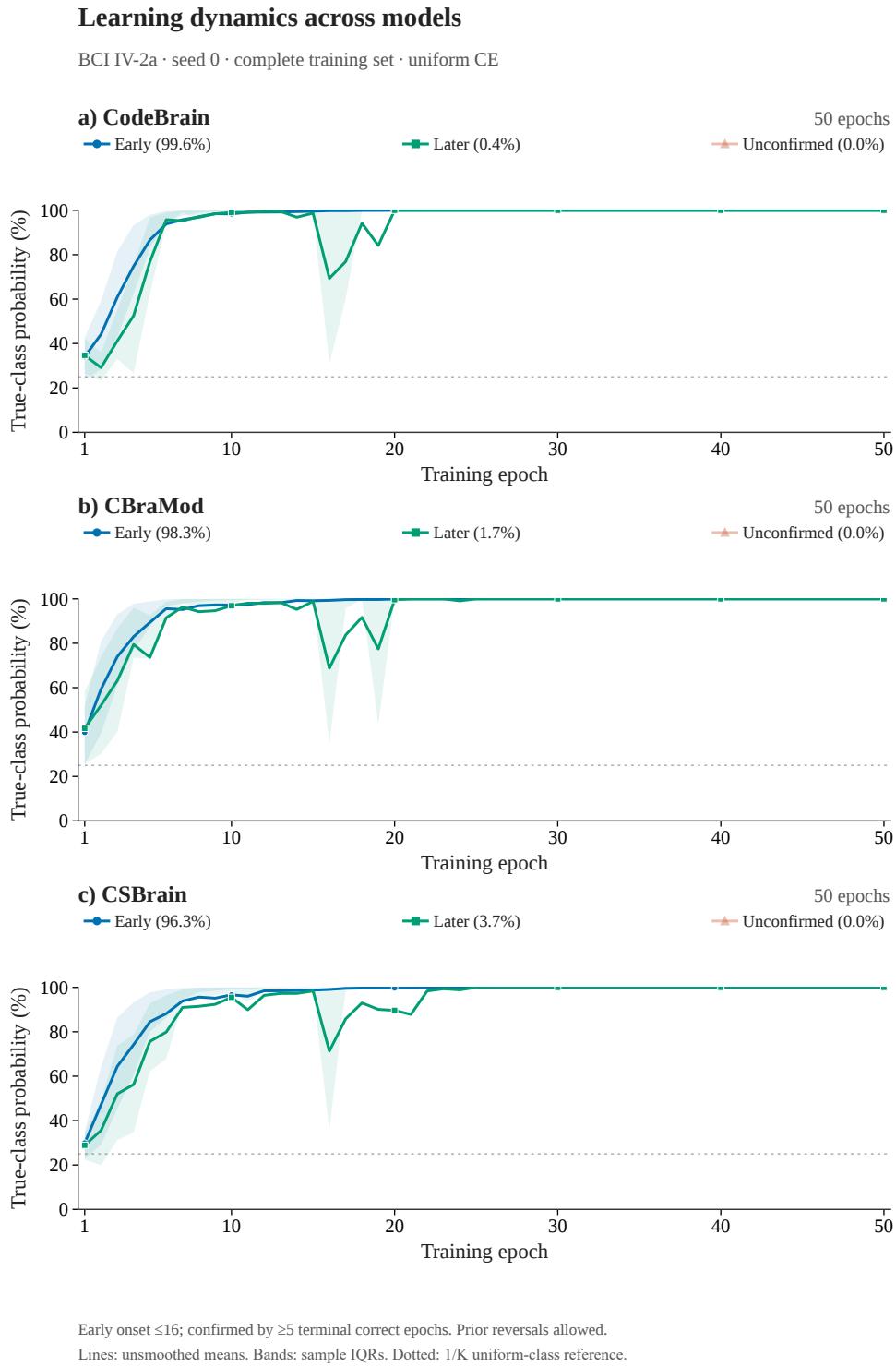

Figure 29: **BCI IV-2a: training responses across three models.** All 2,784 training trials per model, four classes, 50 epochs. Grouping and sample-IQR bands follow Section K.3.1. The dotted $1/4$ line is a uniform-class reference, not a multiclass decision boundary. Empty groups remain explicit.

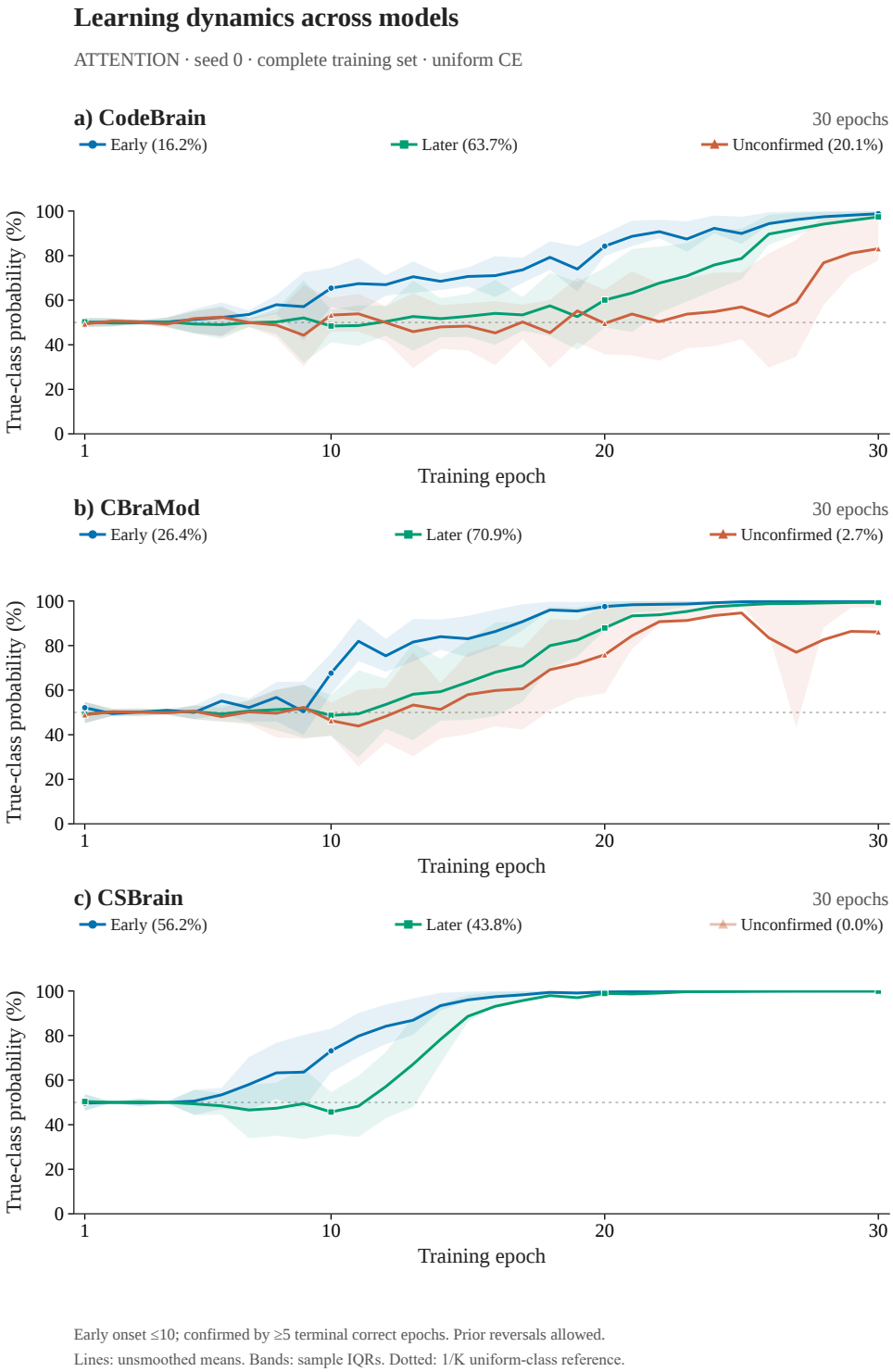

Figure 30: **ATTENTION: training responses across three models.** All training trials per model, two classes, 30 epochs. Early onsets are epochs 1–10; later confirmed onsets are 11–26. Lines and bands retain the observed group means and sample interquartile ranges; percentages use the full training population.

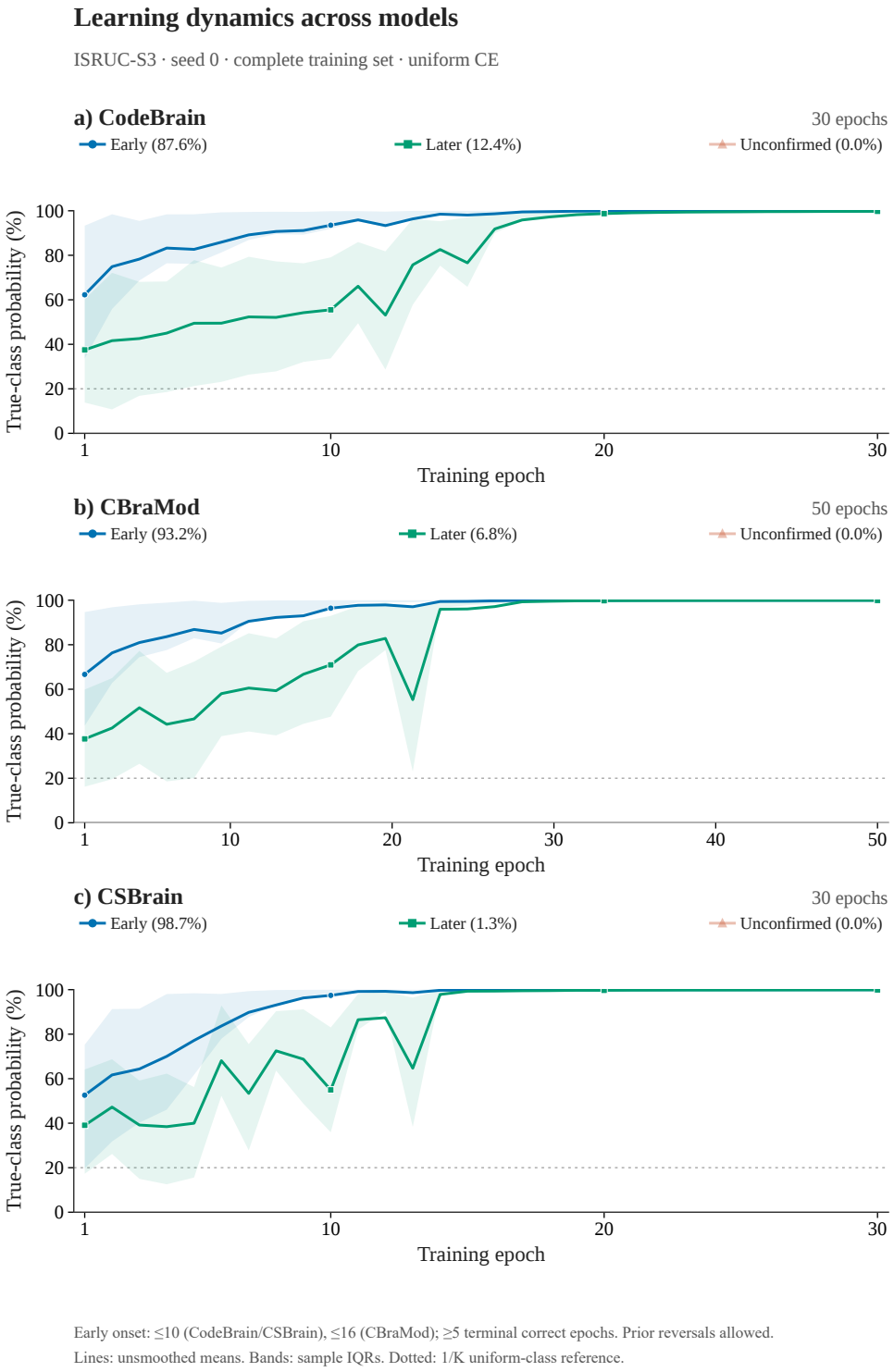

Figure 31: **ISRUC-S3: training responses across three models.** All 7,060 genuine S3 training sleep segments per model, five classes. CodeBrain and CSBrain use 30 ordinary-CE epochs; CBraMod uses 50. The five-epoch terminal-correctness rule is applied to every segment with its model-specific early boundary. Unconfirmed cases include final errors and correct terminal runs shorter than five epochs; the dotted reference is $1/5$.

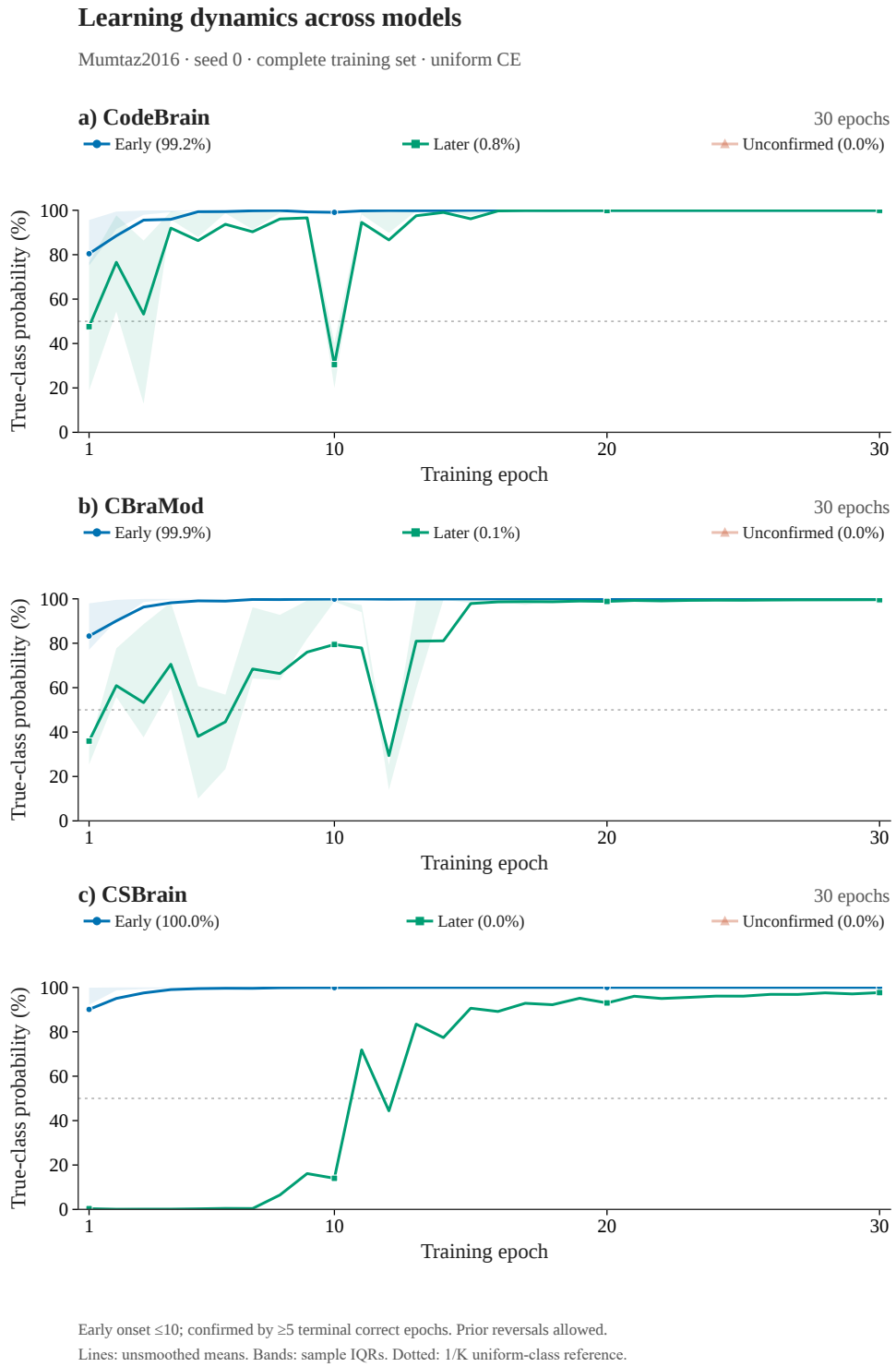

Figure 32: **Mumtaz2016: training responses across three models.** All 4,702 training trials per model, two classes, 30 epochs. Each trial contributes once to an early, later, or unconfirmed group. Sample-IQR bands preserve within-group heterogeneity, with a shared 0–100% probability axis across models.

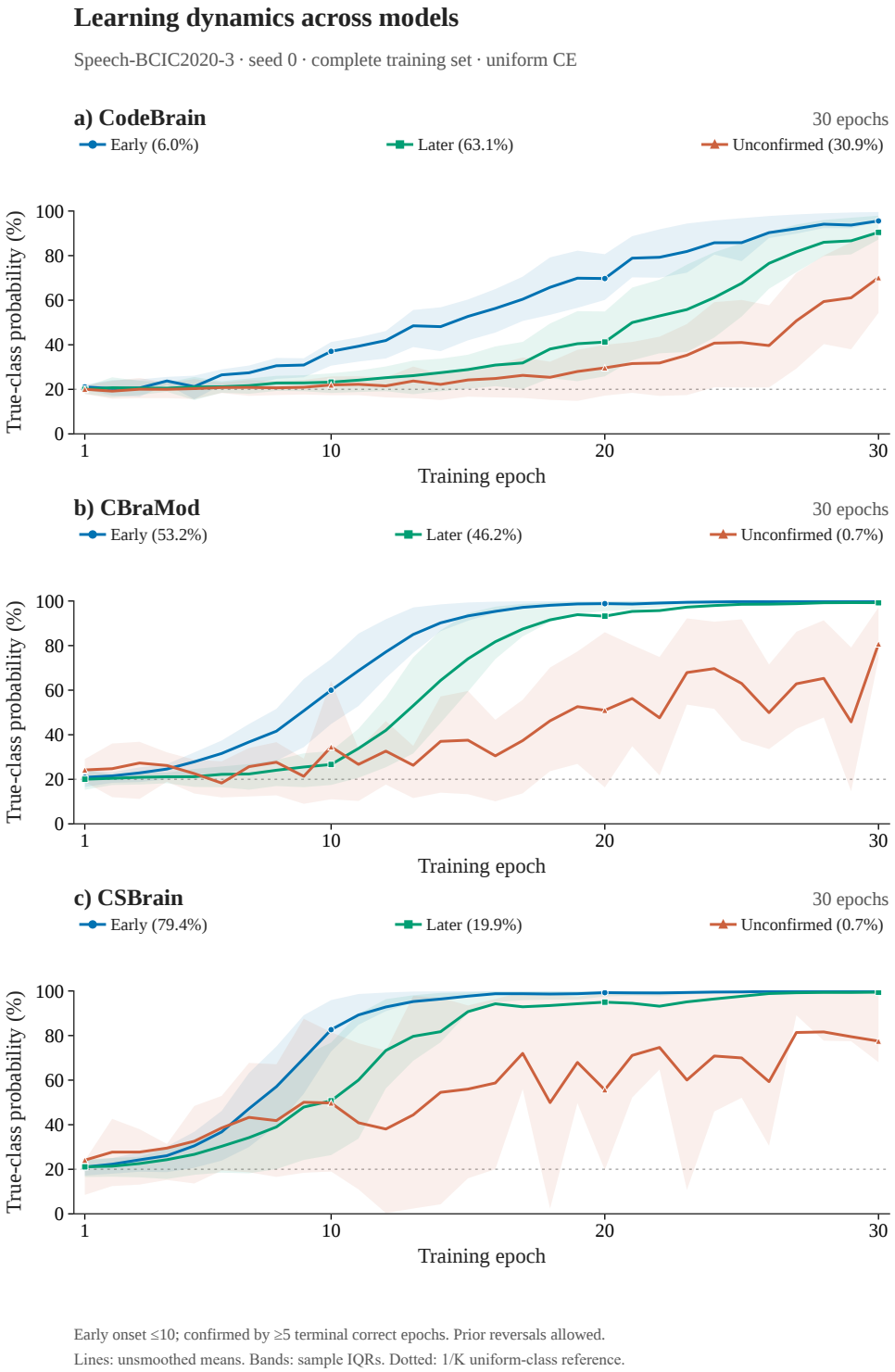

Figure 33: **Speech-BCIC2020-3: training responses across three models.** All 4,500 training trials per model, five classes, 30 epochs. Unsmoothed epoch-end probabilities are summarized under the same persistence rule. Legend counts include the entire training set; the dotted $1/5$ reference does not determine the predicted class.

L Decision-Structure Correlation Analysis

The main text uses online training feedback for Self-Guidance, with the aim of supporting representations that provide clearer Decision Evidence and Decision Trajectory. We examine empirical associations between the resulting sample weights and these two internal cues through sample-level analyses of online weights, paired spatial-evidence maps, and layer-wise trajectory scores. These analyses describe relationships under the available sample rankings and ablations; they do not establish that increasing a training statistic necessarily improves either internal cue. Concretely, we summarize sample-wise online weights during fine-tuning. The Evidence analysis compares 28 paired samples under the full and drop settings, measuring how strongly evidence concentrates on task-relevant motor-imagery regions. The matched-sample analysis relates online training weights to final true-class probability and layer-aggregated trajectory alignment scores. Together, these measurements relate the training feedback used by Self-Guidance to spatial reliance and cross-layer alignment patterns.

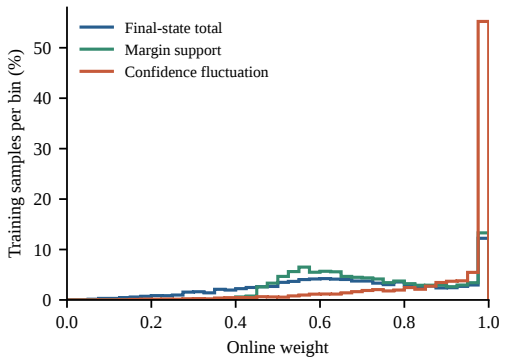

Figure 34: Online-weight factors for all 6,880 CodeBrain–Cho2017 training trials (seed 0).



Figure 35: Online-weight quintile analysis. Groups defined by training weights show corresponding differences in downstream trajectory scores and evidence concentration.

Figures 34 and 35 summarize the online-weight factors and matched-sample associations. The component view uses the complete Cho2017 training set; the separate quintile view examines how training weights relate to downstream support.

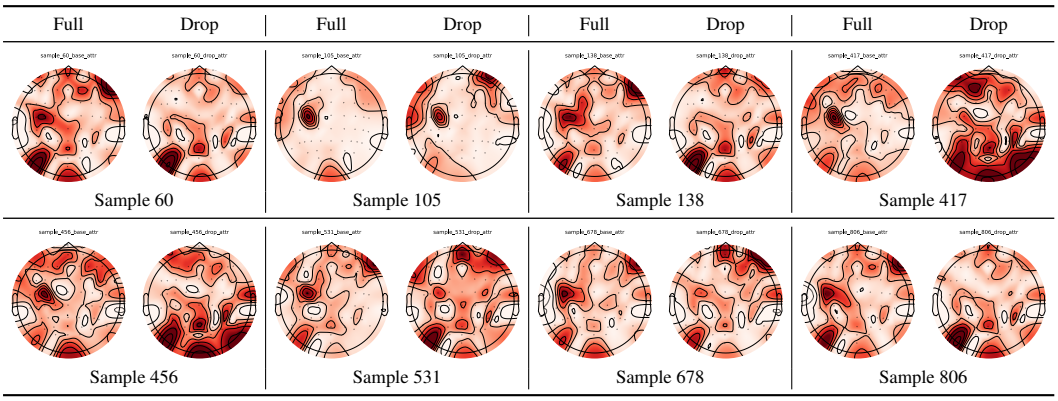

Figure 36: Representative Decision Evidence comparisons. Across 8 paired samples, the Full setting (with complete self-guidance) concentrates stronger evidence on task-relevant regions compared to the Drop setting.

Table 26: Quantitative summary of sample-level associations between Self-Guidance training feedback, Decision Evidence, and Decision Trajectory.

Question	Comparison	Observed pattern
Evidence quality and decision reliability	Full setting vs. drop setting on 28 paired samples. Evidence quality is measured by the C3/C5 ratio, the task-relevant motor-imagery evidence concentration.	Mean C3/C5 ratio drops from 0.1215 to 0.0667, and the ratio decreases in 28/28 pairs. Correct predictions change from 20/28 to 17/28.
Online training weight and downstream support	The 50 matched samples with both online-weight measurements and trajectory summaries are grouped by the total training weight.	The mean final true-class probability rises from 0.5005 in the lowest training-weight quintile to 0.6616 in the highest training-weight quintile; the Pearson correlation between total training weight and final true-class probability is 0.342.
Trajectory alignment score and margin-based weight	The same 50 matched samples are grouped by layer-aggregated trajectory alignment score and compared against the online margin component.	The lowest trajectory-score quintile has mean online-margin weight 0.7713, while the highest quintile has 0.7969, indicating an association between higher trajectory alignment scores and greater margin-based training weight in these samples.

Figure 36, Figures 34 and 35, and Table 26 summarize the observed relationships. When the full setting is replaced by the drop setting, the C3/C5 evidence concentration decreases for every paired sample, and the number of correct predictions also decreases. In the matched samples, larger online training weights are associated with stronger downstream true-class support, while the margin-based weight is associated with layer-aggregated trajectory alignment scores. These observations link Self-Guidance training feedback to spatial evidence and cross-layer alignment patterns in the analyzed settings.

M Efficiency and Complexity Analysis

M.1 Computational Overhead

We consider Self-Assessment and Self-Correction given precomputed internal decision features. Our computational analysis therefore reports the incremental cost of these two heads and excludes feature acquisition. Integrated Gradients serves as one implementation of the attribution input in our experiments. Table 27 lists single-sample backbone forward FLOPs as a reference alongside the estimated cached-head FLOPs.

Table 27: Cho2017-MI per-sample backbone forward FLOPs (PyTorch count) and estimated cached-head FLOPs. Feature precomputation is excluded.

Backbone	Base forward (G)	Cached heads (M)
CodeBrain	≈ 9.74	≈ 1.20
CSBrain	≈ 5.91	≈ 1.20
CBraMod	≈ 2.64	≈ 1.20

N Discussion

The empirical results suggest that decision-structure signals are useful across heterogeneous EEG tasks, but they also show where the method is most and least effective. The largest gains occur on

Mumtaz2016 and BCI-IV-2a; ISRUC-S3 shows a smaller positive gain under the current split. In high-conservatism settings such as Speech-BCIC2020-3, the correction head changes few samples and therefore produces smaller gains; this behavior is preferable to aggressive correction that harms reliable predictions.

The method also depends on the quality of calibration data. The self-assessment head learns to rank errors from observed calibration outcomes, so very small or distribution-shifted calibration sets may weaken risk ranking. The correction head is intentionally conservative, which reduces harmful flips but may leave some recoverable errors unchanged.

O LIMITATIONS, ETHICS, AND REPRODUCIBILITY

Limitations. GACE assumes access to internal EFM decision signals, including layer-wise representations or logits for Decision Trajectory and channel-level attribution for Decision Evidence. The framework also depends on calibration data for post-hoc risk assessment and correction; very small, noisy, or distribution-shifted calibration sets may weaken error ranking and reduce correction opportunities. The current evaluation covers multiple EEG task families and three EFM backbones, but the results should still be interpreted as research evidence for downstream EEG representation adaptation rather than as clinical validation.

Ethics and broader impacts. The experiments use previously released EEG datasets and do not collect new human-subject data. The potential positive impact is improved reliability analysis and adaptation for EEG foundation models across heterogeneous downstream tasks. The main risks are inappropriate clinical interpretation, privacy-sensitive use of EEG-derived models, and over-reliance on post-hoc correction in safety-critical settings. In particular, GACE should not be treated as a medical diagnostic system without dataset-specific clinical review and external validation.

O.1 CODE, DATA, AND REPRODUCIBILITY

Code and configuration availability. An anonymized supplementary code package is available at `https://anonymous.4open.science/status/GACE-AC85`. The package contains the implementation of GACE, dataset/backbone configuration files, scripts for running the main table and appendix diagnostics, and scripts for aggregating the reported metrics. The package does not redistribute raw EEG datasets or pretrained backbone checkpoints; instead, it documents how to obtain each external asset from the original source listed in Appendix O.1. After review, the anonymized package will be replaced by a public release.

Existing assets. Table 28 summarizes the sources and license or usage terms for the datasets and pretrained EEG foundation backbones used in this work. When no standard license name is available, we list the public source, dataset website, or cited source.

Table 28: Sources and usage terms for datasets and pretrained EEG backbones.

Asset	Source	License / usage terms
ATTENTION	Shin et al. simultaneous EEG–NIRS cognitive-task dataset; https://www.nature.com/articles/sdata20183; data page: http://doc.ml.tu-berlin.de/simultaneous_EEG_NIRS/	CC BY 4.0; CC0 metadata.
BCI-IV-2a	BNCI Horizon 2020 dataset 001-2014, BCI Competition IV dataset 2a; https://bnci-horizon-2020.eu/database/data-sets	CC BY-ND 4.0.
Cho2017-MI	Cho2017 motor imagery dataset, GigaDB DOI 10.5524/100295; https://moabb.neurotechx.com/docs/generated/moabb.datasets.Cho2017.html	CC BY 4.0.
ISRUC-S3	ISRUC-Sleep dataset, Subgroup 3; https://sleeptight.isr.uc.pt/	Open-source; https://sleeptight.isr.uc.pt/.
Mumtaz2016	figshare MDD Patients and Healthy Controls EEG Data, DOI 10.6084/m9.figshare.4244171; https://figshare.com/articles/dataset/EEG_Data_New/4244171	CC BY 4.0.
Speech-BCIC2020-3	2020 BCI Competition Track 3 imagined-speech dataset and associated OSF/NeMAR-style metadata; https://brain.korea.ac.kr/bci2020/competition.php	CC BY 4.0.
SEED	SJTU BCMI Lab SEED collection; https://bcmi.sjtu.edu.cn/home/seed/	SJTU SEED usage agreement.
CodeBrain	CodeBrain EEG foundation backbone (Ma et al., 2026).	Source paper.
CBraMod	CBraMod EEG foundation backbone (Wang et al., 2025a); public repository: https://github.com/wjq-learning/CBraMod.	MIT.
CSBrain	CSBrain EEG foundation backbone (Zhou et al., 2025c).	Source paper.

Reproduction commands. The main results can be reproduced from the supplementary package with the following entry points, after placing externally downloaded datasets and pretrained checkpoints under user-specified local roots:

```
conda env create -f environment.yml
conda activate gace

python scripts/prepare_data.py --dataset all --data_root
/path/to/eeg_data
bash scripts/run_main_table.sh --data_root /path/to/eeg_data \
    --ckpt_root /path/to/efm_checkpoints --seeds 0 1 2
bash scripts/run_appendix_diagnostics.sh --data_root
/path/to/eeg_data \
    --ckpt_root /path/to/efm_checkpoints --seeds 0 1 2
python scripts/summarize_results.py --result_root results/
```

The scripts write per-seed result files and a final summary table containing B-ACC, Kappa, weighted F1, and the corresponding Δ values against the matched fine-tuned EFM baseline.

Splits, preprocessing, and calibration. For each dataset–backbone pair, baseline and GACE use identical preprocessing, train/validation/test splits, and random seeds. When an official or commonly used benchmark split is available, we follow that split; otherwise, the split file included in the supplementary configuration fixes the train/validation/test partition before model training. Test samples are not used to fit the risk or correction heads; those heads are trained on calibration samples drawn from the training/validation side of the split. All main-table summaries use seeds $0, 1, 2$, and their reported standard deviations are sample standard deviations across those same three seeds under the matched dataset–backbone protocol.

Hyperparameters and implementation details. The supplementary configuration files specify, for every dataset–backbone pair, the input sampling rate and windowing policy, channel selection/order, batch size, optimizer type, learning rate, weight decay, number of fine-tuning epochs, early-stopping or model-selection criterion, calibration budget, high-risk selection rate, and Self-Assessment/Self-Correction head dimensions. Backbone fine-tuning uses the original backbone implementation and the same optimizer/hyperparameter policy for the baseline and GACE; the only difference is the addition of the self-guidance weights and the post-hoc self-assessment/self-correction heads. Checkpoint epochs for both post-hoc heads are selected using internal K-fold validation on the calibration set.

O.2 Compute Resources

All reported experiments were run as independent single-GPU jobs on a server equipped with 8 NVIDIA RTX 4090 GPUs.