SCOPE: STRUCTURED PROTOTYPE-GUIDED ADAPTATION FOR EEG FOUNDATION MODELS WITH LIMITED LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Electroencephalography foundation models (EFMs) have shown strong potential for transferable representation learning, but their real-world adaptation remains challenging when only a few labeled subjects are available. We trace this challenge to *a structural mismatch between noisy, limited supervision and the highly plastic parameter space of EFMs*, reflected in three key failure modes: overconfident miscalibration, transient optimization collapse, and disruption of pretrained representations under unconstrained updates. To address this mismatch, we propose *SCOPE*, a Structured COnfidence-aware Prototype-guided framework for label-limited EFM adaptation. *SCOPE* first constructs cohort-level external supervision to provide persistent guidance and further derives confidence-aware pseudo-labels to select reliable unlabeled samples. Building on this supervision, *SCOPE* introduces *ProAdapter*, a lightweight prototype-conditioned adapter that modulates frozen EFMs to preserve pretrained representations. We comprehensively evaluate *SCOPE* across 60 adaptation settings spanning 6 tasks, 5 EFM backbones, and 5%-100% labeled subject ratios. In label-limited settings, *SCOPE* outperforms full fine-tuning by 4.6 multiclass Kappa and 4.4 binary PRAUC points on average.

1 INTRODUCTION

Electroencephalography (EEG) is widely used to characterize brain states across clinical contexts, enabling sleep staging (Shuai et al., 2026), affective assessment (Sun et al., 2024), and noninvasive brain-computer interfaces (Ding et al., 2025b). Recent research has increasingly shifted from task-specific models (Ma et al., 2025a; 2026a) toward EEG foundation models (EFMs), which leverage large-scale pretraining to learn transferable representations across tasks (Zhou et al., 2025c; Wang et al., 2026a). In experimental settings, EFMs achieve strong performance via full fine-tuning, making it the dominant adaptation strategy in current EFM research (Xiong et al., 2026).

However, real-world EEG adaptation typically involves only a few labeled subjects alongside abundant unlabeled recordings, since reliable labels require expert subject-level annotation (Fiorillo et al., 2023). Despite strong benchmark performance, **EFMs exhibit clear generalization degradation under full fine-tuning in this label-limited regime**: as shown in Figure 1, training on 30% of subjects produces a substantially larger generalization gap than full supervision (Lea & Jones, 2024; Ma et al., 2025b). A closer look reveals that 1) the training kappa still converges to a level comparable to full supervision, and 2) the gap continues to widen with further training. This indicates that the capacity is more than sufficient and that additional optimization merely overfits the limited supervision rather than consolidating transferable representations. Therefore, the unreliability of full fine-tuning reflects ***a structural mismatch between limited supervision and the large, highly plastic parameter space of EFMs***, motivating us to revisit the problem of adapting EFMs under label-limited conditions, where several challenges are identified.

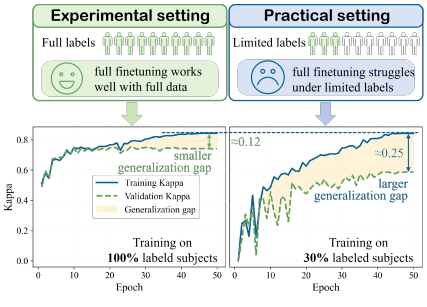

Figure 1: Full fine-tuning of CodeBrain on ISRUC shows a widening generalization gap under limited labels.

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.

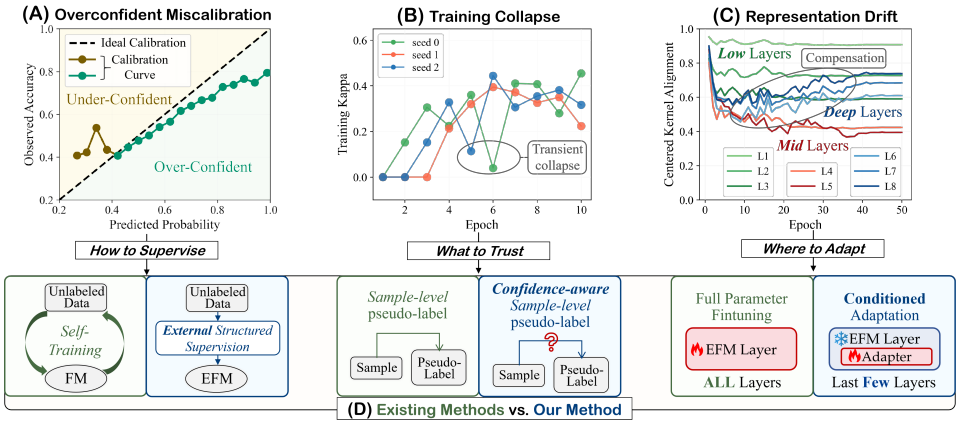

Figure 2: **Empirical motivation and method mapping.** All panels use *CodeBrain* on ISRUC. (A) Equal-width bins validation-set calibration curve, revealing overconfident miscalibration. (B) Training Kappa across seeds, revealing transient collapse where performance abruptly deteriorates. (C) Layer-wise CKA similarity to the pretrained initialization, showing subsequent compensatory drift in deeper layers as the model attempts to recover from this disruption. (D) These observations motivate our shift from self-training, sample-level pseudo-labels, and full fine-tuning toward external cohort-level structured supervision, confidence-aware pseudo-labeling, and conditioned adaptation.

C1) EFM overconfidence undermines reliable self-guidance. Self-training is widely used to leverage unlabeled data under limited supervision in foundation models (Xie et al., 2020; Zhang et al., 2025), assuming that model confidence reflects prediction accuracy (Sohn et al., 2020; Berthelot et al., 2020). However, under limited supervision, EFMs can be overconfident but inaccurate (Figure 2(A)), making self-generated supervision potentially misleading and prone to error amplification.

C2) EFMs are fragile under imperfect supervision. Pseudo-labeling is a common way to supervise unlabeled data (Wang et al., 2023), but EEG properties can make these labels unreliable: continuous brain states induce ambiguous class boundaries, while class imbalance biases learning toward dominant modes (Zhou et al., 2025b; Rizve et al., 2021). Unfortunately, EFMs are sensitive to supervisory bias, with limited-label full fine-tuning triggering transient optimization collapse (Figure 2(B)).

C3) Unconstrained fine-tuning risks overwriting EFM's pretrained representations. Under limited supervision, full fine-tuning causes deep-layer Centered Kernel Alignment (CKA) similarity to drop sharply and then partially rebound (Figure 2(C)), suggesting that unconstrained updates may disrupt useful pretrained representations and impair generalization to downstream tasks.

To address these challenges, we propose *SCOPE*, a **S**tructured **CO**nfidence-aware **P**rototype-guided adaptation framework for **E**EG foundation models. To the best of our knowledge, **this is the first framework that systematically studies how to adapt EEG foundation models under label-limited practical deployment.** *First*, we introduce an *external supervision construction* module that builds cohort-level structured prototypes to provide persistent guidance. *Second*, within the same module, we derive confidence-aware pseudo-labels to provide reliable sample-level supervision. *Third*, we propose *ProAdapter* to inject learned prototypes into deeper EFM layers via lightweight modulation while freezing the backbone to preserve pretrained knowledge. Our contributions are:

- **External Cohort-level Structured Prototypes.** We construct cohort-level prototypes with prior-guided inter-class geometry (Papyan et al., 2020) and balanced intra-class structure (Sinkhorn & Knopp, 1967; Caron et al., 2020), providing persistent external guidance for EFM adaptation.
- **Confidence-aware Sample-level Pseudo-labeling.** We derive externally guided pseudo-labels by measuring the agreement between two complementary geometric predictions, filtering out uncertain samples for reliable sample-level supervision.
- **Prototype-conditioned Adaptation.** We propose *ProAdapter* to align EFM adaptation with structured external supervision by injecting learned prototypes through lightweight modulation.
- **Comprehensive Evaluation**. We conduct a broad evaluation across **60 adaptation settings**, spanning 6 EEG tasks, 5 EFM backbones, and labeled-subject ratios from 5% to 100%, showing that *SCOPE* consistently performs strongly under limited labels while updating few parameters.

2 Related Work

Scalp EEG Foundation Models. Self-supervised pretraining has been widely used to learn transferable EEG representations, including contrastive (Kostas et al., 2021; Karantonis et al., 2025) and reconstruction-based methods (Chien et al., 2022; Mohammadi Foumani et al., 2024). Much of the work focuses on scalp EEG, enabled by standardized benchmarks and data availability (Wang et al., 2025a; Obeid & Picone, 2016; Xiong et al., 2026). Existing EFMs mainly adopt sequence modeling backbones, including Transformer family (Yang et al., 2023; Jiang et al., 2024a; Wang et al., 2024; Zhou et al., 2025d) and state-space models (SSMs) (Wang et al., 2025b; Ma et al., 2026b). Although EFMs are powerful, current practice relies on full fine-tuning, which is suboptimal under label-limited adaptation (Lee et al., 2025; Yang et al., 2026; Wang et al., 2026b).

Adaptation of EEG foundation models. Adapting EFMs under limited supervision requires constraining updates to preserve the pretrained knowledge. Useful design principles can be drawn from parameter-efficient fine-tuning (PEFT) in general foundation models, spanning weight-space approaches that confine updates to low-dimensional subspaces, feature-space modulation, and adapter-based techniques (Hu et al., 2022; Perez et al., 2018; Karimi Mahabadi et al., 2021). Related efforts in EFMs have pursued either parameter efficiency (Jeon et al., 2025; Choi et al., 2026) or data efficiency via informative sample selection (Liao et al., 2026). However, how to adapt EFMs under cross-subject, label-limited conditions remains underexplored. Detailed related work is provided in Appendix B.

3 Methodology

3.1 Problem Formulation and Framework Overview

Problem Formulation. Let $\mathbf{X} \in \mathbb{R}^{C\times T}$ denote an EEG signal with C channels and length T. The training data comprise a small labeled set $\mathcal{D}_l = \{(\mathbf{X}_i, y_i)\}_{i=1}^{N_l}$, where $y_i \in \{1, \ldots, K\}$ is the label, and a much larger unlabeled set $\mathcal{D}_u = \{\mathbf{X}_j\}_{j=1}^{N_u}$. Both sets share a label space but exhibit inter-subject distribution shifts, with test subjects unseen during training to ensure inductive adaptation.

Framework Overview. *SCOPE* follows a two-stage design: *(I) external supervision construction* and *(II) prototype-conditioned adaptation* (Figure 3). *Stage I* addresses **C1 (EFM's overconfidence undermines reliable self-guidance)** by constructing prototypes as persistent external structured supervision in Section 3.2.1; it further addresses **C2 (EFMs are fragile under imperfect supervision)** by filtering out untrustworthy pseudo-labels through confidence-aware selection in Section 3.2.2. *Stage II* addresses **C3 (Unconstrained fine-tuning risks overwriting EFM's pretrained representations)** by fully freezing the EFM and modulating only its deep layers conditioned on prototypes in Section 3.3. Section 3.4 then presents the training strategy and overall objective.

3.2 External structured supervision construction

To provide EFMs with reliable supervision, we progressively construct cohort-level structured prototypes and confidence-aware sample-level pseudo-labels externally.

3.2.1 Cohort-level Structured Prototype Construction

Technical Challenge. Prototypes need to be discriminative across classes while preserving fine-grained intra-class structure. However, ambiguous EEG boundaries and imbalanced intra-class modes make simple prototype construction prone to class mixing and dominant-mode collapse.
Design Rationale. We address this with hierarchical geometry. We first impose inter-class geometric regularization to obtain class-separable, cluster-friendly task-prior embeddings, and then learn multiple balanced prototypes per class to represent diverse intra-class patterns.

Task-Prior Induction via Inter-Class Geometry. A lightweight task-prior network (TPN) h_ψ is trained on the labeled subset $\mathcal{D}_l$, where h_ψ consists of a feature encoder f_ψ and a classification head. Given an input EEG sample $\mathbf{X}_i$, the encoder produces a task-specific embedding

$$\mathbf{z}_i = f_\psi(\mathbf{X}_i) = \mathcal{H}_{\text{sep}}\big(\mathcal{H}_{\text{spa}}(\mathcal{H}_{\text{temp}}(\mathbf{X}_i))\big), \tag{1}$$

where $\mathcal{H}_{\text{temp}}(\cdot)$, $\mathcal{H}_{\text{spa}}(\cdot)$, and $\mathcal{H}_{\text{sep}}(\cdot)$ are a temporal, depthwise spatial, and separable convolution block, respectively. For multi-class tasks, we impose an Equiangular Tight Frame (ETF)-based

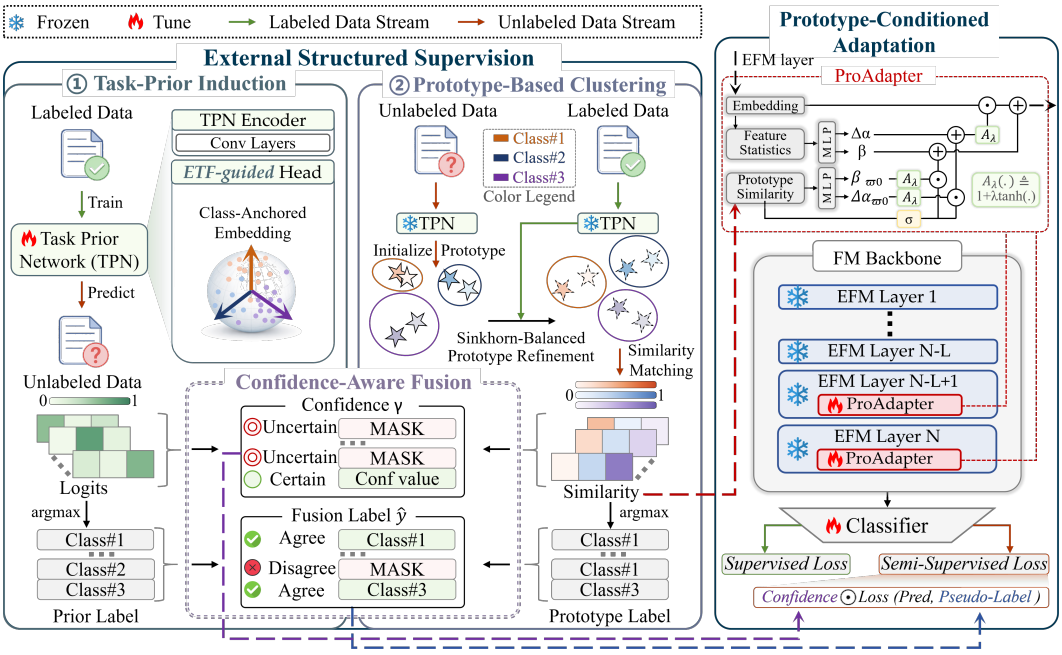

Figure 3: Overview of the two-stage *SCOPE* framework. **Left**: External structured supervision construction progressively induces class-level prototypes and confidence-aware pseudo-labels for unlabeled data. **Right**: A frozen EEG foundation model is adapted via lightweight prototype-conditioned adapters in the last layers, using confidence-weighted supervision for controlled adaptation.

geometric regularization to enforce maximal inter-class angular separation. The formal statement and proof are provided in Appendix A.1.

Let $W = [\mathbf{w}_1, \ldots, \mathbf{w}_K] \in \mathbb{R}^{d\times K}$ denote the weight matrix of the linear head. We normalize each class weight as $\tilde{\mathbf{w}}_k = \mathbf{w}_k/\|\mathbf{w}_k\|_2$ to obtain $\tilde{W} = [\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_K] \in \mathbb{R}^{d\times K}$ and define an ETF regularizer that penalizes deviations from a simplex equiangular tight frame:

$$\mathcal{L}_{\text{ETF}} = \left\| \tilde{W}^\top \tilde{W} - \left(\frac{K}{K-1} I - \frac{1}{K-1} \mathbf{1}\mathbf{1}^\top \right) \right\|_F^2. \tag{2}$$

where $\mathbf{1}\mathbf{1}^\top$ denotes an all-ones matrix. The overall training objective of the task-prior network is

$$\mathcal{L}_{\text{TPN}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{ETF}} \mathcal{L}_{\text{ETF}}, \tag{3}$$

where $\mathcal{L}_{\text{CE}}$ denotes the cross-entropy loss and λ_{ETF} controls the strength of geometric regularization. After training, the h_ψ is frozen and applied to the unlabeled set $\mathcal{D}_u = \{\mathbf{X}_j\}_{j=1}^{N_u}$. For each unlabeled sample $\mathbf{X}_j$, the encoder produces an embedding$\mathbf{z}_j = f_\psi(\mathbf{X}_j)$, which is mapped by the linear head to task-prior logits $\mathbf{s}_j = W^\top \mathbf{z}_j \in \mathbb{R}^K$. We then derive a prior label by $\hat{y}_j^{\text{prior}} = \arg\max_k \ \mathbf{s}_{jk}$.

Prototype Clustering via Intra-Class Variability. Given the inter-class structure induced by the TPN, we further introduce learnable prototypes to capture rich intra-class variability. For each task, we maintain a prototype bank $\mathcal{P}$ with M prototypes for each class $k \in \{1, \ldots, K\}$: $\mathcal{P} = \{p_{k,m} \in \mathbb{R}^d | m \in \{1, 2, ..., M\}\}$. The prototype bank $\mathcal{P}$ is initialized via k-means clustering on unlabeled TPN embeddings grouped by predicted prior labels. For a sample X_i and a prototype $p_{k,m}$, we compute their affinity using cosine similarity:

$$s_{i,(k,m)} = \left(\frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2} \right)^\top \left(\frac{\mathbf{p}_{k,m}}{\|\mathbf{p}_{k,m}\|_2} \right). \tag{4}$$

For class k, let $\mathcal{I}_k^\ell$ denote the index set of samples whose label is k in the current labeled batch, with $B_k = |\mathcal{I}_k^\ell|$. We define the class-wise similarity matrix $S^{(k)} \in \mathbb{R}^{B_k \times M}$ with entries $s_{i,m}^{(k)}$. We obtain

the soft sample-to-prototype assignment $Q^{(k)}$ by maximizing entropy-regularized similarity:

$$\max_{Q^{(k)}\geq 0} \sum_{i\in\mathcal{I}_k^\ell}\sum_{m=1}^{M} Q_{i,m}^{(k)}\big(s_{i,m}^{(k)} - \epsilon \log Q_{i,m}^{(k)}\big), \quad \text{s.t. } \sum_{m=1}^{M} Q_{i,m}^{(k)} = 1,\ \sum_{i\in\mathcal{I}_k^\ell} Q_{i,m}^{(k)} = \frac{B_k}{M}. \tag{5}$$

Here, $\epsilon > 0$ controls entropy regularization. These constraints enforce a balanced assignment of each class's samples across that class's prototypes. We approximate $Q^{(k)}$ using Sinkhorn-Knopp iterations on $\exp(S^{(k)}/\epsilon)$. We refine the prototype bank $\mathcal{P}$ using a cross-entropy loss on labeled samples:

$$\mathcal{L}_{\mathrm{P}} = \mathbb{E}_{k\in\mathcal{K}_{\mathrm{v}}}\, \mathbb{E}_{i\in\mathcal{I}_k^\ell}\, \mathrm{CE}(\mathbf{q}_i^{(k)},\ \sigma(\frac{\mathbf{s}_i^{(k)}}{\tau})), \tag{6}$$

where $\mathbb{E}$ is empirical expectation, τ is the softmax temperature, $\mathbf{s}_i^{(k)}, \mathbf{q}_i^{(k)} \in \mathbb{R}^M$ are the similarity vector to the class prototypes and the corresponding assignment, and $\mathcal{K}_{\mathrm{v}} = \{k \mid |\mathcal{I}_k^\ell| > 0\}$ is the set of classes represented in the current labeled mini-batch. The prototype label for each unlabeled sample is $\hat{y}_j^{\mathrm{proto}} = \arg\max_k \max_m s_{j,(k,m)}$.

3.2.2 Confidence-aware Sample-level Supervision

Technical Challenge. Direct pseudo-labeling can amplify errors under label-limited conditions, especially when confidence is estimated from a single prediction source. High-confidence logits may still reflect spurious biases rather than reliable sample-level supervision.
Design Rationale. We derive pseudo-labels from the agreement between two complementary sources: task-prior predictions, which emphasize inter-class separability, and prototype-based predictions, which capture intra-class variation.

We assign a pseudo-label $\hat{y}_j$ only when the prior prediction $\hat{y}_j^{\mathrm{prior}}$ agrees with the prototype prediction $\hat{y}_j^{\mathrm{proto}}$. We further quantify the reliability of agreed predictions. We normalize the probability distributions: $\mathbf{p}_j^{\mathrm{prior}}, \mathbf{p}_j^{\mathrm{proto}} = \mathrm{Softmax}(\mathbf{s}_j), \mathrm{Softmax}(\boldsymbol{\varpi}_j)$, where $\varpi_{j,k} = \max_{1\leq m\leq M} s_{j,(k,m)}$. We then combine the two distributions through a normalized element-wise product:

$$p_{j,c}^{\mathrm{fuse}} = \frac{p_{j,c}^{\mathrm{prior}}\, p_{j,c}^{\mathrm{proto}}}{\sum_{r=1}^{K} p_{j,r}^{\mathrm{prior}}\, p_{j,r}^{\mathrm{proto}}}. \tag{7}$$

This product favors classes jointly supported by both predictors. It corresponds to the Dempster-Shafer combination rule when belief mass is assigned exclusively to singleton classes, with no explicit ignorance mass (Denœux, 2000); the formal statement and derivation are provided in Appendix A.2.

The confidence of the pseudo-label is then quantified using an entropy-based confidence score:

$$\gamma_j = 1 - \frac{-\sum_{c=1}^{K} p_{j,c}^{\mathrm{fuse}} \log p_{j,c}^{\mathrm{fuse}}}{\log K}, \tag{8}$$

where K denotes the number of classes. An unlabeled sample $\mathbf{X}_j$ is selected for adaptation only if its confidence score γ_j exceeds a predefined threshold ρ.

3.3 Prototype-Conditioned Adaptation

Technical Challenge. The key challenge is to inject prototype-level structure into EFM adaptation without overwriting pretrained representations. Prototype-based pseudo-label selection provides loss-level supervision, but cannot directly constrain intermediate feature adaptation.
Design Rationale. We condition lightweight adapters on sample-specific prototype similarity. This allows prototype geometry to modulate intermediate features, providing persistent structural guidance while keeping the EFM backbone frozen.

Building upon the constructed supervision, we propose a *Prototype-Conditioned Adapter (ProAdapter)*. For each unlabeled sample $\mathbf{X}_j \in \mathcal{D}_u$, we use its prototype similarity vector $\boldsymbol{\varpi}_j$ to condition lightweight adaptation modules inserted into a frozen EFM.

Architecture. *ProAdapter* adopts a feature-wise modulation design to adapt intermediate representations in a parameter-efficient and architecture-agnostic manner. It operates on residual features when

residual connections are available, and otherwise modulates layer outputs. In practice, *ProAdapter* is inserted into the last L layers of the backbone.

Prototype-Conditioned Modulation. Given a prototype similarity $\boldsymbol{\varpi}_j \in \mathbb{R}^K$ and an intermediate representation $\mathbf{h}_l \in \mathbb{R}^{D\times T}$, *ProAdapter* uses feature-wise modulation:

$$\mathrm{ProAdapter}(\mathbf{h}_l; \boldsymbol{\varpi}_j) = \boldsymbol{\alpha}_j \odot \mathbf{h}_l + \boldsymbol{\beta}_j, \tag{9}$$

where $\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j \in \mathbb{R}^D$ are channel-wise scale and shift parameters, broadcast along temporal dimension. Modulation parameters are generated in two branches. First, statistics from $\mathbf{h}_l$ form a self-conditioning vector $\mathbf{c}_j = [\boldsymbol{\mu}_l, \boldsymbol{\sigma}_l] = \big[\mathrm{Mean}_t(\mathbf{h}_l),\ \mathrm{Std}_t(\mathbf{h}_l)\big] \in \mathbb{R}^{2D}$ to predict base modulation parameters:

$$[\Delta\boldsymbol{\alpha}_j^{(0)}; \boldsymbol{\beta}_j^{(0)}] = Linear(\mathbf{c}_j) \in \mathbb{R}^{2D}. \tag{10}$$

In parallel, the prototype similarity $\boldsymbol{\varpi}_j$ produces an additional modulation term whose magnitude is explicitly bounded and gated to ensure stable adaptation:

$$\begin{aligned} &[\Delta\boldsymbol{\alpha}_j^{(\varpi)}; \boldsymbol{\beta}_j^{(\varpi)}] = \lambda_\varpi \tanh\big(Linear(\boldsymbol{\varpi}_j)\big), \qquad g_j = \sigma\big(Linear(\boldsymbol{\varpi}_j)\big), \\ &(\Delta\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j, \boldsymbol{\alpha}_j) = \big(\Delta\boldsymbol{\alpha}_j^{(0)} + g_j\Delta\boldsymbol{\alpha}_j^{(\varpi)}, \boldsymbol{\beta}_j^{(0)} + g_j\boldsymbol{\beta}_j^{(\varpi)}, \mathbf{1} + \lambda\tanh(\Delta\boldsymbol{\alpha}_j)\big). \end{aligned} \tag{11}$$

where λ_ϖ and λ bound the prototype modulation and the final scale deviation from one. The modulated representation is then forwarded to the next layer.

3.4 Training Strategy and Overall Objective

We adopt a warm-up strategy that starts with supervised training on $\mathcal{D}_l$, and gradually incorporates unlabeled data for semi-supervised updates on $\mathcal{D}_u$. Each labeled mini-batch is paired with a fixed number of unlabeled samples to prevent pseudo-label supervision from dominating early training. Unlabeled data are sampled cyclically, so an epoch does not necessarily iterate over $\mathcal{D}_u$.

The semi-supervised loss uses sample-wise confidence weighting. For each unlabeled sample $\mathbf{X}_j \in \mathcal{D}_u$, a pseudo-label $\hat{y}_j$ and a confidence score $\gamma_j \in (0, 1)$ are obtained from the supervision construction, and an agreement-confidence mask m_j is applied. The overall training objective is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{sup}} + \mathbb{E}_{\mathbf{X}_j\sim\mathcal{D}_u}\ m_j\,\gamma_j\ \mathrm{CE}(\mathcal{F}_{\mathrm{EFM}}(\mathbf{X}_j),\ \hat{y}_j)\,, \tag{12}$$

where $CE(\cdot,\cdot)$ denotes the cross-entropy loss, and $\mathcal{L}_{\mathrm{sup}}$ is computed on labeled samples in $\mathcal{D}_l$.

4 Experiments

4.1 Datasets

We evaluate our method on 6 public EEG datasets: **ISRUC** (Khalighi et al., 2016) for five-class sleep staging, **SEED** (Zheng & Lu, 2015) for binary emotion recognition, **Mental Arithmetic** (Zyma et al., 2019) for binary mental workload assessment, **PhysioNet-MI** (Schalk et al., 2004) for four-class motor imagery, **SHU-MI** (Ma et al., 2022) for binary motor imagery, and **CHB-MIT** (Shoeb, 2009) for binary seizure detection. For all datasets, we follow prior EFMs settings under a strict cross-subject protocol (Wang et al., 2025a). We keep the original split unchanged, with 30% of training subjects treated as labeled and the remaining subjects used as unlabeled data. Validation and test sets consist of held-out subjects unseen during training, mimicking realistic label-limited EEG adaptation scenarios. Additional dataset details and split are provided in Appendix E.

4.2 Experiment Settings

Experiment Setup. We evaluate our *SCOPE* framework as a plug-and-play adapter on multiple pretrained EFMs. Training is performed in two stages. In the first stage, we construct supervision using only the training data: labeled samples provide direct supervision, while unlabeled samples are used without labels to induce task priors, class prototypes, and confidence-aware pseudo-labels. In the second stage, all backbone parameters of pretrained EFMs are frozen, and only the proposed *ProAdapter* is optimized. Notably, the validation set is used only for model selection in the second stage and is not involved in any part of the first stage, while the test set is strictly held out throughout.

Table 1: Baseline comparisons under the limited-label setting (30% labeled training subjects).

Backbone	Method	ISRUC		SEED		Mental		PhysioNet-MI		SHU-MI		CHB-MIT	
		Kappa	WF1	PRAUC	AUC	PRAUC	AUC	Kappa	WF1	PRAUC	AUC	PRAUC	AUC
Non-Pretrained	EEGNet	45.40	52.02	53.39	54.15	40.35	63.75	34.20	50.82	51.60	53.02	16.21	69.26
	EEGConformer	47.70	53.93	54.31	53.48	38.92	62.31	34.72	51.38	52.24	53.58	18.38	71.55
Transformer-based backbones													
LaBraM*	Frozen	48.58	58.35	72.45	73.89	60.83	79.25	22.81	41.80	61.81	63.49	30.53	86.30
	+LoRA	55.61	65.05	67.51	57.28	64.47	54.16	22.79	41.69	61.55	63.12	32.16	86.96
	+TASTEFUL	52.16	62.21	64.03	64.12	62.12	77.14	29.05	46.83	50.43	50.36	33.34	87.55
	+FixMatch †	33.14	43.83	74.08	**77.32**	53.92	74.41	41.50	55.64	49.15	49.62	29.97	87.00
	+FineSSL †	54.24	63.76	72.99	73.78	53.12	60.72	8.69	26.50	61.59	63.11	33.66	86.03
	+SCOPE (Ours) †	**56.19**	**65.15**	**74.25**	75.23	**64.64**	**82.93**	**41.80**	**56.33**	**63.33**	**64.21**	**36.05**	**89.13**
	Full finetuning	51.26	61.73	66.84	66.25	45.49	70.72	38.31	52.73	62.96	63.29	33.62	87.25
CBraMod	Frozen	60.34	67.91	57.83	58.74	47.19	63.53	30.73	47.60	54.29	54.64	28.82	87.11
	+LoRA	63.16	70.40	62.59	62.09	43.14	70.45	32.97	49.46	62.05	61.83	23.55	86.15
	+TASTEFUL	60.76	68.25	64.95	**64.73**	41.76	62.72	36.74	52.94	51.31	50.57	33.43	87.61
	+FixMatch †	20.42	23.23	60.21	59.16	46.70	66.85	23.32	40.15	51.14	51.04	30.78	87.53
	+FineSSL †	53.46	61.21	57.84	58.42	41.51	64.43	20.90	33.38	51.29	51.69	33.64	88.92
	+SCOPE (Ours) †	**63.29**	**70.74**	**65.57**	63.71	**54.44**	**74.39**	**37.84**	**53.86**	**66.31**	**67.80**	**35.02**	**89.04**
	Full finetuning	61.95	69.78	64.12	63.39	47.35	69.91	33.16	49.31	64.85	66.63	28.59	85.68
CSBrain	Frozen	56.27	64.98	56.27	56.57	38.30	66.68	30.55	47.50	58.31	58.96	20.54	87.08
	+LoRA	60.30	68.56	56.98	57.45	33.78	63.24	37.81	53.25	56.17	57.51	18.29	87.17
	+TASTEFUL	59.99	68.56	58.38	57.34	**56.56**	**76.12**	25.98	44.72	51.00	51.44	34.24	87.90
	+FixMatch †	44.53	53.48	59.71	57.34	43.24	69.13	27.17	45.49	51.68	51.48	32.37	87.76
	+FineSSL †	47.65	54.61	53.46	53.86	36.41	66.35	7.05	22.16	51.61	52.31	34.21	87.62
	+SCOPE (Ours) †	**61.51**	**69.86**	**63.05**	**62.09**	47.60	71.09	**40.62**	**55.79**	**60.99**	**61.58**	**34.96**	**89.21**
	Full finetuning	60.97	69.19	57.54	57.40	40.08	67.39	37.33	53.20	58.02	58.80	33.90	86.85
SSM-style backbones													
EEGMamba	Frozen	60.73	67.94	58.27	57.30	37.19	64.98	30.80	47.90	56.37	56.92	30.09	87.38
	+LoRA	62.78	69.79	58.64	57.35	38.12	65.45	33.62	50.39	57.11	57.35	32.61	86.53
	+TASTEFUL	49.11	57.75	52.49	52.74	34.49	62.87	16.56	35.30	50.81	51.11	**33.91**	88.49
	+FixMatch †	32.00	44.07	58.86	57.67	42.61	67.61	28.76	45.45	53.38	53.31	24.82	85.31
	+FineSSL †	54.02	65.92	48.98	47.64	39.11	66.86	4.37	16.81	54.06	54.83	32.06	85.50
	+SCOPE (Ours) †	**63.07**	**71.41**	**60.65**	**59.02**	**49.46**	**68.48**	**35.75**	**52.06**	**57.39**	**57.46**	33.39	**88.94**
	Full finetuning	62.02	68.71	59.84	57.98	42.89	65.36	33.58	50.17	57.09	57.22	31.06	85.41
CodeBrain	Frozen	52.54	61.75	63.44	63.93	46.70	71.25	26.05	43.80	58.96	58.30	28.37	86.45
	+LoRA	44.08	53.37	63.83	58.61	44.05	68.49	37.11	53.09	58.01	58.66	15.47	84.60
	+TASTEFUL	47.13	56.44	61.96	63.31	29.86	59.73	20.77	40.58	52.42	52.78	29.42	88.33
	+FixMatch †	57.23	66.74	61.58	60.58	44.53	69.30	21.01	40.20	55.52	54.96	28.81	85.85
	+FineSSL †	48.94	57.50	63.72	62.85	47.14	71.25	10.67	28.87	57.69	57.60	35.08	87.67
	+SCOPE (Ours) †	**63.46**	**71.11**	**65.88**	**65.30**	**49.30**	**72.40**	**37.42**	**53.24**	**60.56**	**60.32**	**36.63**	**88.77**
	Full finetuning	58.46	67.05	64.30	63.27	47.10	72.13	36.05	51.90	57.46	58.27	28.33	87.60

* LaBraM includes SEED in pretraining. † Methods also use unlabeled data for semi-supervised training.

Across all backbone models, we adopt the same task-specific classifier head used in the original EFMs to ensure a fair comparison. Experiments are conducted on 40GB NVIDIA A100 GPUs. All models are trained using AdamW with a batch size of 64 and backbone-specific learning rates following their original setups. We report the mean performance over five random seeds, and provide the standard deviations in Appendix E. For all backbone models, *ProAdapter* is inserted into the last three layers. We modulate transformer layer outputs and insert *ProAdapter* within residual layers for SSM-style backbones. Additional hyperparameter details are provided in Appendix K.

Backbone and Baselines. We evaluate *SCOPE* on 5 publicly available EFMs, spanning different architectural designs. Specifically, we consider 3 transformer-based backbones (**LaBraM** (Jiang et al., 2024a), **CBraMod** (Wang et al., 2025a), and **CSBrain** (Zhou et al., 2025d)) and 2 SSM-style backbones (**EEGMamba** (Wang et al., 2025b) and **CodeBrain** (Ma et al., 2026b)).

Task-specific baselines include **EEGNet** (Lawhern et al., 2018) and **EEGConformer** (Song et al., 2023). For foundation-model backbones, frozen and full fine-tuning serve as reference settings. We further compare two parameter-efficient adaptation methods: general-purpose **LoRA** (Hu et al., 2022) and EFM-specific **TASTEFUL** (Jeon et al., 2025). Self-training baselines include **FixMatch** (Sohn et al., 2020) and **FineSSL** (Gan & Wei, 2024). Additional details are provided in Appendix F.

Table 2: Ablation study on ISRUC with labels from 30% of the training subjects.

			CSBrain *(Transformer-based)*				**CodeBrain** *(SSM-Style)*			
			Kappa	Δ%	WF1	Δ%	Kappa	Δ%	WF1	Δ%
Pseudo-label Loss	Prototype Conditioning	ProAdapter	*Ablation of Framework Components*							
×	×	×	56.27±1.49	↓8.52	64.98±1.16	↓6.99	52.54±1.40	↓17.21	61.75±1.25	↓13.16
×	×	✓	58.30±1.63	↓5.22	66.74±1.51	↓4.47	59.01±1.45	↓7.01	66.78±1.03	↓6.09
×	✓	✓	59.42±1.04	↓3.40	67.65±1.35	↓3.16	61.32±1.42	↓3.37	68.50±1.28	↓3.67
✓	×	×	58.49±0.76	↓4.91	66.52±0.78	↓4.78	56.96±0.52	↓10.24	64.46±0.77	↓9.35
✓	×	✓	58.66±1.35	↓4.63	66.59±1.10	↓4.68	59.60±1.33	↓6.08	66.97±1.46	↓5.82
			Ablation on Training Strategy							
	w/o Warm-up †		59.29±0.62	↓3.61	67.46±0.77	↓3.44	61.84±1.05	↓2.55	69.12±1.26	↓2.80
	Sequential †		59.20±1.16	↓3.76	67.67±1.21	↓3.13	58.28±1.08	↓8.16	66.53±1.35	↓6.44
	Two-Stage †		58.31±0.45	↓5.20	66.42±0.69	↓4.92	57.10±1.29	↓10.02	64.60±1.10	↓9.15
	SCOPE (Ours) †		**61.51**±0.55	–	**69.86**±0.69	–	**63.46**±0.98	–	**71.11**±0.74	–

Evaluation Metric. Cohen's Kappa and weighted F1 are reported for multiclass tasks, while AUROC and PRAUC are used for binary tasks. Model selection is based on Kappa or AUROC, and the best checkpoint is used for test evaluation. Additional metric definitions are provided in Appendix D.

4.3 Performance Comparison

Table 1 summarizes results under the limited-label setting, where the best results are highlighted in **bold**, the second-best results are underlined, and our method is color-coded. Across all five backbones and six datasets, our method consistently achieves strong performance. For example, compared with frozen backbones, on ISRUC we improve Kappa by +7.61 with LaBraM and +10.92 with CodeBrain; on SEED, we improve PRAUC by +6.78 with CSBrain; and on Mental Arithmetic, we improve PRAUC by +12.27 with EEGMamba. These gains are consistently observed across Transformer-based and SSM-style backbones, indicating that *SCOPE* generalizes well across different architectural families under cross-subject distribution shifts.

A closer comparison shows that full fine-tuning can underperform the frozen backbone when the backbone has already seen the target dataset during pretraining, as with LaBraM on SEED. LoRA can fall below the frozen baseline, as observed with CodeBrain on ISRUC. Similarly, classical semi-supervised methods exhibit dataset- and backbone-dependent failures, such as FineSSL on PhysioNet-MI and FixMatch with the LaBraM backbone on CHB-MIT. In contrast, *SCOPE* shows more stable performance across backbone-dataset combinations.

> ***Takeaway 1.*** Adapting EFMs under limited subjects cannot be reliably addressed by directly applying generic methods since they can be sensitive to backbone-dataset combinations.

We further evaluate label-ratio sensitivity on ISRUC with all five backbones, using 5%, 10%, 20%, 50%, 75%, and 100% labeled training subjects. While our primary focus is the label-limited regime, we additionally include higher-label settings to characterize the regime where *SCOPE* remains advantageous over full fine-tuning. Detailed results are provided in Appendix G.1. Notably, these gains are obtained with only 2-5% trainable parameters and substantially reduced training time and GPU memory. Detailed statistics are reported in Appendix H. We further show in Appendix C that *SCOPE* alleviates the failure modes of full fine-tuning under limited supervision observed in Figure 2.

4.4 Ablation Study

We conduct ablation studies on the ISRUC dataset with 30% labeled training subjects using *CSBrain* and *CodeBrain* as representative backbones of different architectures. Table 2 summarizes the results.

1) Ablation of Framework Components. *SCOPE* contains two main components: external supervision construction and ProAdapter, with the learned prototypes serving as the interface between

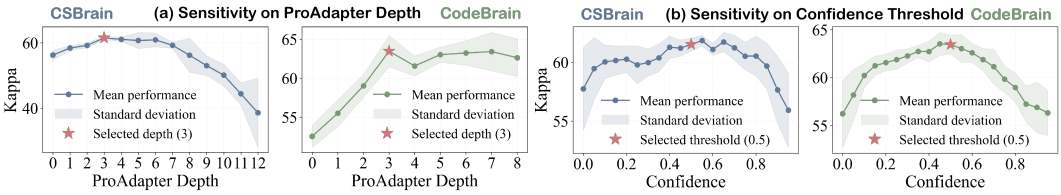

Figure 4: Sensitivity analysis on *ProAdapter* depth and confidence threshold.

them. External supervision is consumed through two pathways: the pseudo-label loss and prototype conditioning. Since prototype conditioning is only defined within ProAdapter, the three factors form six valid configurations, where disabling all three corresponds to the frozen-backbone baseline. Starting from this baseline, adding **ProAdapter** improves performance, especially on CodeBrain, while further introducing **prototype conditioning** reduces the gap to 2-4%. Independently, the **pseudo-label loss** also improves over the frozen baseline, but combining it with ProAdapter remains inferior to the full model. The complete *SCOPE* achieves the best results, indicating that *loss-level pseudo-label supervision and feature-level prototype conditioning provide complementary guidance*.

2) Ablation on Training Strategy. Removing the supervised warm-up (**w/o Warm-up**) degrades performance, showing that initialization is necessary. The **Sequential** strategy processes labeled samples before unlabeled ones within each epoch and further reduces performance, indicating that pseudo-labeled updates require anchoring by labeled supervision. The **Two-Stage** strategy, which trains on labeled data followed by unlabeled data only, causes the largest drop, suggesting that *benefit from pseudo-labeled data is limited when supervision is fully decoupled*.

4.5 Sensitivity Analysis

We analyze the sensitivity of the core design choices on ISRUC with 30% labeled training subjects using *CSBrain* and *CodeBrain*. Additional analyses on the *number of prototypes per class*, *ETF loss weight*, *pseudo-labeled data ratio*, and *pseudo-label quality* are provided in Appendix G and I.

1) Sensitivity to *ProAdapter* Depth. Figure 4(a) analyzes the sensitivity of *ProAdapter* to the adapter depth L. Performance exhibits a non-monotonic trend as L increases, first improving, then saturating, and potentially degrading as adaptation becomes deeper. The deeper Transformer-based *CSBrain* (12 layers) exhibits sharper degradation under excessive adaptation, whereas the shallower SSM-based *CodeBrain* (8 layers) remains more robust. We adopt a unified adapter depth of three layers.

> ***Takeaway 2:*** The optimal shallow adapter depth is consistent with our CKA analysis in Figure 2(C), suggesting that preserving lower and middle representations while adapting task-specific top layers better balances knowledge retention and task specialization.

2) Sensitivity to Confidence Threshold. We analyze the sensitivity to the confidence threshold ρ, where only unlabeled samples with confidence greater than ρ are included for adaptation. Figure 4(b) shows that both models exhibit a clear trade-off: low thresholds admit noise and degrade performance, whereas overly high thresholds restrict supervision to only easy samples and reduce data coverage.

5 Conclusion

We present *SCOPE*, a confidence-aware structured prototype-guided framework for adapting EEG foundation models under label-limited conditions. To the best of our knowledge, **this is the first systematic study on EEG foundation models adaptation under limited supervision**. Motivated by the observed failure modes of full fine-tuning, *SCOPE* constructs external cohort-level structured supervision, derives confidence-aware pseudo-labels, and injects prototype information into frozen EFMs through lightweight prototype-conditioned adapters. Across diverse EEG tasks, backbone architectures, and labeled-subject ratios, *SCOPE* consistently improves adaptation performance while maintaining efficiency. We hope this work provides a step toward more reliable, data-efficient, and practical deployment of EEG foundation models in real-world scenarios.

AI USE STATEMENT

We used generative AI tools to assist with language editing and LaTeX formatting.

ETHICS STATEMENT

This work studies label-limited adaptation of EEG foundation models using six public EEG datasets. The framework is intended for methodological research; clinical or other consequential applications would require additional validation, appropriate data governance, and human oversight.

REPRODUCIBILITY STATEMENT

Section 3 describes the two-stage *SCOPE* framework. Algorithm 1 summarizes the training procedure. Appendix E documents the datasets, preprocessing and subject-wise splits. Implementation settings and hyperparameters are provided in Appendix K. The main experiments report performance averaged over five random seeds. The codebase is available through the anonymized repository: `https://anonymous.4open.science/r/SCOPE-A0F2`.

REFERENCES

Ahmed Waleed Al-Asadi, Pedram Salehpour, and Hadi S Aghdasi. A robust semi-supervised deep learning approach for emotion recognition using eeg signals. *International Journal of Machine Learning and Cybernetics*, 15(10):4445–4458, 2024.

Irfan Al-Hussaini, Cao Xiao, M Brandon Westover, and Jimeng Sun. Sleeper: interpretable sleep staging via prototypes from expert rules. In *Machine Learning for Healthcare Conference*, pp. 721–739. PMLR, 2019.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*, 2020.

Donghong Cai, Junru Chen, Yang Yang, Teng Liu, and Yafeng Li. Mbrain: A multi-channel self-supervised learning framework for brain signals. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 130–141, 2023.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Junru Chen, Yang Yang, Tao Yu, Yingying Fan, Xiaolong Mo, and Carl Yang. Brainnet: Epileptic wave detection from seeg with hierarchical graph diffusion learning. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 2741–2751, 2022.

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*, 2026.

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.

Heejae Choi, Hyunseo Kim, and Eun Yi Kim. Pearl: Prompt-based eeg adaptation via resource-efficient learning for generalizable brain-computer interface. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6791–6795. IEEE, 2026.

Chenglong Dai, Jia Wu, Jessica J. M. Monaghan, Guanghui Li, Hao Peng, Stefanie I. Becker, and David McAlpine. Semi-supervised EEG clustering with multiple constraints. *IEEE Trans. Knowl. Data Eng.*, 35(8):8529–8544, 2023.

Thierry Denœux. A neural network classifier based on Dempster–Shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30(2):131–150, 2000.

Thierry Denœux. Quantifying prediction uncertainty in regression using random fuzzy sets: the ennreg model. *IEEE Transactions on Fuzzy Systems*, 31(10):3690–3699, 2023a.

Thierry Denœux. Reasoning with fuzzy and uncertain evidence using epistemic random fuzzy sets: General framework and practical models. *Fuzzy Sets and Systems*, 453:1–36, 2023b.

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*, 2025a.

Yidan Ding, Chalisa Udompanyawit, Yisha Zhang, and Bin He. Eeg-based brain-computer interface enables real-time robotic hand control at individual finger level. *Nature Communications*, 16(1): 1–20, 2025b.

Berkay Döner, Thorir Mar Ingolfsson, Luca Benini, and Yawei Li. Luna: Efficient and topology-agnostic foundation model for eeg signal analysis. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 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.

Zitao Fang, CHENXUAN LI, Hongting Zhou, 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.

Luigi Fiorillo, Giuliana Monachino, Julia van der Meer, Marco Pesce, Jan D Warncke, Markus H Schmidt, Claudio LA Bassetti, Athina Tzovara, Paolo Favaro, and Francesca D Faraci. U-sleep's resilience to aasm guidelines. *NPJ digital medicine*, 6(1):33, 2023.

Kai Gan and Tong Wei. Erasing the bias: fine-tuning foundation models for semi-supervised learning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 14453–14470, 2024.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*, 2022.

Ling Huang, Su Ruan, Pierre Decazes, and Thierry Denœux. Lymphoma segmentation from 3d pet-ct images using a deep evidential network. *International Journal of Approximate Reasoning*, 149: 39–60, 2022.

Ling Huang, Su Ruan, Yucheng Xing, and Mengling Feng. A review of uncertainty quantification in medical image analysis: Probabilistic and non-probabilistic methods. *Medical Image Analysis*, 97: 103223, 2024a.

Ling Huang, Yucheng Xing, Thierry Denoeux, and Mengling Feng. An evidential time-to-event prediction model based on gaussian random fuzzy numbers. In *International Conference on Belief Functions*, pp. 49–57. Springer, 2024b.

Ling Huang, Yucheng Xing, Qika Lin, Jinming Duan, Su Ruan, and Mengling Feng. Esurvfusion: An evidential multimodal survival fusion model based on epistemic random fuzzy sets. *IEEE Trans. Fuzzy Syst.*, 34(1):76–88, 2026.

Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, et al. Neftune: Noisy embeddings improve instruction finetuning. In *The Twelfth International Conference on Learning Representations*, 2024.

Jaehyun Jeon, Seungwoo Jeong, Yeajin Shon, and Heung-Il Suk. Parameter-efficient transfer learning for eeg foundation models via task-relevant feature focusing. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 1068–1077, 2025.

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*, 2024a.

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*, 2025.

Xue Jiang, Lubin Meng, Ziwei Wang, and Dongrui Wu. Deep source semi-supervised transfer learning (DS3TL) for cross-subject EEG classification. *IEEE Trans. Biomed. Eng.*, 71(4):1308–1318, 2024b.

Jin Jing, Aline Herlopian, Ioannis Karakis, Marcus Ng, Jonathan J Halford, Alice Lam, Douglas Maus, Fonda Chan, Marjan Dolatshahi, Carlos F Muniz, et al. Interrater reliability of experts in identifying interictal epileptiform discharges in electroencephalograms. *JAMA neurology*, 77(1): 49–57, 2020.

Antonis Karantonis, Konstantinos Barmpas, Dimitrios Adamos, Nikolaos Laskaris, Stefanos Zafeiriou, and Yannis Panagakis. Subject-aware contrastive learning for eeg foundation models. In *NeurIPS 2025 Workshop on Learning from Time Series for Health*, 2025.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in neural information processing systems*, 34:1022–1035, 2021.

Sirvan Khalighi, Teresa Sousa, José Moutinho Santos, and Urbano Nunes. Isruc-sleep: A comprehensive public dataset for sleep researchers. *Computer methods and programs in biomedicine*, 124: 180–192, 2016.

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.

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.

Andrew S Lea and David S Jones. Mind the gap—machine learning, dataset shift, and history in the age of clinical algorithms. *New England Journal of Medicine*, 390(4):293–295, 2024.

Na Lee, Konstantinos Barmpas, Yannis Panagakis, Dimitrios A. Adamos, Nikolaos A. Laskaris, and Stefanos Zafeiriou. Are large brainwave foundation models capable yet ? insights from fine-tuning. In *Forty-second International Conference on Machine Learning*, volume 267, 2025.

Zelin Liao, Yonghao Song, Chengjian Xu, Haochuan Zhang, and Qingqing Zheng. Eegtune: A data-efficient fine-tuning framework for eeg foundation models. *Pattern Recognition*, pp. 113591, 2026.

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*, 2026.

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, Qika Lin, Ziyu Jia, and Mengling Feng. St-usleepnet: A spatial-temporal coupling prominence network for multi-channel sleep staging. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 4182–4190, 2025a.

Jingying Ma, Jinwei Wang, Lanlan Lu, Yexiang Sun, Mengling Feng, Feifei Zhang, Peng Shen, Zhiqin Jiang, Shenda Hong, and Luxia Zhang. Development and validation of a dynamic kidney failure prediction model based on deep learning: A real-world study with external validation. *arXiv preprint arXiv:2501.16388*, 2025b.

Jingying Ma, Qika Lin, Feng Wu, Yucheng Xing, Ziyu Jia, and Mengling Feng. Logicsleep: a neurosymbolic-guided framework for explainable and personalized sleep staging. *npj Artificial Intelligence*, 2026a.

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*, 2026b.

Jun Ma, Banghua Yang, Wenzheng Qiu, Yunzhe Li, Shouwei Gao, and Xinxing Xia. A large eeg dataset for studying cross-session variability in motor imagery brain-computer interface. *Scientific Data*, 9:531, 2022.

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.

Hamid Niknazar and Sara C Mednick. A multi-level interpretable sleep stage scoring system by infusing experts' knowledge into a deep network architecture. *IEEE transactions on pattern analysis and machine intelligence*, 46(7):5044–5061, 2024.

Iyad Obeid and Joseph Picone. The temple university hospital eeg data corpus. *Frontiers in neuroscience*, 10:196, 2016.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Sen Qiu, Yongtao Chen, Yulin Yang, Pengfei Wang, Zhelong Wang, Hongyu Zhao, Yuntong Kang, and Ruicheng Nie. A review on semi-supervised learning for eeg-based emotion recognition. *Information Fusion*, 104:102190, 2024.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2021.

Sunny Sanyal, Hayden Prairie, Rudrajit Das, Ali Kavis, and Sujay Sanghavi. Upweighting easy samples in fine-tuning mitigates forgetting. In *Forty-second International Conference on Machine Learning*, 2025.

Gerwin Schalk, Dennis J McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R Wolpaw. Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on biomedical engineering*, 51(6):1034–1043, 2004.

Glenn Shafer. Dempster-shafer theory. *Encyclopedia of artificial intelligence*, 1:330–331, 1992.

Ali Hossam Shoeb. *Application of machine learning to epileptic seizure onset detection and treatment*. PhD thesis, Massachusetts Institute of Technology, 2009.

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.

Zitao Shuai, Zongzhe Xu, David Yang, Wei Wang, and Yuzhe Yang. Osf: On pre-training and scaling of sleep foundation models. In *Forty-third International Conference on Machine Learning*, 2026.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020.

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, 2023.

Jingnan Sun, Anruo Shen, Yike Sun, Xiaogang Chen, Yunxia Li, Xiaorong Gao, and Bai Lu. Adaptive spatiotemporal encoding network for cognitive assessment using resting state eeg. *npj Digital Medicine*, 7(1):375, 2024.

Dennis Tang, Jon Donnelly, Alina Jade Barnett, Lesia Semenova, Jin Jing, Peter Hadar, Ioannis Karakis, Olga Selioutski, Kehan Zhao, M Brandon Westover, et al. This eeg looks like these eegs: Interpretable interictal epileptiform discharge detection with protoeeg-knn. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 615–625. Springer, 2025.

Zheng Tong, Philippe Xu, and Thierry Denoeux. An evidential classifier based on dempster-shafer theory and deep learning. *Neurocomputing*, 450:275–293, 2021.

Shihao Tu, Linfeng Cao, Daoze Zhang, Junru Chen, Lvbin Ma, Yin Zhang, and Yang Yang. Dmnet: Self-comparison driven model for subject-independent seizure detection. *Advances in Neural Information Processing Systems*, 37:28254–28280, 2024.

Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

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, 2024.

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*, 2026a.

Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Bernt Schiele, and Xing Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Yihe Wang, Zhiqiao Kang, Bohan Chen, Yu Zhang, and Xiang Zhang. Benchmarking erp analysis: manual features, deep learning, and foundation models. *IEEE Transactions on Biomedical Engineering*, 2026b.

Weining Weng, Yang Gu, Shuai Guo, Yuan Ma, Zhaohua Yang, Yuchen Liu, and Yiqiang Chen. Self-supervised learning for electroencephalogram: A systematic survey. *ACM Computing Surveys*, 57(12):1–38, 2025.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. Noisytune: A little noise can help you finetune pretrained language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 680–685, 2022.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10687–10698, 2020.

Yucheng Xing, Ling Huang, Jingying Ma, Ruping Hong, Jiangdong Qiu, Pei Liu, Kai He, Huazhu Fu, and Mengling Feng. Dpsurv: Dual-prototype evidential fusion for uncertainty-aware and interpretable whole-slide image survival prediction. *arXiv preprint arXiv:2510.00053*, 2025.

Yucheng Xing, Pei Liu, Jingying Ma, Ruping Hong, Jiangdong Qiu, Tianyu Liu, Kai He, Ling Huang, and Mengling Feng. Bridging the modality bottleneck in pathology MIL through virtual molecular staining. *CoRR*, abs/2605.16392, 2026.

Wei Xiong, Jiangtong Li, Jie Li, Kun Zhu, and Changjun Jiang. Eeg-fm-bench: A comprehensive benchmark for the systematic evaluation and diagnostic analyses of eeg foundation models. In *Forty-third International Conference on machine learning*, 2026.

I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

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, Guoan Wang, Jingmei Yang, Alison Anderson, Dinesh Giritharan, Patrick Kwan, et al. Large-scale ai and foundation models for neuroscience: A comprehensive review. *Meta-Radiology*, pp. 100224, 2026.

Zhizhang Yuan, Fanqi Shen, Meng Li, Yuguo Yu, Chenhao Tan, and Yang Yang. Brainwave: A brain signal foundation model for clinical applications. *arXiv preprint arXiv:2402.10251*, 2024.

Daoze Zhang, Zhizhang Yuan, Junru Chen, Kerui Chen, and Yang Yang. Brant-x: A unified physiological signal alignment framework. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4155–4166, 2024.

Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12414–12424, 2021.

Ping Zhang, Zheda Mai, Quang-Huy Nguyen, and Wei-Lun Chao. Revisiting semi-supervised learning in the era of foundation models. In *Advances in Neural Information Processing Systems*, 2025.

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.

Huihui Zhou, Aiping Liu, Shizhen Ding, Jing Yao, and Xun Chen. An interpretable single-channel eeg sleep staging model based on prototype matching and multitask learning. *IEEE Sensors Journal*, 25(2):3782–3793, 2025a.

Songchi Zhou, Ge Song, Haoqi Sun, Deyun Zhang, Yue Leng, M Brandon Westover, and Shenda Hong. Continuous sleep depth index annotation with deep learning yields novel digital biomarkers for sleep health. *npj Digital Medicine*, 8(1):203, 2025b.

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 Process. Mag.*, 42(5):22–35, 2025c.

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*, 2025d.

Igor Zyma, Sergii Tukaev, Ivan Seleznov, Ken Kiyono, Anton Popov, Mariia Chernykh, and Oleksii Shpenkov. Electroencephalograms during mental arithmetic task performance. *Data*, 4(1):14, 2019.

APPENDIX CONTENTS

A Mathematical Justification of Design Components **18**
- A.1 Proof of Proposition A.1 . . . 18
- A.2 Proof of Lemma A.2 . . . 18

B Detailed Related Work **19**

C Failure Mode Analysis under Limited Supervision **20**
- C.1 Generalization Gap under Limited Supervision . . . 20
- C.2 Overconfident Miscalibration of EFM Backbones under Full-finetuning . . . 21
- C.3 Prediction Collapse under Limited Supervision . . . 22
- C.4 Layer-wise Representation Drift . . . 23

D Metrics **24**

E Dataset Description and Detailed Results **24**
- E.1 Sleep Staging . . . 25
- E.2 Emotion Recognition . . . 25
- E.3 Workload Assessment Mental Arithmetic . . . 26
- E.4 Multi-effector Motor Imagery . . . 27
- E.5 Hand Motor Imagery . . . 28
- E.6 Seizure Detection . . . 29

F Baseline and Backbone Models **29**
- F.1 Baseline Models . . . 29
- F.2 Backbone Models . . . 31

G Additional Ablation Studies on Design Choices **32**
- G.1 Ablation Study on Different Label Ratio . . . 32
- G.2 Ablation Study on ProAdapter Depth L . . . 37
- G.3 Ablation Study on Confidence Threshold ρ . . . 38
- G.4 Ablation Study on the number of prototype per class M . . . 40
- G.5 Ablation Study on ETF Loss Weights λ_{ETF} . . . 40
- G.6 Ablation Study on Pseudo-labeled Data Ratio . . . 41

H Efficiency Analysis **42**
- H.1 Backbone Adaptation Efficiency . . . 42
- H.2 Overall Computational Cost . . . 44

I Pseudo-Label Quality Analysis **45**

J Pseudocode for Training Procedure **49**

K Implementation Details **49**

L Code Access **51**

M Limitations **51**

A Mathematical Justification of Design Components

A.1 Proof of Proposition A.1

Proposition A.1. *Let $\{\tilde{\mathbf{w}}_k\}_{k=1}^K \subset \mathbb{S}^{d-1}$ be the normalized classifier weights. If they satisfy the simplex equiangular tight frame condition $\tilde{\mathbf{w}}_k^\top \tilde{\mathbf{w}}_{k'} = -\frac{1}{K-1}, \quad \forall\, k \neq k'$, the minimum pairwise angular separation is maximized.*

Proof. Since the cosine function is monotonically decreasing on $[0, \pi]$, this is equivalent to minimizing the maximum pairwise cosine similarity, defined as $\eta_{\max} = \max_{k \neq k'} \tilde{\mathbf{w}}_k^\top \tilde{\mathbf{w}}_{k'}$.

Consider the squared Euclidean norm of the sum of these vectors. By the non-negativity of the norm, we have:

$$\left\| \sum_{k=1}^K \tilde{\mathbf{w}}_k \right\|^2 = \sum_{k=1}^K \|\tilde{\mathbf{w}}_k\|^2 + \sum_{k \neq k'} \tilde{\mathbf{w}}_k^\top \tilde{\mathbf{w}}_{k'} \geq 0. \tag{13}$$

Since $\|\tilde{\mathbf{w}}_k\| = 1$ for all k, the first term sums to K.

$$K + \sum_{k \neq k'} \tilde{\mathbf{w}}_k^\top \tilde{\mathbf{w}}_{k'} \geq 0 \tag{14}$$

Then:

$$\sum_{k \neq k'} \tilde{\mathbf{w}}_k^\top \tilde{\mathbf{w}}_{k'} \geq -K \tag{15}$$

For the second term, in the summation $\sum_{k \neq k'} \tilde{\mathbf{w}}_k^\top \tilde{\mathbf{w}}_{k'}$, there are $K(K-1)$ terms (i.e., all off-diagonal elements). We can bound the sum using $\eta_{\max}$:

$$\sum_{k \neq k'} \tilde{\mathbf{w}}_k^\top \tilde{\mathbf{w}}_{k'} \leq \sum_{k \neq k'} \eta_{\max} = K(K-1)\eta_{\max}. \tag{16}$$

Substituting this into the expansion of the norm yields:

$$K(K-1)\eta_{\max} \geq -K. \tag{17}$$

Dividing both sides by $K(K-1)$ yields

$$\eta_{\max} \geq \frac{-K}{K(K-1)}. \tag{18}$$

Rearranging the inequality provides a lower bound on the maximum cosine similarity (known as the Rankin bound):

$$\eta_{\max} \geq -\frac{1}{K-1}. \tag{19}$$

For any set of K unit vectors, the largest cosine similarity among them is at least $-\frac{1}{K-1}$.

The proposition states that the vectors satisfy the simplex equiangular tight frame (ETF) condition: $\tilde{\mathbf{w}}_k^\top \tilde{\mathbf{w}}_{k'} = -\frac{1}{K-1}$ for all $k \neq k'$. In this configuration, the pairwise cosine similarity is uniform and exactly equals the theoretical lower bound. Consequently, the maximum cosine similarity is minimized, which implies that the minimum pairwise angular separation is maximized.

A.2 Proof of Lemma A.2

Lemma A.2. *Let $\Omega = \{\omega_1, \ldots, \omega_K\}$ be the set of K classes, and let m_1, m_2 be basic belief assignments supported only on singleton classes, with zero ignorance mass. Provided $\sum_{k=1}^K m_1(\{\omega_k\}) m_2(\{\omega_k\}) > 0$, Dempster–Shafer fusion reduces to the normalized product:*

$$m(\{\omega_c\}) = \frac{m_1(\{\omega_c\}) m_2(\{\omega_c\})}{\sum_{k=1}^K m_1(\{\omega_k\}) m_2(\{\omega_k\})}. \tag{20}$$

Proof. For $r \in \{1, 2\}$, singleton-only support means $m_r(A) = 0$ whenever $|A| \neq 1$, and $\sum_{k=1}^{K} m_r(\{\omega_k\}) = 1$. Dempster's combination rule for any nonempty $A \subseteq \Omega$ is

$$m(A) = \frac{1}{1 - \kappa_{\text{conf}}} \sum_{\substack{B,E \subseteq \Omega \\ B \cap E = A}} m_1(B) m_2(E), \tag{21}$$

where κ_{conf} denotes the conflict mass:

$$\kappa_{\text{conf}} = \sum_{\substack{B,E \subseteq \Omega \\ B \cap E = \emptyset}} m_1(B) m_2(E). \tag{22}$$

Since only singleton sets carry mass, the only nonzero contribution to $m(\{\omega_c\})$ comes from $B = E = \{\omega_c\}$. Hence,

$$\sum_{\substack{B,E \subseteq \Omega \\ B \cap E = \{\omega_c\}}} m_1(B) m_2(E) = m_1(\{\omega_c\}) m_2(\{\omega_c\}). \tag{23}$$

Likewise, the total non-conflicting mass is

$$1 - \kappa_{\text{conf}} = \sum_{k=1}^{K} m_1(\{\omega_k\}) m_2(\{\omega_k\}) > 0. \tag{24}$$

Substituting these identities into the combination rule gives the stated normalized product.

B Detailed Related Work

Brain Foundation Models In this work, we focus on scalp EEG foundation models as general-purpose encoders, targeting scalable and non-invasive EEG representation learning. Beyond Transformer- and SSM-based backbones, recent EEG foundation model studies have explored more flexible architectural designs, including modular routing with mixture-of-experts mechanisms (Fang et al., 2025; Chen et al., 2026). In addition to model architecture, recent efforts have also expanded EFM development along several complementary directions, such as scaling pretraining over larger and more diverse EEG corpora (El Ouahidi et al., 2025), improving robustness to heterogeneous electrode layouts (Döner et al., 2025), and injecting neuroscience-inspired inductive biases into representation learning (Ding et al., 2025a; Wang et al., 2026a). Other efforts combine Large Language Models (LLM) to model EEG (Jiang et al., 2025; Lu et al., 2025; Liu et al., 2026).

Beyond the rapid development of scalp EEG foundation models, related efforts have explored other derivative forms of EEG. For example, iEEG (intracranial electroencephalography) provides high signal fidelity and spatial resolution (Chen et al., 2022; Cai et al., 2023). Some recent studies have further explored unified representation learning across iEEG and other medical time series (Zhang et al., 2024; Yuan et al., 2024). These finer-grained physiological modalities are very important to specific downstream tasks, such as epileptic seizure detection and disease monitoring (Tu et al., 2024; Shoeibi et al., 2021).

Semi-supervised Learning Semi-supervised learning (SSL) aims to leverage abundant unlabeled data together with limited labeled samples, and has become a promising paradigm for learning under data-scarce settings (Van Engelen & Hoos, 2020). In the deep learning era, a series of effective SSL methods have been proposed, among which FixMatch (Sohn et al., 2020) and MixMatch (Berthelot et al., 2019) are representative approaches. These methods typically rely on lightweight backbone models and exploit consistency regularization and pseudo-labeling to achieve strong empirical performance. With the emergence of foundation models, SSL faces new challenges. Foundation models usually contain a large number of parameters, making full fine-tuning computationally expensive (Zhang et al., 2025). To address this issue, parameter-efficient tuning strategies, such as LoRA (Hu et al., 2022), have been widely adopted. Despite their strong zero-shot and few-shot capabilities, foundation models remain vulnerable to noisy pseudo-labels in semi-supervised settings, which can significantly degrade downstream performance (Yalniz et al., 2019; Xie et al., 2020).

In the EEG domain, semi-supervised learning has been studied for various applications, including emotion recognition, sleep staging, and brain-computer interface-related tasks (Qiu et al., 2024; Dai et al., 2023; Jiang et al., 2024b; Al-Asadi et al., 2024). However, most existing methods are designed for task-specific or relatively small-scale models (Weng et al., 2025). As research attention gradually shifts toward EEG foundation models (EFMs), the integration of semi-supervised learning with EFMs remains largely underexplored.

Prototype Learning Prototype-based metric learning methods, such as Prototypical Networks(Snell et al., 2017), are widely used for few-shot classification by representing each class with a prototype (typically the class centroid) in an embedding space and classifying queries by distance to these prototypes. Prototype learning is widely used in the fields of computer vision, such as ProDA (Zhang et al., 2021). In EEG analysis, prototype-based methods have also been explored to improve interpretability (Tang et al., 2025; Al-Hussaini et al., 2019; Zhou et al., 2025a; Niknazar & Mednick, 2024). However, unlike generic visual embeddings, EEG prototypes need domain-specific constraints. Overall, prototype-based learning provides a structured way to capture class-level information.

Dempster–Shafer theory Decision making under uncertainty remains a fundamental challenge, as modeling uncertainty and integrating heterogeneous evidence are inherently difficult. Dempster–Shafer (DS) theory (Shafer, 1992) extends probabilistic reasoning by allowing belief to be assigned to sets of hypotheses rather than only singleton outcomes, enabling explicit modeling of uncertainty and evidence conflict. A key component of DS theory, Dempster's rule of combination, provides a principled framework for aggregating multiple independent sources of evidence.

Early work introduced DS theory into neural networks by developing an adaptive evidence-based pattern classifier (Denœux, 2000). This pioneering work established a connection between evidential reasoning and neural network–based learning, enabling uncertainty-aware decision making beyond probabilistic outputs. Building upon this foundation, DS theory has since been extended to modern deep learning settings and applied to a variety of tasks, including image classification and segmentation (Tong et al., 2021; Huang et al., 2022; 2024a), where modeling uncertainty and handling ambiguous predictions are particularly important.

More recently, Dempster–Shafer theory has been extended to regression tasks through the introduction of Gaussian Random Fuzzy Numbers (GRFNs), which enable continuous-valued predictions while explicitly modeling epistemic uncertainty (Denœux, 2023b). Specifically, A DS-based evidential framework is proposed (Denœux, 2023a) for traditional regression problems, formulating predictions as fuzzy random variables with associated belief structures. Building upon this formulation, recent work further extended the framework to more complex time-to-event and survival analysis settings, demonstrating the applicability of DS-based regression under censored data and long-term outcome modeling (Huang et al., 2026; 2024b; Xing et al., 2025; 2026).

C Failure Mode Analysis under Limited Supervision

In this section, we revisit the failure modes of full fine-tuning under limited supervision identified in the Introduction, and analyze how *SCOPE* mitigates them through additional empirical evidence and comparative visualizations.

C.1 Generalization Gap under Limited Supervision

We first analyze the generalization behavior of full fine-tuning under limited labeled subjects. Figure 5 compares full fine-tuning and *SCOPE* on *CodeBrain* using 30% labeled subjects from ISRUC. For full fine-tuning, the training Kappa steadily increases and eventually reaches a high level, whereas the validation Kappa saturates much earlier and remains substantially lower. As a result, the generalization gap continuously widens during training. This suggests that the pretrained EFM has sufficient capacity to fit the limited labeled subjects, but the additional optimization mainly memorizes shortcut patterns rather than improving transferable representations.

In contrast, *SCOPE* exhibits a much smaller discrepancy between training and validation performance. Although its training Kappa is lower than that of full fine-tuning, its validation Kappa is consistently higher and more stable.

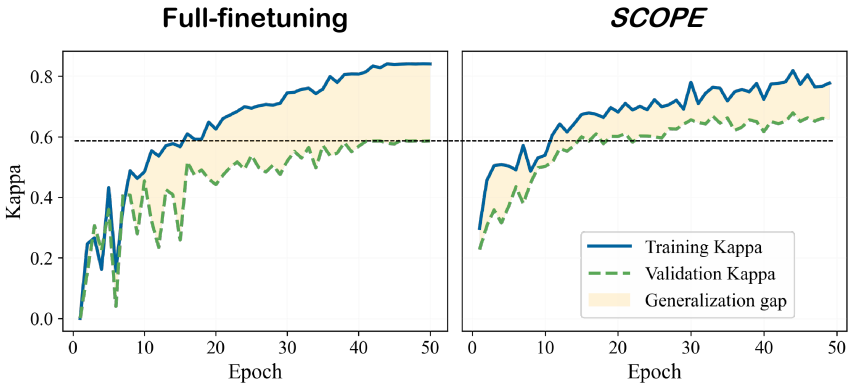

Figure 5: Generalization gap comparison under limited supervision. On ISRUC with 30% labeled subjects, full fine-tuning of *CodeBrain* obtains high training Kappa but suffers from a widening train–validation gap. In contrast, *SCOPE* reduces this gap and achieves higher validation Kappa.

C.2 Overconfident Miscalibration of EFM Backbones under Full-finetuning

As shown in Figure 2(A), fully fine-tuned EFMs exhibit clear confidence miscalibration: many predictions are assigned high confidence even when their empirical accuracy is substantially lower. Such overconfidence is particularly problematic for pseudo-label-based adaptation, where prediction confidence is commonly used as a proxy for pseudo-label reliability. We therefore further examine whether an adapted EFM can reliably provide supervision for itself under limited labeled data.

Evidence from existing self-training baselines. We first reorganize the empirical evidence from the semi-supervised baselines reported in Table 1. FixMatch and FineSSL, both of which rely on model-generated pseudo-labels, do not consistently improve label-limited EFM adaptation. In some settings, they even cause substantial performance degradation, such as FineSSL on PhysioNet-MI and FixMatch with the LaBraM backbone on CHB-MIT. These results suggest that directly relying on the EFM's own predictions does not provide uniformly reliable supervision under limited supervision.

Direct EFM self-training. To more directly isolate this issue, we additionally construct a controlled EFM self-training comparison. This comparison is not an existing baseline; rather, it directly tests whether the same fully fine-tuned EFM can improve itself using its own pseudo-labels. Specifically, for each backbone, we start from the checkpoint obtained by full fine-tuning on the available labeled subjects, use this checkpoint to generate pseudo-labels for the unlabeled recordings, and then continue fine-tuning the EFM jointly on the labeled and pseudo-labeled samples. All other experimental settings are kept consistent with the corresponding SCOPE experiments.

Table 3 reports the results on ISRUC under 20% and 30% labeled-subject ratios across five EFM backbones. SCOPE consistently outperforms direct EFM self-training across all ten backbone–label-ratio settings, with average improvements of **3.38 Kappa** and **3.04 F1**. Self-training can benefit from the additional unlabeled data in some cases, particularly for F1 under the 20% setting, but its improvements over full fine-tuning remain inconsistent: it improves Kappa in only **5/10** settings and F1 in **7/10** settings. Notably, its effectiveness also varies substantially across backbones; for example, the improvement is relatively small for CBraMod at 30% labels, while CodeBrain exhibits a clear degradation relative to full fine-tuning.

Consistency with the component ablation. The framework ablation provides complementary evidence. As shown in Table 2, removing the external supervision construction module leaves the model primarily dependent on its own predictions, yielding a self-training-like variant. This consistently degrades performance on both Transformer-based and SSM-based backbones. Together with the calibration analysis, existing semi-supervised baselines, and the direct EFM self-training comparison above, these results indicate that **additional unlabeled data alone is not sufficient**: the central challenge is to construct supervision that is more reliable than directly trusting the adapted

Table 3: Comparison with direct EFM self-training on ISRUC under 20% and 30% labeled training subjects.

Backbone	Method	20% labeled		30% labeled	
		Kappa	WF1	Kappa	WF1
Transformer-based backbones					
LaBraM	**SCOPE (Ours)** †	**51.67**±1.05	**60.95**±0.57	**56.19**±0.67	**65.15**±0.47
	Self-training †	47.87±0.68	56.44±0.77	49.46±0.83	59.57±1.00
	Full finetuning	48.55±1.25	53.87±1.03	51.26±1.26	61.73±1.30
CBraMod	**SCOPE (Ours)** †	**60.81**±0.38	**68.96**±0.36	**63.29**±1.12	**70.74**±0.64
	Self-training †	59.77±0.89	66.95±0.91	60.57±0.53	69.82±0.24
	Full finetuning	59.52±1.21	63.81±1.30	61.95±1.02	69.78±0.95
CSBrain	**SCOPE (Ours)** †	**60.59**±0.91	**68.90**±0.09	**61.51**±0.55	**69.86**±0.69
	Self-training †	56.95±0.41	65.46±0.31	60.68±0.36	68.98±0.31
	Full finetuning	52.59±0.97	62.33±0.90	60.97±1.04	69.19±1.09
SSM-style backbones					
EEGMamba	**SCOPE (Ours)** †	**60.24**±0.25	**69.63**±0.23	**63.07**±0.89	**71.41**±0.79
	Self-training †	57.08±0.20	66.43±0.22	62.50±0.65	70.32±0.61
	Full finetuning	53.59±0.86	64.39±0.80	62.02±0.87	68.71±0.91
CodeBrain	**SCOPE (Ours)** †	**57.48**±1.10	**65.76**±1.24	**63.46**±0.98	**71.11**±0.74
	Self-training †	53.06±0.44	63.21±0.37	56.56±0.77	64.91±0.52
	Full finetuning	49.79±1.77	59.90±1.62	58.46±1.42	67.05±1.57

† Methods additionally use unlabeled data with pseudo-label-based supervision. For self-training, each EFM is first fully fine-tuned on the available labeled subjects, used to generate pseudo-labels for the unlabeled recordings, and then further fine-tuned jointly on labeled and pseudo-labeled samples. Best and second-best results for each backbone and label ratio are shown in **bold** and underline, respectively.

EFM's own confidence. This observation motivates SCOPE to introduce external task-prior and prototype-based supervision rather than relying solely on EFM self-guidance.

C.3 Prediction Collapse under Limited Supervision

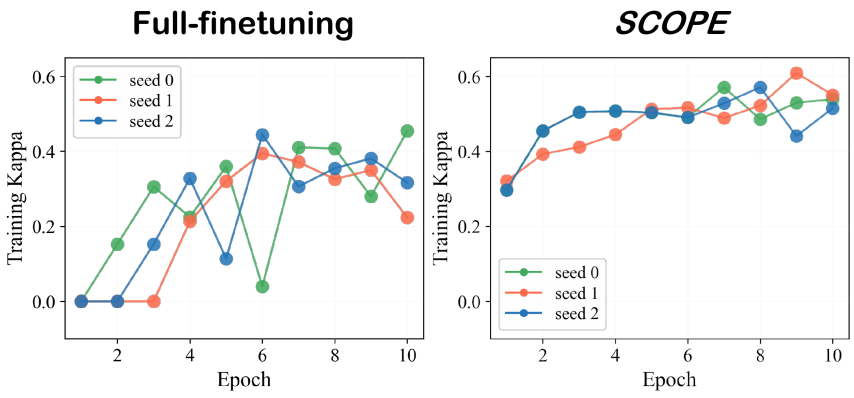

Figure 6: Prediction collapse comparison under limited supervision. We adapt *CodeBrain* on ISRUC with 30% labeled training subjects and report training Kappa across three random seeds. Full fine-tuning exhibits highly unstable optimization trajectories, with substantial fluctuations and abrupt performance degeneration during training. In contrast, *SCOPE* shows smoother and more stable optimization across seeds.

We further investigate the optimization instability of EFM adaptation under limited supervision. Figure 6 compares full fine-tuning and *SCOPE* on *CodeBrain* using 30% labeled subjects from ISRUC. Full fine-tuning exhibits highly unstable training trajectories across random seeds, characterized by

large fluctuations and abrupt performance degeneration during optimization. For example, training Kappa can improve substantially over several epochs and then sharply deteriorate in subsequent epochs. Such behavior suggests that, under limited labels, imperfect supervision signals can induce large updates in the highly plastic EFM parameter space, substantially disrupting previously learned representations and leading to unstable optimization.

In contrast, *SCOPE* reduces these fluctuations. Its training Kappa increases more consistently and follows smoother trajectories across seeds, indicating that constraining adaptation with more reliable supervision helps mitigate severe optimization degeneration under limited supervision.

C.4 Layer-wise Representation Drift

Finally, we analyze how pretrained representations evolve during adaptation using Centered Kernel Alignment (CKA). Specifically, for each layer, we measure the similarity between its current representation and the corresponding pretrained representation across training epochs. Importantly, representation change itself is not necessarily undesirable, since adapting to a new task naturally requires the pretrained feature space to be modified. We therefore focus not only on the magnitude of representation change, but also on its optimization trajectory.

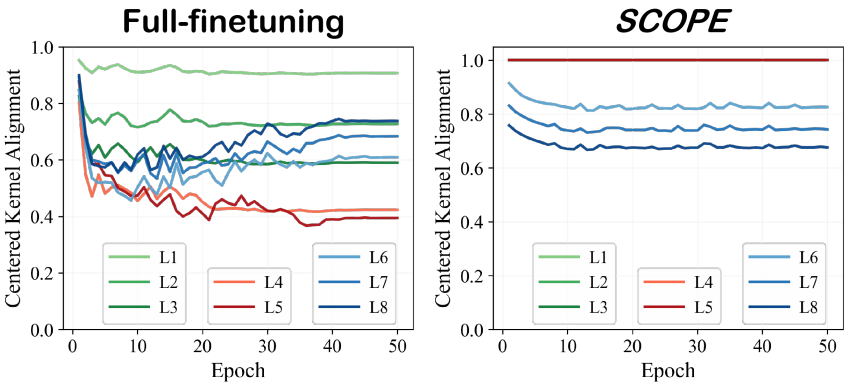

Figure 7: Layer-wise representation dynamics measured by CKA under limited supervision. We adapt *CodeBrain* on ISRUC with 30% labeled subjects and compute the CKA between each layer's current representation and its pretrained initialization across training epochs. Full fine-tuning exhibits abrupt early deviation followed by a pronounced rebound toward the pretrained representation in several deeper layers. In contrast, *SCOPE* produces a smoother adaptation trajectory in the adapted layers, with gradual representation change followed by stable convergence.

Figure 7 reveals qualitatively different representation dynamics between full fine-tuning and *SCOPE*. Under full fine-tuning, many layers rapidly deviate from their pretrained representations during the early stage of optimization. More notably, several deeper layers subsequently exhibit a clear CKA rebound: after initially moving substantially away from the pretrained feature space, their representations progressively return toward the pretrained initialization as training proceeds. This non-monotonic trajectory suggests that full fine-tuning may first induce excessive representational disruption under imperfect limited supervision, after which subsequent optimization partially recovers previously learned structure.

This behavior is distinct from the representation change naturally required for downstream adaptation. An effective adaptation process need not preserve the pretrained representation unchanged; rather, it should introduce task-relevant changes without first severely disrupting useful pretrained structure. The drop-and-rebound behavior under full fine-tuning therefore indicates an unstable adaptation trajectory, consistent with the substantial training fluctuations observed in Figure 6.

In contrast, *SCOPE* exhibits smoother representation dynamics. Because the first five backbone layers are frozen, their pretrained representations are directly preserved. For the deeper layers where prototype-conditioned adaptation is applied, CKA decreases gradually and then stabilizes, indicating a more controlled transition from pretrained to task-adapted representations.

D METRICS

To comprehensively evaluate our model, we compare it with a set of strong baselines commonly used in EEG analysis. These baselines are evaluated using metrics tailored for class-imbalanced scenarios, which are prevalent in EEG datasets. The metrics include:

For binary classification:

AUROC and **PRAUC** assess the performance of binary classifiers under different thresholds. While AUROC measures the trade-off between sensitivity and specificity, PRAUC focuses on precision-recall trade-offs, which are especially informative under severe class imbalance.

- For **AUROC**, it measures the model's ability to distinguish between positive and negative classes by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various thresholds. TPR and FPR can be defined as:

$$TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN}, \tag{25}$$

  where TP denotes the number of true positives, TN denotes the number of true negatives, FP denotes the number of false positives, FN denotes the number of false negatives. Then AUROC is defined as:

$$AUROC = \int_0^1 TPR \, d(FPR), \tag{26}$$

  For **PRAUC**, it evaluates the trade-off between precision and recall at different decision thresholds. It is calculated as the area under the curve plotted with precision on the y-axis and recall on the x-axis:

$$PRAUC = \int_0^1 Precision \, d(Recall), \tag{27}$$

  where the $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$.

For multi-class classification:

- **Cohen's Kappa**, which quantifies inter-class agreement beyond chance and is employed as the primary metric for multi-class classification. The Kappa can be calculated by:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{28}$$

  where p_o denotes the observed agreement and p_e denotes the expected agreement.

- **Weighted F1 Score**, which combines precision and recall while adjusting for class support, ensuring fair performance measurement across imbalanced datasets.

$$weight\ F1 = \sum_{i=1}^{K} w_i \frac{2 * Precision_i * Recall_i}{Precision_i + Recall_i}, \tag{29}$$

  where w_i is the weight of class i.

E DATASET DESCRIPTION AND DETAILED RESULTS

For evaluation, we consider six public EEG datasets spanning distinct cognitive and affective paradigms, including sleep staging, emotion recognition, and mental workload assessment. These datasets cover both multi-class and binary classification tasks and exhibit substantial diversity in recording protocols, channel configurations, signal durations, and subject populations, with detailed dataset statistics summarized in Table 4. To ensure a realistic assessment of cross-subject generalization, all datasets are evaluated under strict subject-wise splits, where labeled and unlabeled training subjects, validation subjects, and test subjects are mutually exclusive.

Table 4: Dataset Configurations

Dataset	Task	Labeled Training Subjects	Unlabeled Training Subjects	Validation Subjects	Test Subjects	Shape
ISRUC	Sleep Staging	1-24	25-80	81-90	91-100	6 ch × 30s
SEED	Emotion Recognition	1-3	4-10	11-12	13-15	62 ch × 4s
Mental	Workload Assessment	1-8	9-28	29-32	33-36	20 ch × 5s
PhysioNet-MI	Motor Imagery	1-21	22-70	71-89	90-109	64 ch × 4s
SHU-MI	Motor Imagery	1-4	5-15	16-20	21-25	32 ch × 4s
CHB-MIT	Seizure Detection	1-6	7-20	21-22	23-24	20 ch × 5s

E.1 Sleep Staging

The subset1 of ISRUC-Sleep dataset (Khalighi et al., 2016) is used for the sleep staging task. It consists of overnight polysomnographic EEG recordings collected from 100 subjects and annotated according to the American Academy of Sleep Medicine (AASM) standard into five sleep stages: Wake, N1, N2, N3, and REM. EEG signals are recorded using six channels and sampled at 200 Hz. Following standard practice, the continuous recordings are segmented into non-overlapping 30-second epochs, with each epoch treated as an individual sample. In this work, we adopt a strict subject-wise split to evaluate cross-subject generalization, where subjects 1–24 are used for labeled training, subjects 25–80 for unlabeled training, subjects 81–90 for validation, and subjects 91–100 for testing. Cohen's Kappa and weighted F1 score are used as evaluation metrics for this multi-class classification task.

Considering the sequential nature of sleep staging, and following prior work(Wang et al., 2025a; Zhou et al., 2025d; Wang et al., 2025b; Ma et al., 2026b), we formulate the task as a sequence-to-sequence prediction problem. Specifically, a sequence of 20 consecutive epochs is used as input, and the corresponding sequence of sleep stage labels is predicted. To model temporal dependencies across epochs, an additional Transformer layer is applied before the final classifier. This sequential modeling is implemented using the existing sequence-aware classifier of the EFM. The final classifier is a two-layer MLP, mapping epoch-level representations to the five sleep stages, with dimensions $30\text{s} \times 200\,\text{Hz} \rightarrow 512 \rightarrow 5$. Tables 5 and 6 present the baseline comparison results with standard deviations.

Table 5: Baseline comparisons under the limited-label setting on ISRUC with Transformer-based backbones.

Method	LaBraM		CBraMod		CSBrain	
	Kappa	WF1	Kappa	WF1	Kappa	WF1
Frozen	48.58±2.95	58.35±2.38	60.34±0.93	67.91±0.78	56.27±1.49	64.98±1.16
+LoRA	<u>55.61</u>±0.44	<u>65.05</u>±0.36	<u>63.16</u>±1.17	<u>70.40</u>±1.24	60.30±0.69	68.56±0.80
+TASTEFUL	52.16±1.66	62.21±1.90	60.76±0.60	68.25±0.50	59.99±0.70	68.56±0.67
+FixMatch †	33.14±1.29	43.83±2.49	20.42±3.79	23.23±3.45	44.53±1.59	53.48±2.27
+FineSSL †	54.24±1.84	63.76±1.97	53.46±1.97	61.21±2.27	47.65±0.62	54.61±0.44
+SCOPE (Ours) †	**56.19**±0.67	**65.15**±0.47	**63.29**±1.12	**70.74**±0.64	**61.51**±0.55	**69.86**±0.69
Full finetuning	51.26±1.26	61.73±1.30	61.95±1.02	69.78±0.95	<u>60.97</u>±1.04	<u>69.19</u>±1.09

† Methods also use unlabeled data for semi-supervised training.

E.2 Emotion Recognition

The SEED dataset (Zheng & Lu, 2015) is employed for EEG-based emotion recognition. It contains EEG recordings from 15 subjects exposed to audiovisual stimuli designed to elicit emotional responses. Original annotations include three emotion categories: Positive, Neutral, and Negative. To align with common baseline settings, we remove samples labeled as Neutral and formulate the task as a binary classification problem between Positive and Negative emotions. EEG signals are recorded using 62 channels and segmented into non-overlapping 4-second windows. For fair evaluation under

Table 6: Baseline comparisons under the limited-label setting on ISRUC with SSM-style backbones.

Method	EEGMamba		CodeBrain	
	Kappa	WF1	Kappa	WF1
Frozen	60.73±0.67	67.94±1.11	52.54±1.40	61.75±1.25
+LoRA	62.78±1.02	69.79±0.48	44.08±6.73	53.37±6.49
+TASTEFUL	49.11±1.78	57.75±2.35	47.13±3.00	56.44±2.94
+FixMatch †	32.00±1.93	44.07±1.25	57.23±0.64	66.74±0.51
+FineSSL †	54.02±7.36	65.92±1.90	48.94±0.83	57.50±0.96
+SCOPE (Ours) †	**63.07**±0.89	**71.41**±0.79	**63.46**±0.98	**71.11**±0.74
Full finetuning	62.02±0.87	68.71±0.91	58.46±1.42	67.05±1.57

† Methods also use unlabeled data for semi-supervised training.

inter-subject variability, a subject-wise split is adopted, where subjects 1–3 are used for labeled training, subjects 4–10 for unlabeled training, subjects 11–12 for validation, and subjects 13–15 for testing. Model performance is evaluated using AUROC and PRAUC, which are appropriate for binary classification under class imbalance.

The classifier is implemented as a multi-layer perceptron (MLP) that operates on the flattened input representation, with architecture $62 \times 4 \times 200 \to 4 \times 200 \to 200 \to 1$. Each hidden layer is followed by an ELU activation and dropout for regularization. Tables 7 and 8 present the baseline comparison results with standard deviations.

Table 7: Baseline comparisons under the limited-label setting on SEED with Transformer-based backbones.

Method	LaBraM		CBraMod		CSBrain	
	PRAUC	AUROC	PRAUC	AUROC	PRAUC	AUROC
Frozen	72.45±0.72	73.89±0.50	57.83±2.60	58.74±2.10	56.27±0.43	56.57±0.62
+LoRA	67.51±3.34	57.28±8.43	62.59±8.48	62.09±7.17	56.98±0.36	57.45±0.49
+TASTEFUL	64.03±3.26	64.12±2.61	64.95±0.81	**64.73**±1.11	58.38±0.45	57.34±0.60
+FixMatch †	74.08±1.26	**77.32**±1.48	60.21±3.03	59.16±3.03	59.71±0.76	57.34±0.95
+FineSSL †	72.99±0.57	73.78±0.21	57.84±1.31	58.42±3.31	53.46±1.05	53.86±1.27
+SCOPE (Ours) †	**74.25**±0.44	75.23±0.78	**65.57**±1.86	63.71±1.82	**63.05**±0.59	**62.09**±0.66
Full finetuning	66.84±1.60	66.25±1.77	64.12±0.95	63.39±1.64	57.54±3.08	57.40±2.22

* LaBraM is pretrained on cohorts including the SEED dataset.

† Methods also use unlabeled data for semi-supervised training.

Table 8: Baseline comparisons under the limited-label setting on SEED with SSM-style backbones.

Method	EEGMamba		CodeBrain	
	PRAUC	AUROC	PRAUC	AUROC
Frozen	58.27±0.75	57.30±0.67	63.44±1.54	63.93±2.06
+LoRA	58.64±1.38	57.35±1.32	63.83±2.67	58.61±3.38
+TASTEFUL	52.49±1.87	52.74±1.87	61.96±1.11	63.31±1.01
+FixMatch †	58.86±0.82	57.67±1.07	61.58±1.19	60.58±0.82
+FineSSL †	48.98±0.80	47.64±1.37	63.72±0.93	62.85±0.59
+SCOPE (Ours) †	**60.65**±0.50	**59.02**±0.52	**65.88**±0.51	**65.30**±0.45
Full finetuning	59.84±1.03	57.98±0.93	64.30±3.83	63.27±4.75

† Methods also use unlabeled data for semi-supervised training.

E.3 Workload Assessment Mental Arithmetic

The Mental Arithmetic dataset (Zyma et al., 2019) supports the task of mental stress detection using EEG signals. It contains recordings from 36 subjects under two distinct cognitive conditions: resting

and active engagement in mental arithmetic. EEG data labeled as "no stress" correspond to resting periods prior to the task, while "stress" labels are assigned to recordings during task performance. The signals were acquired using 20 electrodes placed according to the international 10–20 system, with an original sampling rate of 500 Hz. For consistency, the signals are resampled to 200 Hz and band-pass filtered between 0.5-45 Hz to suppress noise. Each recording is segmented into 5-second windows, yielding a total of 1,707 samples. We adopt a subject-wise split for fair evaluation with existing baselines: subjects 1-8 for labeled training, 9-28 for unlabeled training, 29-32 for validation, and 33-36 for testing.

For this task with 5-second EEG segments sampled at 200 Hz, the classifier is implemented as a multi-layer perceptron (MLP). For backbones that do not use a reference electrode (i.e., CSBrain(Zhou et al., 2025d) and LaBraM(Jiang et al., 2024a)), the input consists of 19 channels, and the classifier maps flattened representations of size $19 \times 5 \times 200 \to 5 \times 200 \to 200 \to 1$, with GELU activations and dropout applied between layers. For other backbones that retain the reference electrode, 20-channel inputs are used, and the classifier follows the architecture $20 \times 5 \times 200 \to 5 \times 200 \to 200 \to 1$, with ELU activations and dropout applied between layers. Tables 9 and 10 present the baseline comparison results with standard deviations.

Table 9: Baseline comparisons under the limited-label setting on Mental Arithmetic with Transformer-based backbones.

Method	LaBraM		CBraMod		CSBrain	
	PRAUC	AUROC	PRAUC	AUROC	PRAUC	AUROC
Frozen	60.83±0.53	79.25±0.18	47.19±0.86	63.53±1.32	38.30±1.76	66.68±0.87
+LoRA	64.47±2.92	54.16±1.78	43.14±2.16	70.45±0.91	33.78±0.38	63.24±0.38
+TASTEFUL	62.12±1.68	77.14±1.89	41.76±1.60	62.72±0.63	**56.56**±2.66	**76.12**±2.87
+FixMatch †	53.92±12.48	74.41±7.56	46.70±4.30	66.85±2.07	43.24±1.89	69.13±0.60
+FineSSL †	53.12±2.41	60.72±2.85	41.51±7.61	64.43±2.41	36.41±1.05	66.35±1.86
+SCOPE (Ours) †	**64.64**±2.49	**82.93**±1.89	**54.44**±0.73	**74.39**±0.98	47.60±0.83	71.09±1.04
Full finetuning	45.49±4.16	70.72±2.35	47.35±2.75	69.91±2.72	40.08±5.68	67.39±2.74

† Methods also use unlabeled data for semi-supervised training.

Table 10: Baseline comparisons under the limited-label setting on Mental Arithmetic with SSM-style backbones.

Method	EEGMamba		CodeBrain	
	PRAUC	AUROC	PRAUC	AUROC
Frozen	37.19±1.01	64.98±1.30	46.70±2.09	71.25±2.72
+LoRA	38.12±0.93	65.45±0.81	44.05±0.84	68.49±1.32
+TASTEFUL	34.49±4.78	62.87±4.44	29.86±3.20	59.73±2.72
+FixMatch †	42.61±1.47	67.61±1.06	44.53±2.07	69.30±1.31
+FineSSL †	39.11±1.73	66.86±0.72	47.14±4.08	71.25±2.83
+SCOPE (Ours) †	**49.46**±0.18	**68.48**±0.34	**49.30**±1.02	**72.40**±0.70
Full finetuning	42.89±2.73	65.36±1.15	47.10±2.09	72.13±2.72

† Methods also use unlabeled data for semi-supervised training.

E.4 Multi-effector Motor Imagery

The PhysioNet-MI dataset (Schalk et al., 2004) is a large-scale EEG motor movement and motor imagery dataset collected using the BCI2000 system. It includes recordings from 109 subjects performing motor execution and motor imagery tasks, with 64 EEG channels sampled at 160 Hz. Following prior motor imagery evaluation protocols, we focus on motor imagery classification and segment EEG signals into 4-second windows. To ensure fair cross-subject evaluation under limited supervision, we adopt a subject-wise split: subjects 1-21 are used as labeled training subjects, subjects 22-70 as unlabeled training subjects, subjects 71-89 for validation, and subjects 90-109 for testing.

For this task with 4-second EEG segments resampled at 200 Hz, the classifier is implemented as a multi-layer perceptron (MLP). The classifier follows the architecture $64 \times 4 \times 200 \rightarrow 4 \times 200 \rightarrow 200 \rightarrow 4$, with ELU activations and dropout applied between layers. Tables 11 and 12 present the baseline comparison results with standard deviations.

Table 11: Baseline comparisons under the limited-label setting on PhysioNet-MI with Transformer-based backbones.

Method	LaBraM		CBraMod		CSBrain	
	Kappa	WF1	Kappa	WF1	Kappa	WF1
Frozen	22.81±0.66	41.80±0.61	30.73±3.04	47.60±3.24	30.55±1.72	47.50±1.82
+LoRA	22.79±1.23	41.69±0.54	32.97±0.88	49.46±1.07	<u>37.81</u>±0.86	<u>53.25</u>±0.52
+TASTEFUL	29.05±2.75	46.83±1.91	<u>36.74</u>±0.85	<u>52.94</u>±0.84	25.98±3.70	44.72±3.00
+FixMatch †	<u>41.50</u>±1.81	<u>55.64</u>±1.29	23.32±1.63	40.15±3.02	27.17±1.35	45.49±0.92
+FineSSL †	8.69±2.72	26.50±3.68	20.90±13.11	33.38±15.20	7.05±2.26	22.16±3.85
+SCOPE (Ours) †	**41.80**±0.37	**56.33**±0.31	**37.84**±1.79	**53.86**±1.25	**40.62**±0.40	**55.79**±0.36
Full finetuning	38.31±1.90	52.73±1.53	33.16±2.74	49.31±2.33	37.33±2.26	53.20±1.72

† Methods also use unlabeled data for semi-supervised training.

Table 12: Baseline comparisons under the limited-label setting on PhysioNet-MI with SSM-style backbones.

Method	EEGMamba		CodeBrain	
	Kappa	WF1	Kappa	WF1
Frozen	30.80±1.93	47.90±1.31	26.05±3.40	43.80±3.65
+LoRA	<u>33.62</u>±1.01	<u>50.39</u>±0.77	<u>37.11</u>±0.49	<u>53.09</u>±0.43
+TASTEFUL	16.56±1.90	35.30±1.50	20.77±2.48	40.58±1.87
+FixMatch †	28.76±0.91	45.45±0.65	21.01±0.68	40.20±0.44
+FineSSL †	4.37±3.51	16.81±5.05	10.67±1.85	28.87±2.62
+SCOPE (Ours) †	**35.75**±0.11	**52.06**±0.34	**37.42**±0.53	**53.24**±0.50
Full finetuning	33.58±2.07	50.17±1.55	36.05±5.23	51.90±5.03

† Methods also use unlabeled data for semi-supervised training.

E.5 Hand Motor Imagery

The SHU-MI dataset (Ma et al., 2022) is a multi-session EEG motor imagery dataset designed to study cross-session and cross-subject variability in brain-computer interfaces. It contains recordings from 25 subjects collected across five sessions on different days. Each session includes left-hand and right-hand motor imagery trials, making the dataset suitable for binary motor imagery classification. EEG signals were recorded with 32 channels at 250 Hz, and each trial contains a 4-second motor imagery period. For consistency with other datasets, we resample the signals to 200 Hz and use 4-second windows. We adopt a subject-wise split: subjects 1-4 are used as labeled training subjects, subjects 5-15 as unlabeled training subjects, subjects 16-20 for validation, and subjects 21-25 for testing.

For this task with 4-second EEG segments sampled at 200 Hz, the classifier is implemented as a multi-layer perceptron (MLP). For backbones that do not use a reference electrode (i.e., CSBrain(Zhou et al., 2025d) and LaBraM(Jiang et al., 2024a)), the input consists of 30 channels, and the classifier maps flattened representations of size $30 \times 4 \times 200 \rightarrow 4 \times 200 \rightarrow 200 \rightarrow 1$, with GELU activations and dropout applied between layers. For other backbones that retain the reference electrode, 32-channel inputs are used, and the classifier follows the architecture $32 \times 4 \times 200 \rightarrow 4 \times 200 \rightarrow 200 \rightarrow 1$, with ReLU activations and dropout applied between layers. Tables 13 and 14 present the baseline comparison results with standard deviations.

Table 13: Baseline comparisons under the limited-label setting on SHU-MI with Transformer-based backbones.

Method	LaBraM		CBraMod		CSBrain	
	PRAUC	AUROC	PRAUC	AUROC	PRAUC	AUROC
Frozen	61.81±1.16	63.49±0.96	54.29±1.42	54.64±1.52	58.31±0.40	58.96±0.44
+LoRA	61.55±1.55	63.12±0.66	62.05±1.24	61.83±1.30	56.17±0.23	57.51±0.37
+TASTEFUL	50.43±0.34	50.36±0.58	51.31±0.97	50.57±0.45	51.00±0.64	51.44±0.98
+FixMatch †	49.15±0.60	49.62±0.79	51.14±1.50	51.04±1.34	51.68±0.81	51.48±0.79
+FineSSL †	61.59±1.85	63.11±1.25	51.29±1.43	51.69±1.39	51.61±0.96	52.31±0.97
+SCOPE (Ours) †	**63.33**±1.48	**64.21**±1.31	**66.31**±0.69	**67.80**±0.38	**60.99**±1.62	**61.58**±1.39
Full finetuning	62.96±2.35	63.29±1.53	64.85±1.60	66.63±1.53	58.02±1.90	58.80±1.80

† Methods also use unlabeled data for semi-supervised training.

Table 14: Baseline comparisons under the limited-label setting on SHU-MI with SSM-style backbones.

Method	EEGMamba		CodeBrain	
	PRAUC	AUROC	PRAUC	AUROC
Frozen	56.37±0.29	56.92±0.12	58.96±0.33	58.30±0.27
+LoRA	57.11±0.41	57.35±0.65	58.01±0.91	58.66±1.10
+TASTEFUL	50.81±0.57	51.11±0.76	52.42±0.52	52.78±0.92
+FixMatch †	53.38±2.25	53.31±2.03	55.52±0.78	54.96±0.58
+FineSSL †	54.06±1.10	54.83±0.86	57.69±1.36	57.60±1.28
+SCOPE (Ours) †	**57.39**±0.33	**57.46**±0.34	**60.56**±0.95	**60.32**±1.68
Full finetuning	57.09±0.52	57.22±0.92	57.46±2.98	58.27±3.34

† Methods also use unlabeled data for semi-supervised training.

E.6 Seizure Detection

CHB-MIT (Shoeb, 2009) is a public scalp EEG database collected from pediatric subjects with intractable seizures at the Children's Hospital Boston. The recordings consist of long-term continuous EEG signals with expert seizure annotations, making the dataset widely used for seizure detection. Following common preprocessing practice, we formulate the task as binary seizure detection and segment EEG recordings into 10-second windows using 16 selected channels. We adopt a subject-wise split to evaluate cross-subject generalization: subjects 1-6 are used as labeled training subjects, subjects 7-20 as unlabeled training subjects, subjects 21-22 for validation, and subjects 23-24 for testing.

The classifier is implemented as a multi-layer perceptron (MLP) that operates on the flattened input representation, with architecture $16 \times 10 \times 200 \to 10 \times 200 \to 200 \to 1$. Each hidden layer is followed by an ELU activation and dropout for regularization. Tables 15 and 16 present the baseline comparison results with standard deviations.

F Baseline and Backbone Models

F.1 Baseline Models

We evaluate our method under a data-limited adaptation setting, where only a small portion of labeled samples is available. We consider two complementary groups of baselines.

First, we include commonly used lightweight EEG models trained directly on the target datasets under the same limited-supervision condition. These models reflect the performance achievable without leveraging large-scale pretraining.

Table 15: Baseline comparisons under the limited-label setting on CHB-MIT with Transformer-based backbones.

Method	LaBraM		CBraMod		CSBrain	
	PRAUC	AUROC	PRAUC	AUROC	PRAUC	AUROC
Frozen	30.53±3.72	86.30±0.90	28.82±9.23	87.11±4.09	20.54±4.00	87.08±0.95
+LoRA	32.16±4.16	86.96±0.43	23.55±8.16	86.15±2.72	18.29±3.54	87.17±0.51
+TASTEFUL	33.34±6.80	87.55±2.47	33.43±4.38	87.61±0.48	34.24±3.76	87.90±1.01
+FixMatch †	29.97±2.36	87.00±0.62	30.78±6.08	87.53±1.63	32.37±3.33	87.76±0.24
+FineSSL †	33.66±3.39	86.03±0.84	33.64±4.76	88.92±1.66	34.21±2.29	87.62±1.24
+SCOPE (Ours) †	**36.05**±0.94	**89.13**±0.45	**35.02**±3.84	**89.04**±1.49	**34.96**±4.73	**89.21**±0.94
Full finetuning	33.62±7.03	87.25±1.12	28.59±10.34	85.68±2.27	33.90±3.78	86.85±1.56

† Methods also use unlabeled data for semi-supervised training.

Table 16: Baseline comparisons under the limited-label setting on CHB-MIT with SSM-style backbones.

Method	EEGMamba		CodeBrain	
	PRAUC	AUROC	PRAUC	AUROC
Frozen	30.09±1.56	87.38±0.18	28.37±3.03	86.45±1.46
+LoRA	32.61±0.88	86.53±0.13	15.47±2.15	84.60±2.01
+TASTEFUL	**33.91**±0.19	88.49±0.75	29.42±4.41	88.33±0.36
+FixMatch †	24.82±3.41	85.31±1.15	28.81±9.05	85.85±4.66
+FineSSL †	32.06±1.50	85.50±0.94	35.08±4.49	87.67±1.28
+SCOPE (Ours) †	33.39±0.41	**88.94**±1.80	**36.63**±3.92	**88.77**±1.47
Full finetuning	31.06±1.57	85.41±1.16	28.33±8.25	87.60±2.33

† Methods also use unlabeled data for semi-supervised training.

- **EEGNet** (Lawhern et al., 2018): A compact Convolutional Neural Network that introduced depthwise and separable convolutions to create an efficient architecture for EEG classification. It is designed to generalize effectively across diverse BCI paradigms, demonstrating robust performance even with limited training data.
- **EEG Conformer** (Song et al., 2023): A compact Convolutional–Transformer architecture for EEG classification that integrates efficient temporal–spatial convolutional feature extraction with self-attention to capture long-range temporal dependencies. By combining local feature learning with global correlation modeling, it provides a framework that generalizes across different EEG decoding paradigms.

Second, we compare against representative adaptation baselines for pretrained backbones, including (i) parameter-efficient fine-tuning (PEFT) methods and (ii) standard semi-supervised learning (SSL) algorithms. These baselines help disentangle the effect of our task-prior supervision mechanism from (a) generic PEFT strategies for adapting large pretrained models and (b) generic SSL objectives based on pseudo-labeling and consistency regularization.

- **LoRA** (Hu et al., 2022): A parameter-efficient adaptation method that freezes the pretrained model weights and introduces trainable low-rank decomposition matrices into Transformer layers. This design significantly reduces the number of trainable parameters and computational cost for downstream tasks while enabling effective task-specific adaptation of large pretrained models.
- **TASTEFUL** (Jeon et al., 2025): A parameter-efficient transfer learning method designed for EEG foundation models. It freezes the pretrained backbone and uses learnable queries with cross-attention to extract task-relevant intermediate features. Learnable spatial position vectors further guide feature selection, yielding compact representations for downstream tasks with a small number of trainable parameters.

Table 17: Baseline comparison in terms of data visibility and parameter efficiency, where r_l denotes the labeled-subject ratio.

Method	Unlabeled Data	Params Tuned
Frozen	$\times$	0%
LoRA	$\times$	2-5%
TASTEFUL	$\times$	5-7%
FixMatch	$1 - r_l$	100%
FineSSL	$1 - r_l$	60-72%
SCOPE (Ours)	$1 - r_l$	2-5%
Limited-label Full Fine-tuning	$\times$	100%

- **FixMatch** (Sohn et al., 2020): A simple and effective semi-supervised learning approach that combines pseudo-labeling with consistency regularization. It generates pseudo-labels from high-confidence predictions on weakly augmented unlabeled samples and trains the model to predict the same labels under strong data augmentation.
- **FineSSL** (Gan & Wei, 2024): A semi-supervised learning framework that adapts pretrained foundation models to improve the robustness and effectiveness of SSL methods. It addresses bias-related issues in foundation models through balanced margin softmax and decoupled label smoothing, and can be seamlessly integrated with various fine-tuning strategies and modern SSL algorithms.

As summarized in Table 17, the considered baselines cover different levels of data visibility and parameter efficiency, ensuring that *SCOPE* is compared against at least one fair counterpart along each evaluation axis. Specifically, LoRA provides a parameter-efficient supervised baseline using the same labeled ratio ρ; FixMatch and FineSSL use the same labeled and unlabeled data as *SCOPE*; and r_l full fine-tuning evaluates unconstrained full-model adaptation under the same labeled supervision.

F.2 Backbone Models

As a plug-and-play adapter, our method is architecture-agnostic and can be integrated into a wide range of EFMs without modifying backbone parameters. We therefore select five publicly available EEG foundation models with released pretrained weights as backbones, spanning diverse architectural paradigms(three transformer-based models: **LaBraM**, **CBraMod**, **CSBrain**; and two SSM-style models: **EEGMamba**, **CodeBrain**). We briefly introduce each backbone below.

- **LaBraM** (Jiang et al., 2024a): A scalable Transformer framework for learning general-purpose EEG representations from extensive datasets. It is pretrained on a diverse collection of recordings to capture features that are broadly applicable to downstream BCI tasks, utilizing efficient self-attention and task-specific adapters to facilitate fine-tuning.
- **CBraMod** (Wang et al., 2025a): An EEG foundation model designed to handle complex dependencies in brain signals. It features a criss-cross Transformer architecture with parallel attention mechanisms that independently model spatial and temporal relationships within the data.
- **CSBrain** (Zhou et al., 2025d): An EEG foundation model that explicitly models the cross-scale spatiotemporal structure of brain signals. Its backbone alternates between Cross-scale Spatiotemporal Tokenization (CST) and Structured Sparse Attention (SSA) layers. CST aggregates multi-resolution temporal patterns and region-wise spatial features into compact, scale-aware tokens, while SSA captures long-range dependencies across temporal windows and anatomical brain regions through structured and sparse attention.
- **EEGMamba** (Wang et al., 2025b): An EEG foundation backbone built upon state space models to handle long EEG sequences and heterogeneous tasks. It integrates a Spatio-Temporal-Adaptive (ST-Adaptive) module to unify inputs with varying channel numbers and signal lengths, followed by bidirectional Mamba blocks for linear-time global temporal modeling. To support multi-task EEG decoding, EEGMamba further incorporates a task-aware Mixture of Experts (MoE) with a universal expert, enabling both task-specific specialization and shared representation learning across EEG tasks.

- **CodeBrain** (Ma et al., 2026b): An efficient two-stage EEG foundation model. It first employs a novel TFDual-Tokenizer to generate discrete representations by independently processing temporal and frequency components. Subsequently, its EEGSSM architecture, which integrates structured global convolutions with a sliding window attention mechanism, is trained via masked prediction to efficiently capture the multi-scale dependencies inherent in brain signals.

G Additional Ablation Studies on Design Choices

To validate our design choices, we present additional ablation results using two representative EEG foundation models on the representative ISRUC dataset: CSBrain (Zhou et al., 2025d), a Transformer-based backbone, and CodeBrain (Ma et al., 2026b), which adopts a structured state-space modeling (SSM) paradigm. Together, they reflect two dominant modeling families in current EEG foundation model research. These results demonstrate that our method is largely model-agnostic rather than tailored to a specific backbone.

G.1 Ablation Study on Different Label Ratio

We evaluate *SCOPE* on ISRUC under seven labeled-subject ratios: 5%, 10%, 20%, 30%, 50%, 75%, and 100%. Our primary objective is to support the practical use of pretrained EEG foundation models when labeled data from the target task are scarce. The higher-label settings characterize how the relative benefits of *SCOPE* and full fine-tuning change as more supervision becomes available. The 30% results are reported in Table 1, and the remaining settings are presented in Tables 18, 19, and 20.

Table 21 summarizes the average ranks over Kappa and weighted F1 for each backbone and labeled-subject ratio. Across the five settings at or below 50%, *SCOPE* achieves the best aggregate rank for every backbone, with an overall average rank of 1.14 across backbones, compared with 3.58 for full fine-tuning. *SCOPE* also exceeds full fine-tuning on both metrics for all five backbones at every evaluated ratio from 5% through 75%. These results support its effectiveness for adaptation with limited labeled subjects, although other methods can attain better ranks in individual settings.

Under 100% supervision, full fine-tuning ranks first on both metrics for all five backbones, while *SCOPE* ranks second for four backbones and third for CSBrain. Averaging over the 75% and 100% settings yields similar overall ranks for *SCOPE* and full fine-tuning (1.70 and 1.65, respectively), but this aggregate masks their reversal between the two settings. These findings clarify the intended scope of our contribution: *SCOPE* addresses the practical challenge of transferring EEG foundation models with scarce target-task labels.

Table 18: Baseline comparisons under the limited-label setting on ISRUC dataset (5% and 10% labeled training subjects).

Backbone	Method	5% (4 subjects)		10% (8 subjects)	
		Kappa	WF1	Kappa	WF1
Non-Pretrained	EEGNet	23.76±3.08	37.75±2.37	32.55±3.53	43.85±2.62
	EEGConformer	26.25±2.06	39.69±3.24	35.68±3.58	42.77±3.72
Transformer-based backbones					
LaBraM	Frozen	30.82±1.16	40.61±1.35	36.69±1.41	45.13±1.55
	+LoRA	31.43±2.54	41.34±2.35	33.96±0.77	42.89±0.73
	+TASTEFUL	33.78±0.70	43.50±0.76	31.15±2.51	42.74±1.23
	+FixMatch †	6.49±6.54	19.85±8.48	5.08±6.39	13.52±6.84
	+FineSSL †	30.70±0.73	40.17±0.49	34.22±0.58	43.81±0.63
	+SCOPE (Ours) †	**38.81**±0.41	**44.15**±0.55	**37.84**±1.26	**46.92**±1.04
	Full finetuning	35.02±2.16	40.97±1.55	36.37±1.68	40.15±1.62
CBraMod	Frozen	19.47±1.11	19.63±0.54	17.33±3.97	18.91±1.90
	+LoRA	18.88±1.08	20.08±0.76	18.90±1.19	19.74±0.54
	+TASTEFUL	8.22±2.63	**22.62**±3.33	18.08±6.08	**33.18**±5.36
	+FixMatch †	7.19±7.59	17.04±10.26	2.37±3.63	12.88±6.64
	+FineSSL †	8.01±2.11	18.19±4.76	**20.96**±2.61	26.21±4.19
	+SCOPE (Ours) †	**20.28**±1.27	20.31±0.65	18.99±0.81	20.14±0.68
	Full finetuning	19.60±1.71	19.78±0.75	18.94±2.39	19.78±0.40
CSBrain	Frozen	35.58±5.69	41.28±5.70	35.13±4.42	41.11±4.51
	+LoRA	33.37±2.24	41.32±3.27	38.57±3.84	46.01±4.22
	+TASTEFUL	18.55±2.33	31.90±2.36	16.88±1.83	22.14±6.05
	+FixMatch †	6.85±6.83	18.65±10.19	6.13±4.68	15.18±6.68
	+FineSSL †	30.64±11.91	40.69±7.81	36.06±6.61	43.53±5.71
	+SCOPE (Ours) †	**38.19**±2.13	**44.12**±2.83	**39.26**±1.38	**46.03**±1.49
	Full finetuning	27.69±4.92	33.86±4.73	30.77±4.26	39.26±4.15
SSM-style backbones					
EEGMamba	Frozen	35.19±3.06	42.00±3.81	40.31±1.12	50.44±1.07
	+LoRA	35.68±3.19	42.75±3.96	43.25±1.06	49.65±0.69
	+TASTEFUL	37.17±1.31	**47.03**±0.44	38.65±0.15	47.01±0.29
	+FixMatch †	13.63±3.89	26.40±6.37	14.86±2.27	26.52±4.79
	+FineSSL †	19.97±2.12	23.41±5.13	20.86±0.66	19.80±1.02
	+SCOPE (Ours) †	**38.75**±1.79	45.40±1.84	**47.25**±0.49	**52.95**±0.43
	Full finetuning	33.51±3.05	41.14±2.65	43.57±1.24	48.07±1.18
CodeBrain	Frozen	39.01±1.22	47.47±0.31	37.33±0.88	47.69±0.60
	+LoRA	36.45±2.74	45.47±1.76	37.36±2.37	46.45±1.75
	+TASTEFUL	37.17±1.31	47.03±0.44	38.65±0.15	47.01±0.29
	+FixMatch †	14.18±0.47	29.64±0.95	12.53±1.53	26.12±2.90
	+FineSSL †	36.61±0.82	47.24±0.25	21.84±2.05	21.91±4.96
	+SCOPE (Ours) †	**39.24**±0.43	**47.83**±0.59	**39.54**±0.53	**47.92**±0.36
	Full finetuning	30.97±1.94	43.93±2.02	37.85±1.08	46.93±1.30

Table 19: Baseline comparisons under the limited-label setting on ISRUC dataset (20% and 50% labeled training subjects).

Backbone	Method	20% labeled		50% labeled	
		Kappa	WF1	Kappa	WF1
Non-Pretrained	EEGNet	41.80±3.09	48.79±2.23	52.04±1.94	62.37±1.57
	EEGConformer	42.43±2.89	49.64±2.77	53.93±3.68	63.54±3.21
Transformer-based backbones					
LaBraM	Frozen	43.66±1.25	56.86±1.21	55.07±0.52	64.47±0.55
	+LoRA	50.57±0.87	54.57±1.11	57.92±0.98	66.69±1.26
	+TASTEFUL	48.48±3.58	58.37±3.88	55.30±0.59	64.56±0.26
	+FixMatch †	12.57±5.25	18.21±6.43	40.99±1.10	52.55±1.00
	+FineSSL †	50.94±1.22	60.01±0.87	55.56±2.06	64.80±1.68
	+SCOPE (Ours) †	**51.67**±1.05	**60.95**±0.57	**65.52**±1.39	**73.27**±1.12
	Full finetuning	48.55±1.25	53.87±1.03	63.83±1.61	71.47±1.52
CBraMod	Frozen	60.10±1.94	67.89±1.97	65.22±1.14	72.43±0.52
	+LoRA	52.76±3.52	61.58±3.90	58.83±2.02	68.00±1.54
	+TASTEFUL	57.00±3.20	63.45±4.47	68.19±0.98	75.17±0.71
	+FixMatch †	1.61±1.08	9.73±1.11	21.36±9.71	27.48±10.16
	+FineSSL †	43.81±0.69	49.52±0.58	63.14±3.11	71.19±2.48
	+SCOPE (Ours) †	**60.81**±0.38	**68.96**±0.36	**72.48**±0.80	**78.51**±0.75
	Full finetuning	59.52±1.21	63.81±1.30	70.29±1.10	76.67±0.89
CSBrain	Frozen	53.11±1.73	63.36±1.59	63.92±2.06	71.81±1.68
	+LoRA	40.56±1.85	49.45±1.53	37.29±4.19	46.75±3.71
	+TASTEFUL	51.08±3.89	59.53±5.10	65.35±0.25	73.10±0.23
	+FixMatch †	39.58±2.64	51.05±3.46	63.20±0.90	71.41±0.70
	+FineSSL †	45.30±1.90	53.14±1.66	63.22±0.85	71.39±0.69
	+SCOPE (Ours) †	**60.59**±0.91	**68.90**±0.09	**65.48**±0.95	**73.13**±0.79
	Full finetuning	52.59±0.97	62.33±0.90	63.83±1.61	71.70±1.40
SSM-style backbones					
EEGMamba	Frozen	58.78±0.92	67.60±0.68	67.39±0.86	74.83±0.57
	+LoRA	59.15±0.85	67.90±0.82	**70.33**±0.62	**76.85**±0.68
	+TASTEFUL	51.51±2.09	61.82±2.10	57.99±1.26	66.79±1.14
	+FixMatch †	22.41±1.42	31.72±1.36	36.78±1.14	49.28±0.96
	+FineSSL †	47.59±1.67	55.45±1.28	69.09±0.45	75.99±0.47
	+SCOPE (Ours) †	**60.24**±0.25	**69.63**±0.23	69.33±0.54	76.21±0.49
	Full finetuning	53.59±0.86	64.39±0.80	68.79±1.35	75.77±1.27
CodeBrain	Frozen	49.89±1.65	57.51±2.34	59.33±1.48	67.90±1.34
	+LoRA	52.07±1.28	60.12±1.31	46.31±3.37	56.54±3.15
	+TASTEFUL	46.64±2.16	54.23±2.41	51.64±1.37	61.16±1.19
	+FixMatch †	42.31±1.10	49.10±1.00	59.86±2.10	69.90±1.70
	+FineSSL †	42.32±1.12	49.08±1.05	50.87±2.10	60.96±1.79
	+SCOPE (Ours) †	**57.48**±1.10	**65.76**±1.24	**64.81**±0.65	**71.92**±0.73
	Full finetuning	49.79±1.77	59.90±1.62	61.89±1.57	69.71±1.28

Table 20: Baseline comparisons under the high-label setting on ISRUC dataset (75% and 100% labeled training subjects).

Backbone	Method	75%		100%	
		Kappa	WF1	Kappa	WF1
Non-Pretrained	EEGNet	64.55±1.92	69.90±1.47	70.40±1.73	75.13±1.24
	EEGConformer	65.70±1.77	70.62±1.16	71.43±1.62	76.34±1.51
Transformer-based backbones					
LaBraM	Frozen	55.73±0.70	65.39±1.39	56.38±2.41	65.44±2.09
	+LoRA	57.15±0.58	64.77±1.04	59.79±0.51	68.24±0.61
	+TASTEFUL	57.59±2.19	65.55±1.40	60.63±1.16	67.90±1.20
	+FixMatch †	47.64±2.01	58.34±1.90	53.13±1.15	62.98±1.02
	+FineSSL †	56.23±0.12	65.57±0.10	56.68±0.08	66.59±0.11
	+SCOPE (Ours) †	**65.82**±0.73	**73.68**±0.92	66.95±1.12	74.61±1.20
	Full finetuning	65.02±1.25	73.29±1.05	**72.31**±1.82	**78.10**±1.33
CBraMod	Frozen	68.07±1.25	76.76±0.71	71.50±0.72	77.57±0.71
	+LoRA	64.52±1.20	71.77±1.04	65.93±2.13	72.96±2.12
	+TASTEFUL	70.02±0.98	76.56±0.67	72.28±1.32	78.45±1.08
	+FixMatch †	55.07±0.99	62.13±0.97	66.94±1.30	73.22±1.51
	+FineSSL †	65.91±0.14	72.78±0.22	68.21±0.31	74.81±0.41
	+SCOPE (Ours) †	**73.99**±0.70	**79.65**±0.74	73.40±1.69	79.28±1.30
	Full finetuning	71.93±1.24	77.30±1.25	**74.42**±1.52	**80.11**±0.99
CSBrain	Frozen	67.49±1.52	74.46±1.28	68.18±0.79	75.12±0.56
	+LoRA	67.58±0.19	74.23±0.16	67.86±0.78	74.78±0.64
	+TASTEFUL	**70.37**±0.41	**76.68**±0.48	72.37±1.34	78.09±1.51
	+FixMatch †	50.87±0.04	61.57±0.25	63.55±1.59	73.18±1.22
	+FineSSL †	65.52±0.17	73.16±0.14	66.97±0.06	74.60±0.04
	+SCOPE (Ours) †	68.39±0.57	75.23±0.39	70.45±1.11	76.51±1.25
	Full finetuning	67.52±1.36	74.57±1.20	**73.44**±0.91	**79.32**±1.05
SSM-style backbones					
EEGMamba	Frozen	68.74±0.64	75.23±0.64	71.04±0.91	77.66±1.00
	+LoRA	69.65±0.86	75.85±0.89	73.36±0.29	77.64±0.27
	+TASTEFUL	60.93±1.34	69.15±1.12	61.60±2.81	69.73±2.58
	+FixMatch †	61.42±0.08	63.26±0.55	64.43±0.60	69.67±0.76
	+FineSSL †	68.06±1.21	74.46±1.13	71.35±0.51	78.12±0.33
	+SCOPE (Ours) †	**69.93**±0.14	**76.51**±0.20	74.03±0.65	80.23±1.04
	Full finetuning	69.79±1.13	76.16±1.08	**74.86**±1.22	**80.54**±0.76
CodeBrain	Frozen	59.99±1.04	68.21±1.27	63.30±1.36	72.49±1.39
	+LoRA	59.57±0.59	67.65±0.51	61.64±1.10	69.60±0.63
	+TASTEFUL	63.16±3.09	70.53±2.81	62.40±3.20	69.90±2.78
	+FixMatch †	56.85±0.76	64.53±1.02	59.06±0.30	65.80±0.07
	+FineSSL †	56.23±0.40	65.57±0.08	59.47±0.09	67.00±0.14
	+SCOPE (Ours) †	**66.86**±0.83	**72.81**±0.60	69.45±0.92	76.89±0.48
	Full finetuning	64.21±1.75	71.61±1.49	**74.76**±0.40	**80.20**±0.18

Table 21: Average method ranks on ISRUC across seven labeled-subject ratios. Lower ranks indicate better performance.

Backbone	Method	Label-limited setting						Higher-label setting		
		5%	10%	20%	30%	50%	$\leq 50\%$	75%	100%	$> 50\%$
LaBraM	Frozen	5.00	2.00	5.00	6.00	6.00	4.80	5.50	6.00	5.75
	+LoRA	3.50	4.50	4.00	2.00	3.00	3.40	5.00	3.50	4.25
	+TASTEFUL	2.50	5.50	4.00	4.00	5.00	4.20	3.50	3.50	3.50
	+FixMatch	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00
	+FineSSL	6.00	3.50	2.00	3.00	4.00	3.70	4.00	5.00	4.50
	+SCOPE (Ours)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	2.00	**1.50**
	Full finetuning	3.00	4.50	5.00	5.00	2.00	3.90	2.00	**1.00**	**1.50**
CBraMod	Frozen	4.00	6.00	2.00	5.00	4.00	4.20	3.50	4.00	3.75
	+LoRA	3.50	4.50	5.00	2.00	6.00	4.20	6.00	7.00	6.50
	+TASTEFUL	3.00	3.00	4.00	4.00	3.00	3.40	3.50	3.00	3.25
	+FixMatch	7.00	7.00	7.00	7.00	7.00	7.00	7.00	6.00	6.50
	+FineSSL	6.00	**1.50**	6.00	6.00	5.00	4.90	5.00	5.00	5.00
	+SCOPE (Ours)	**1.50**	2.50	**1.00**	**1.00**	**1.00**	**1.40**	**1.00**	2.00	**1.50**
	Full finetuning	3.00	3.50	3.00	3.00	2.00	2.90	2.00	**1.00**	**1.50**
CSBrain	Frozen	2.50	4.00	2.00	5.00	3.00	3.30	4.50	4.00	4.25
	+LoRA	2.50	2.00	6.50	3.25	7.00	4.25	4.00	5.00	4.50
	+TASTEFUL	6.00	6.00	4.00	3.75	2.00	4.35	**1.00**	2.00	**1.50**
	+FixMatch	7.00	7.00	6.50	7.00	5.50	6.60	7.00	7.00	7.00
	+FineSSL	4.00	3.00	5.00	6.00	5.50	4.70	6.00	6.00	6.00
	+SCOPE (Ours)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	2.00	3.00	2.50
	Full finetuning	5.00	5.00	3.00	2.00	4.00	3.80	3.50	**1.00**	2.25
EEGMamba	Frozen	4.00	3.00	3.00	4.00	5.00	3.80	4.00	4.50	4.25
	+LoRA	3.00	3.00	2.00	2.00	**1.00**	2.20	3.00	4.00	3.50
	+TASTEFUL	**1.50**	5.00	5.00	6.00	6.00	4.70	6.50	6.50	6.50
	+FixMatch	6.50	6.50	7.00	7.00	7.00	6.80	6.50	6.50	6.50
	+FineSSL	6.50	6.50	6.00	5.00	3.00	5.40	5.00	3.50	4.25
	+SCOPE (Ours)	**1.50**	**1.00**	**1.00**	**1.00**	2.00	**1.30**	**1.00**	2.00	**1.50**
	Full finetuning	5.00	3.00	4.00	3.00	4.00	3.80	2.00	**1.00**	**1.50**
CodeBrain	Frozen	2.00	3.50	3.50	4.00	4.00	3.40	4.00	3.00	3.50
	+LoRA	5.00	4.50	2.00	7.00	7.00	5.10	5.00	5.00	5.00
	+TASTEFUL	3.50	2.50	5.00	6.00	5.00	4.40	3.00	4.00	3.50
	+FixMatch	7.00	6.50	6.50	3.00	2.50	5.10	6.50	7.00	6.75
	+FineSSL	3.50	6.50	6.50	5.00	6.00	5.50	6.50	6.00	6.25
	+SCOPE (Ours)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	2.00	**1.50**
	Full finetuning	6.00	3.50	3.50	2.00	2.50	3.50	2.00	**1.00**	**1.50**

Methods are ranked separately by mean Kappa and weighted F1 within each backbone and labeled-subject ratio, with ties assigned average ranks. Each ratio column averages the two metric ranks. The $\leq 50\%$ column averages ten ranks across 5%, 10%, 20%, 30%, and 50%; the $> 50\%$ column averages four ranks across 75% and 100%. **Bold** and underline indicate the best and second-best distinct average ranks within each backbone, respectively; tied values receive the same highlighting. Highlighting and aggregation use unrounded ranks.

G.2 Ablation Study on ProAdapter Depth L

We explore the impact of *ProAdapter* depth, defined as the number of backbone layers augmented with adapter modules, while keeping all other settings fixed. For both CSBrain (12-layer Transformer) and CodeBrain (8-layer SSM), ProAdapters are inserted in a *top-down* manner, i.e., starting from the highest layers and progressively extending toward lower layers.

Table 22, Table 23, and Figure 8 show the results. Across both backbones, performance exhibits a broadly inverted U-shaped trend with respect to adapter depth, initially improving before plateauing or declining at greater depths. When only a small number of layers are adapted, the model lacks sufficient capacity to internalize task-specific structure, resulting in underfitting. Conversely, equipping too many layers with adapters leads to degraded performance, as excessive parameter updates begin to resemble full fine-tuning, allowing noisy supervision to propagate broadly and weakening the constraint imposed by the frozen backbone.

The top-down insertion strategy is motivated by the hierarchical nature of EEG foundation models: higher layers encode more abstract, task-relevant representations, making them more suitable for feature-wise modulation, while lower layers primarily capture generic signal characteristics that should remain stable under limited supervision.

Importantly, adapter depth also introduces a practical trade-off between accuracy and efficiency. Deeper adapters incur higher computational and parameter overhead without proportional performance gains. Based on this trade-off, we adopt a depth of three *ProAdapter* layers in all experiments, which consistently achieves strong performance while preserving the efficiency advantages of parameter-efficient adaptation.

Table 22: Ablation study on *ProAdapter* depths using **CSBrain** (12-layer Transformer backbone).

ProAdapter Depth	**Kappa** (↑)	**Weighted-F1** (↑)	**Backward FLOPs (G)**
0	56.27 ± 1.49	64.98 ± 1.16	1.58
1	58.42 ± 0.69	66.77 ± 0.92	2.77
2	59.23 ± 0.70	67.23 ± 0.75	2.79
3	**61.51** ± 0.55	**69.86** ± 0.69	2.81
4	61.07 ± 0.63	69.39 ± 0.54	2.83
5	60.73 ± 2.44	70.01 ± 3.38	2.85
6	60.91 ± 2.09	69.43 ± 2.42	2.87
7	59.28 ± 0.90	67.14 ± 0.94	2.90
8	56.22 ± 5.16	63.08 ± 5.84	2.92
9	53.04 ± 2.97	61.31 ± 3.50	2.94
10	50.13 ± 3.30	57.08 ± 6.30	2.96
11	44.44 ± 3.37	53.83 ± 5.48	2.98
12	38.63 ± 10.56	50.32 ± 9.92	3.00

Table 23: Ablation study on *ProAdapter* depths using **CodeBrain** (8-layer SSM backbone).

ProAdapter Depth	**Kappa** (↑)	**Weighted-F1** (↑)	**Backward FLOPs (G)**
0	52.54 ± 1.40	61.75 ± 1.25	1.58
1	55.51 ± 1.72	64.20 ± 1.76	2.08
2	59.02 ± 1.46	67.35 ± 1.08	3.90
3	**63.46** ± 0.98	71.11 ± 0.74	5.70
4	61.57 ± 1.27	71.29 ± 1.56	7.52
5	63.02 ± 0.96	70.05 ± 1.11	9.33
6	63.23 ± 1.56	**71.65** ± 2.12	11.15
7	63.41 ± 2.45	70.33 ± 3.30	12.96
8	62.63 ± 2.43	69.43 ± 1.14	13.83

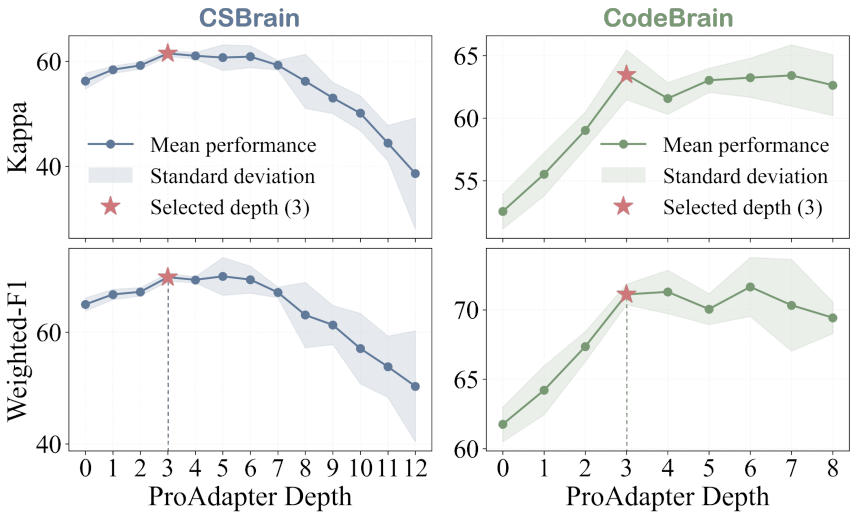

Figure 8: Ablation study on ProAdapter depths

G.3 Ablation Study on Confidence Threshold ρ

We conduct an ablation study on the confidence threshold used for pseudo-label selection in confidence-aware fusion. The threshold ρ is swept from 0 to 0.95 with a step size of 0.05, where only pseudo-labels whose confidence exceeds the threshold are retained for training.

As shown in Table 24 and Figure 9, model performance exhibits a non-monotonic trend with respect to the confidence threshold. The confidence weighting is important for EEG data because it shows high inter-subject variability, under which a single supervision source, even expert annotation (Jing et al., 2020), can produce supervision with highly heterogeneous reliability across samples. Incorporating unreliable supervision can misdirect optimization (Han et al., 2018), while overly confident supervision biases learning toward easy cases, limiting exposure to informative but ambiguous samples and impairing generalization (Wu et al., 2022; Jain et al., 2024; Sanyal et al., 2025). An intermediate threshold yields the best performance by balancing pseudo-label quantity and reliability. In this regime, the model benefits from additional supervision while avoiding excessive noise, resulting in more stable and consistent improvements across evaluation metrics. This behavior aligns with the design motivation of confidence-aware fusion, which explicitly regulates how unreliable supervision influences adaptation under limited labeled data.

Based on this trade-off, we select a fixed confidence threshold $\rho = 0.5$ for all experiments, which consistently achieves strong performance across backbones and tasks.

Table 24: Ablation study on confidence threshold for two representative EEG foundation model backbones. We retain pseudo-labeled samples whose confidence is strictly greater than the threshold.

Confidence	CSBrain (Transformer)		CodeBrain (SSM)	
	Kappa (↑)	**Weighted-F1** (↑)	**Kappa** (↑)	**Weighted-F1** (↑)
0.00	57.74 ± 3.52	66.54 ± 3.65	56.24 ± 3.48	65.78 ± 4.03
0.05	59.46 ± 2.63	67.55 ± 2.92	58.19 ± 2.65	66.32 ± 2.72
0.10	60.03 ± 1.43	67.96 ± 1.28	60.23 ± 1.27	68.71 ± 1.36
0.15	60.15 ± 1.23	68.30 ± 1.29	61.25 ± 0.98	68.74 ± 1.26
0.20	60.26 ± 1.12	68.15 ± 0.99	61.57 ± 0.86	68.56 ± 0.64
0.25	59.78 ± 1.61	69.08 ± 1.73	61.88 ± 1.01	68.87 ± 0.96
0.30	59.97 ± 1.13	69.60 ± 1.72	62.27 ± 0.72	69.42 ± 0.95
0.35	60.38 ± 1.18	69.96 ± 1.12	62.74 ± 0.63	71.02 ± 0.78
0.40	61.27 ± 0.95	69.01 ± 0.43	62.73 ± 1.08	70.85 ± 1.22
0.45	61.20 ± 0.68	69.52 ± 0.70	**63.53** ± 1.19	**71.31** ± 1.49
0.50	61.51 ± 0.55	**69.86** ± 0.69	63.46 ± 0.98	71.11 ± 0.74
0.55	**61.84** ± 0.43	69.66 ± 0.52	63.04 ± 1.43	70.94 ± 1.57
0.60	61.09 ± 0.50	69.12 ± 0.84	62.60 ± 0.97	70.38 ± 1.02
0.65	61.72 ± 0.81	69.50 ± 0.95	61.89 ± 1.14	69.46 ± 1.38
0.70	61.23 ± 1.05	68.97 ± 1.24	61.15 ± 1.82	67.56 ± 1.71
0.75	60.52 ± 1.17	68.15 ± 1.02	59.85 ± 1.37	66.60 ± 1.80
0.80	60.53 ± 1.68	67.94 ± 1.85	58.98 ± 1.49	66.62 ± 1.81
0.85	59.68 ± 1.95	66.95 ± 2.55	57.25 ± 1.85	64.34 ± 1.76
0.90	57.65 ± 2.85	65.35 ± 2.63	56.90 ± 2.64	64.77 ± 3.14
0.95	55.93 ± 3.14	63.76 ± 3.78	56.33 ± 2.36	64.25 ± 2.64

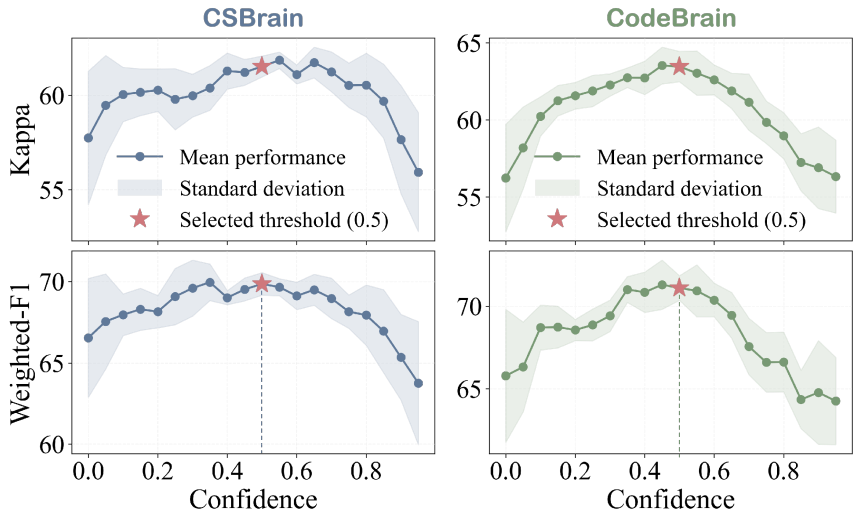

Figure 9: Ablation study on confidence threshold for two representative EEG foundation model backbones.

G.4 Ablation Study on the Number of Prototype per Class M

We investigate the effect of the number of prototypes learned per class, M, which controls the granularity of class-level structure used in both prototype-based prediction and prototype-conditioned adaptation.

Table 25 and Figure 10 show the result of ablation study on prototype number per class. Using too few prototypes limits the model's capacity to capture intra-class variability, forcing heterogeneous samples to collapse into a single representative and leading to underfitting. In contrast, an excessive number of prototypes increases structural redundancy and introduces unstable assignments under limited supervision, which may amplify noise in prototype learning and weaken the reliability of prototype-conditioned modulation.

As shown in the following results, an intermediate number of prototypes per class consistently yields the best performance, indicating a favorable balance between expressive class structure and stable optimization. This observation supports the design choice of learning a compact yet diverse set of class prototypes to guide adaptation under label-limited conditions.

Table 25: Ablation study on prototype number per class for two representative EEG foundation model backbones.

Prototype Number per Class	**CSBrain (Transformer)**		**CodeBrain (SSM)**	
	Kappa (↑)	**Weighted-F1** (↑)	**Kappa** (↑)	**Weighted-F1** (↑)
1	60.38 ± 0.44	68.60 ± 0.77	60.35 ± 0.81	67.95 ± 0.76
2	60.08 ± 1.21	68.08 ± 1.30	62.44 ± 1.15	69.99 ± 1.79
3	**61.51** ± 0.55	**69.86** ± 0.69	**63.46** ± 0.98	**71.11** ± 0.74
4	56.89 ± 1.00	65.77 ± 1.33	57.02 ± 1.09	66.01 ± 1.12
5	59.72 ± 2.61	66.64 ± 3.42	60.17 ± 2.69	68.49 ± 2.83

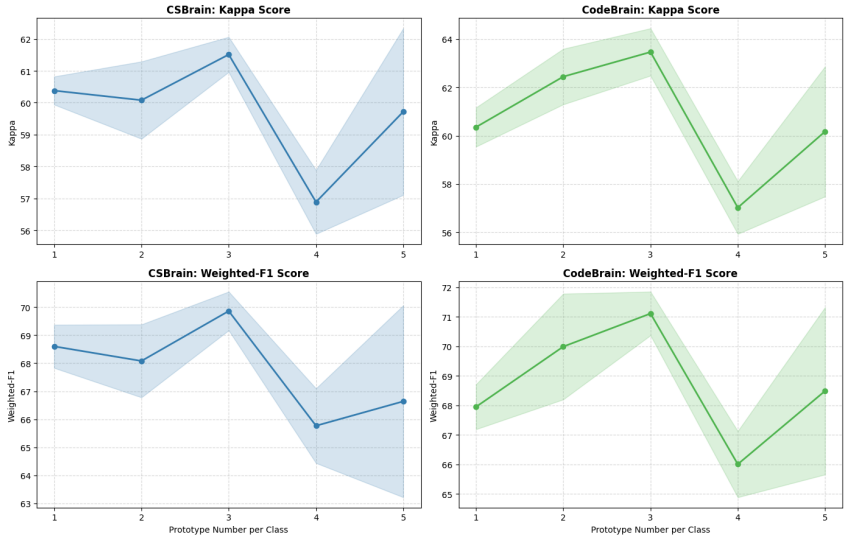

Figure 10: Ablation study on prototype number per class

G.5 Ablation Study on ETF Loss Weights λ_{ETF}

We ablate the weight of the Equiangular Tight Frame (ETF) loss applied in the task-prior network (TPN), which controls the strength of geometric class separation during supervised training. Specifically, we vary the ETF loss weight λ_{ETF} over $0, 0.05, 0.1, 0.2, 0.5$, where $\lambda_{\text{ETF}} = 0$ disables ETF regularization entirely, while all other components remain unchanged.

As shown in Table 26 and Figure 11, disabling ETF regularization or assigning it an excessively small weight provides insufficient geometric constraints on the task-prior classifier, resulting in weaker class separation and consequently less informative guidance for subsequent prototype learning and confidence estimation. In contrast, overly large values of λ_{ETF} place excessive emphasis on the predefined class geometry, which can restrict task-specific fitting under limited supervision.

Overall, an intermediate value of $\lambda_{\text{ETF}} = 0.1$ consistently achieves the best balance between geometric regularization and task-specific adaptation and is therefore adopted in all experiments.

Table 26: Ablation study on ETF Loss Weights for two representative EEG foundation model backbones.

ETF Loss Weights (λ_{ETF})	**CSBrain (Transformer)**		**CodeBrain (SSM)**	
	Kappa (↑)	**Weighted-F1** (↑)	**Kappa** (↑)	**Weighted-F1** (↑)
0	60.04 ± 2.39	68.15 ± 2.07	61.48 ± 1.72	68.87 ± 1.35
0.05	61.44 ± 1.15	68.99 ± 1.37	62.77 ± 1.23	70.93 ± 1.45
0.1	**61.51** ± 0.55	**69.86** ± 0.69	**63.46** ± 0.98	**71.11** ± 0.74
0.2	60.74 ± 0.87	68.73 ± 1.02	62.58 ± 1.43	70.40 ± 1.26
0.5	59.93 ± 2.79	67.64 ± 2.67	61.51 ± 1.52	69.26 ± 1.31

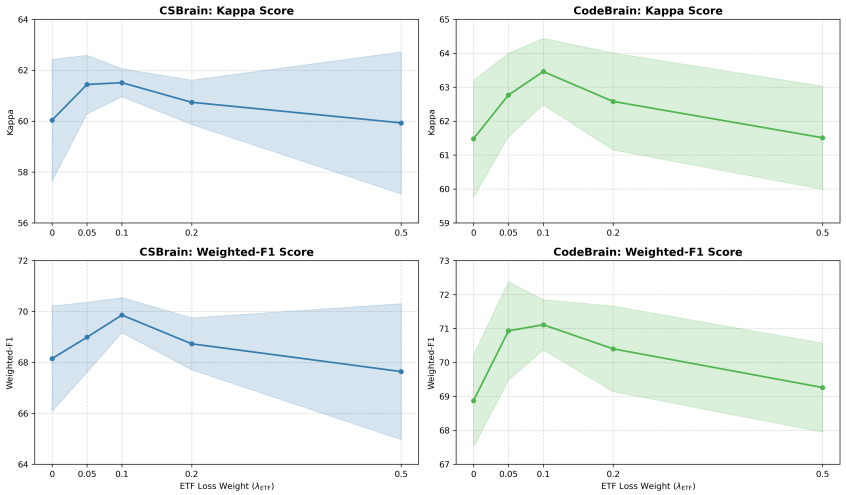

Figure 11: Ablation study on ETF Loss Weights.

G.6 Ablation Study on Pseudo-labeled Data Ratio

We study the effect of the pseudo-labeled data ratio, which controls the relative exposure of the model to pseudo-labeled samples during training. Specifically, this ratio determines how many pseudo-labeled samples are introduced after each batch of labeled data, while keeping the batch size fixed at 64 throughout all experiments. Table 27 and Figure 12 show the results of the ablation study on the pseudo-labeled data ratio. For example, a ratio of 0.5 corresponds to 32 pseudo-labeled samples following one labeled batch. This design enforces a controlled balance between reliable supervision from labeled data and additional signals from pseudo-labeled data. When the pseudo-labeled ratio is too high, noisy pseudo-labels may dominate the optimization process, causing the model to drift away from the true task objective. Conversely, an overly small ratio underutilizes unlabeled data and limits the benefit of confidence-aware pseudo-labeling.

By fixing the batch size and varying only the pseudo-labeled ratio, we ensure that each labeled batch serves as a stable anchor that regularly re-aligns the model toward ground-truth supervision. The following ablation results demonstrate that an intermediate ratio yields the best trade-off between leveraging unlabeled data and preserving stable, label-driven adaptation.

Table 27: Ablation study on pseudo-labeled data ratio for two representative EEG foundation model backbones.

Pseudo-labeled Data Ratio	CSBrain (Transformer)		CodeBrain (SSM)	
	Kappa (↑)	**Weighted-F1** (↑)	**Kappa** (↑)	**Weighted-F1** (↑)
0.5	59.18 ± 0.55	66.64 ± 0.83	58.15 ± 0.64	66.68 ± 0.51
1	60.24 ± 0.82	67.88 ± 0.75	60.68 ± 1.08	68.40 ± 1.89
1.5	61.28 ± 0.79	**70.36** ± 1.22	62.24 ± 1.14	70.45 ± 1.24
2	**61.51** ± 0.55	69.86 ± 0.69	**63.46** ± 0.98	71.11 ± 0.74
2.5	61.18 ± 0.73	68.27 ± 0.99	63.41 ± 1.67	**71.34** ± 1.71
3	60.41 ± 1.82	67.63 ± 2.54	63.09 ± 2.12	70.28 ± 2.85

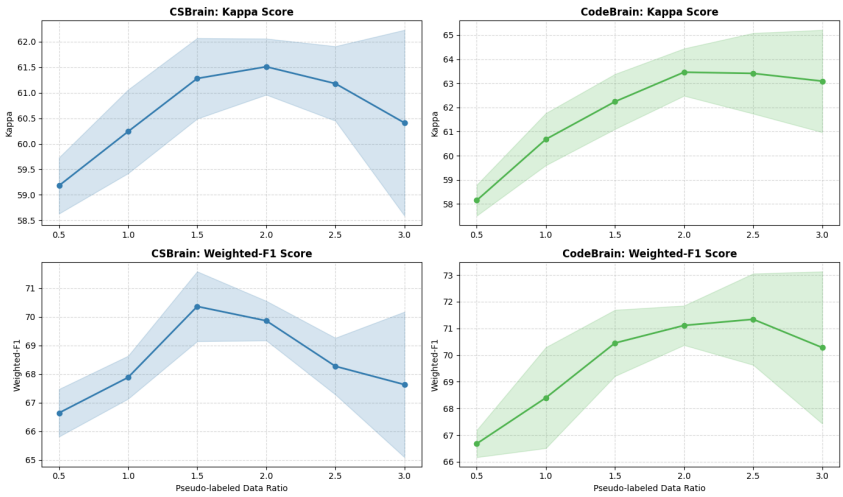

Figure 12: Ablation Study on Pseudo-labeled Data Ratio.

H Efficiency Analysis

We provide a comprehensive analysis of the computational efficiency of *SCOPE* from two perspectives. First, we evaluate the efficiency of EFM encoder adaptation by comparing trainable parameters, backward FLOPs, GPU memory usage, and training time with representative parameter-efficient adaptation and semi-supervised learning baselines. Second, we assess the end-to-end computational cost of the complete *SCOPE* pipeline, including all task-specific supervision construction and EFM adaptation stages. All experiments in this section use a batch size of 16.

H.1 Backbone Adaptation Efficiency

We provide a detailed breakdown of trainable parameters, Backward FLOPs, training time per step, and GPU memory usage across all semi-supervised learning and adapter methods. Figure 13 and Table 28 show the information about the baseline and our models under a batch size of 16. FixMatch combines pseudo-labeling with consistency regularization between weakly and strongly augmented unlabeled samples.

We further visualize the trade-off between performance and computational efficiency in Figure 14. The horizontal axis denotes training time per step (lower is faster), and the vertical axis denotes Kappa (higher is better). The bubble size indicates GPU memory consumption. SCOPE achieves a favorable trade-off between predictive performance and computational efficiency. On both backbones, it achieves the highest Kappa and the shortest training time per step among the compared methods, while requiring substantially less GPU memory than full fine-tuning.

Table 28: Computational cost comparison on the ISRUC dataset. Lower is better for all metrics.

Backbone	Method	Trainable Encoder Params (%)	Backward FLOPs	GPU Memory (MB)	Time / Step (s)
CSBrain	LoRA	5.00%	1.76G	1,034	1.53
	TASTEFUL	6.89%	4.99G	3,875	1.78
	Ours	4.21%	2.81G	3,141	1.49
	FixMatch	100%	10.10G	15,666	6.23
	FineSSL	71.58%	2.62G	5,995	2.89
	Full Fine-tune	100%	19.24G	20,557	3.02
CodeBrain	LoRA	1.91%	2.25G	3,577	1.60
	TASTEFUL	4.70%	3.47G	3,769	1.86
	Ours	2.50%	5.70G	5,530	1.46
	FixMatch	100%	20.76G	25,227	6.15
	FineSSL	59.54%	6.50G	13,990	5.87
	Full Fine-tune	100%	26.60G	21,020	2.72



Figure 13: Comparison of computational efficiency across different adaptation methods on ISRUC with a batch size of 16.

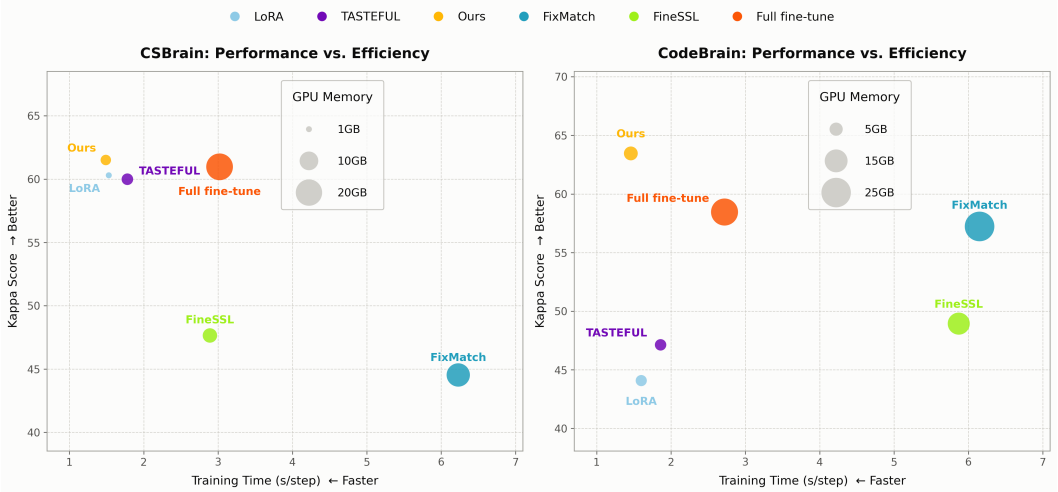

Figure 14: Performance–efficiency trade-off on ISRUC with a batch size of 16. Kappa is plotted against training time per step, with bubble size indicating GPU memory consumption.

H.2 Overall Computational Cost

We extend the backbone-only analysis to the complete *SCOPE* pipeline, including TPN training, prototype generation, and EFM adaptation. We evaluate CodeBrain (15.2M parameters) and CSBrain (9.3M parameters) on ISRUC, SEED, and Mental Arithmetic with 30% labeled training subjects. *SCOPE* uses the labeled training subjects together with the remaining unlabeled training subjects, whereas full fine-tuning uses only the labeled subset. Total training time and GFLOPs are summed across the three stages; peak GPU memory is their maximum. Each backbone's reported total includes the full cost of supervision construction, without amortization across backbones.

Table 29 shows that *SCOPE* reduces training time by 1.7%-28.5%, peak memory by 46.6%-86.7%, and GFLOPs by 47.1%-92.2% relative to full fine-tuning across the six evaluated settings.

Table 29: End-to-end computational cost with 30% labeled training subjects. SCOPE includes all supervision construction and adaptation stages; Full FT denotes full fine-tuning. Lower values indicate lower cost.

		ISRUC		**SEED**		**Mental Arithmetic**	
Backbone	**Metric**	**SCOPE**	**Full FT**	**SCOPE**	**Full FT**	**SCOPE**	**Full FT**
CodeBrain (15.2M)	Total training time (min)	62.2	72.5	10.8	13.5	4.29	6.00
	Peak memory (MB)	5,530	21,020	1,034	2,255	575	1,076
	Total GFLOPs	24.43	55.30	1.46	2.76	0.51	1.07
CSBrain (9.3M)	Total training time (min)	39.8	40.5	5.8	6.3	3.2	3.3
	Peak memory (MB)	3,141	20,557	803	6,041	430	2,309
	Total GFLOPs	7.00	43.41	0.51	3.41	0.16	2.06

Table 30 separates supervision construction from backbone adaptation. EFM adaptation accounts for most training time, while TPN training and prototype generation together take 6.00 minutes on ISRUC, 0.34 minutes on SEED, and 0.29 minutes on Mental Arithmetic. Shared-stage overhead is therefore most noticeable on ISRUC. Across all six evaluated settings, peak GPU memory occurs during EFM adaptation. These observations explain why adaptation-stage savings alone do not fully characterize end-to-end efficiency.

Table 30: Stage-wise computational costs with 30% labeled training subjects. Shared stages are reported once per dataset; backbone-specific adaptation costs exclude these stages.

		Shared stages		**EFM adaptation**	
Dataset	**Metric**	**TPN**	**Prototype generation**	**CodeBrain**	**CSBrain**
ISRUC (6 channels, 30 seconds)	Time (min)	5.25	0.75	56.2	33.8
	Peak memory (MB)	2,482	1,557	5,530	3,141
	GFLOPs	1.81	0.72	21.90	4.47
SEED (62 channels, 4 seconds)	Time (min)	0.28	0.06	10.5	5.5
	Peak memory (MB)	61	51	1,034	803
	GFLOPs	0.17	0.08	1.21	0.26
Mental Arithmetic (20 channels, 5 seconds)	Time (min)	0.26	0.03	4.0	2.9
	Peak memory (MB)	50	41	575	430
	GFLOPs	0.03	0.01	0.47	0.12

TPN training and prototype generation can be reused across EFM backbones when the task, data split, and supervision-construction configuration are fixed. Such reuse can further reduce the aggregate cost of evaluating multiple backbones, beyond the per-backbone savings reported here.

I PSEUDO-LABEL QUALITY ANALYSIS

WITH SINKHORN CONSTRAINT

We analyze the quality of pseudo-labels on the ISRUC dataset, which is a multi-class and highly imbalanced EEG benchmark. To provide a comprehensive assessment under class imbalance, we report balanced accuracy (BAcc), Cohen's Kappa, and weighted F1 score. These metrics together characterize both per-class discrimination and overall agreement, making them well suited for evaluating pseudo-label reliability in this setting. Table 31 shows the pseudo-label quality under confidence-based filtering on the ISRUC dataset, and Figure 15 shows the detailed confusion matrix under different confidence levels in the table. As ρ increases, the number of agreement samples between TPN and the prototype decreases. Conversely, the kappa and weighted-F1 of their agreement samples improve. In other words, while increasing confidence results in high-quality pseudo-labels, their quantity diminishes. Reviewing the F1 results for different sleep classifications shows that as ρ increases, both models perform better in classifying Wake and N3 states, whereas classification results for N1 and N2 show a declining trend. The REM state remains largely unchanged with variations in ρ, indicating that under higher confidence conditions, the model tends to produce results for more prevalent types.

Table 31: Pseudo-label quality under confidence-based filtering on ISRUC. For a threshold ρ, we retain pseudo-labeled samples with confidence $\geq \rho$. We report coverage and pseudo-label quality in terms of Cohen's Kappa, weighted F1, and per-class F1.

				Per-class F1				
ρ	**Coverage (%)**	**Kappa**	**Weighted-F1**	**Wake**	**N1**	**N2**	**N3**	**REM**
0.00	72.66	62.49	70.67	87.01	41.57	69.68	78.27	53.93
0.05	72.62	62.50	70.68	87.04	41.54	69.68	78.29	53.93
0.10	72.47	62.59	70.76	87.12	41.51	69.76	78.37	53.90
0.15	72.02	62.78	70.93	87.22	41.42	69.91	78.48	53.96
0.20	70.75	63.32	71.42	87.57	41.07	70.31	78.68	54.13
0.25	68.57	64.15	72.18	88.10	39.50	71.00	78.92	54.64
0.30	65.26	65.28	73.21	88.76	37.46	71.83	79.18	54.91
0.35	61.06	67.02	74.76	89.52	36.06	73.02	79.72	55.68
0.40	56.53	68.84	76.36	90.35	34.10	74.09	80.59	55.91
0.45	51.97	70.56	77.85	91.28	31.35	74.54	81.69	56.62
0.50	47.24	72.15	79.25	92.20	29.66	74.36	83.12	55.66
0.55	42.09	74.42	81.08	93.15	26.43	74.68	85.11	56.28
0.60	37.34	76.47	82.84	93.95	24.31	74.60	87.12	55.84
0.65	33.36	78.12	84.30	94.62	20.57	73.66	88.89	55.81
0.70	29.68	79.65	85.70	95.28	17.03	72.14	90.34	55.47
0.75	26.12	81.16	87.03	95.69	13.76	70.45	91.95	54.37
0.80	22.80	82.93	88.51	96.21	10.63	68.34	93.60	53.81
0.85	19.35	84.84	90.09	96.70	5.94	64.55	95.20	53.38
0.90	15.56	86.09	91.11	97.13	5.76	53.23	96.32	52.01
0.95	10.87	88.15	92.46	97.39	2.47	38.80	97.68	53.57

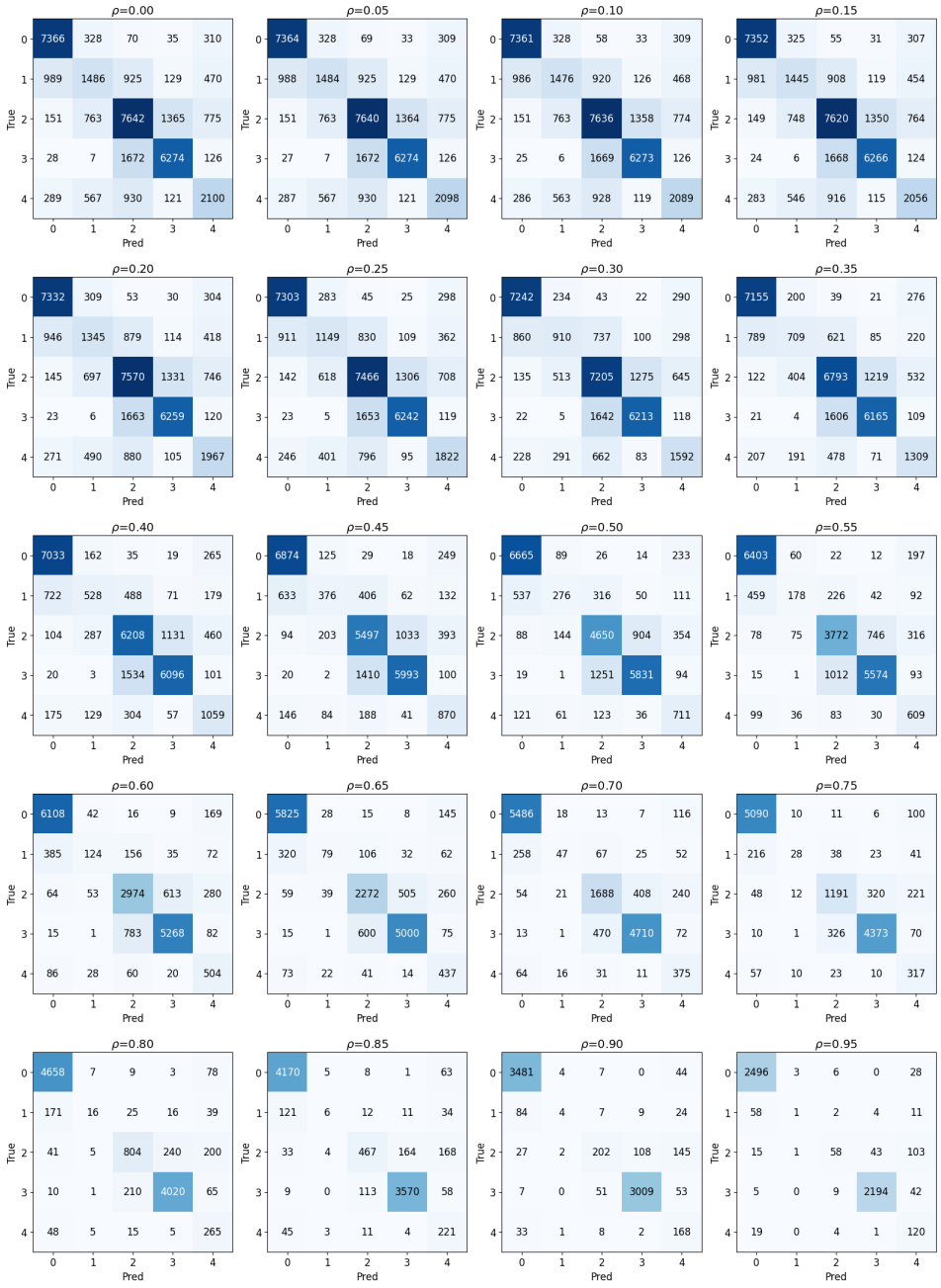

Figure 15: Confusion matrix under different confidence levels.

WITHOUT SINKHORN CONSTRAINT

For comparison, we also conducted an ablation experiment without the Sinkhorn-Knopp algorithm. Table 32 shows the pseudo-label quality under the ablation study of removing Sinkhorn-Knopp algorithm on the ISRUC dataset. Figure 16 shows the detailed confusion matrix under different confidence levels. Compared to the version with the Sinkhorn-Knopp algorithm, the proportion of Coverage is lower at low confidence levels. Although it achieves better overall Kappa and F1 scores, the classification results are extremely poor for N1, as the prototype model hardly predicts any N1s. This label quality introduces new difficulties for subsequent fine-tuning because the model does not encounter enough N1 samples. After incorporating the Sinkhorn-Knopp algorithm, the prototype model can more actively predict rare samples like N1. While this comes at a cost to some accuracy in other categories, due to their larger representation in the dataset, there won't be a situation where categories are missing from pseudo-label datasets. Therefore, this proves that adding constraints in prototype learning can better assist pseudo-label training.

Table 32: Ablation study on removing Sinkhorn–Knopp algorithm on pseudo-label learning. Pseudo-label quality under confidence-based filtering on ISRUC. For a threshold ρ, we retain pseudo-labeled samples with confidence $\geq \rho$.

				Per-class F1				
ρ	**Coverage (%)**	**Kappa**	**Weighted-F1**	**Wake**	**N1**	**N2**	**N3**	**REM**
0.00	55.23	71.70	75.68	89.98	2.09	77.90	85.16	71.13
0.05	55.23	71.70	75.68	89.98	2.09	77.90	85.16	71.13
0.10	55.22	71.71	75.68	89.99	2.10	77.90	85.17	71.11
0.15	55.11	71.78	75.77	90.03	2.11	77.94	85.20	71.18
0.20	54.70	72.04	76.03	90.15	2.16	78.13	85.27	71.32
0.25	53.81	72.47	76.54	90.34	2.28	78.46	85.36	71.42
0.30	52.57	73.11	77.26	90.63	2.46	78.90	85.50	71.64
0.35	50.75	74.02	78.23	91.10	2.37	79.56	85.78	71.82
0.40	48.43	75.05	79.25	91.69	2.53	80.24	86.15	72.08
0.45	46.14	76.10	80.29	92.17	2.70	80.93	86.60	72.24
0.50	43.72	77.19	81.33	92.76	2.72	81.48	87.16	72.50
0.55	41.11	78.43	82.51	93.38	2.94	82.08	87.93	72.28
0.60	38.46	79.57	83.57	93.91	3.16	82.49	88.66	72.46
0.65	35.57	80.82	84.76	94.53	3.21	82.80	89.49	72.27
0.70	32.66	82.03	85.97	95.20	3.64	82.81	90.22	72.05
0.75	29.44	83.55	87.44	95.89	4.29	82.53	91.24	72.91
0.80	26.03	84.61	88.54	96.31	4.90	81.41	92.31	71.79
0.85	22.41	86.19	90.15	96.99	4.86	79.69	93.52	71.73
0.90	18.41	88.08	91.90	97.39	5.95	77.35	95.29	67.78
0.95	13.46	90.75	94.18	98.35	7.79	68.44	96.94	62.91

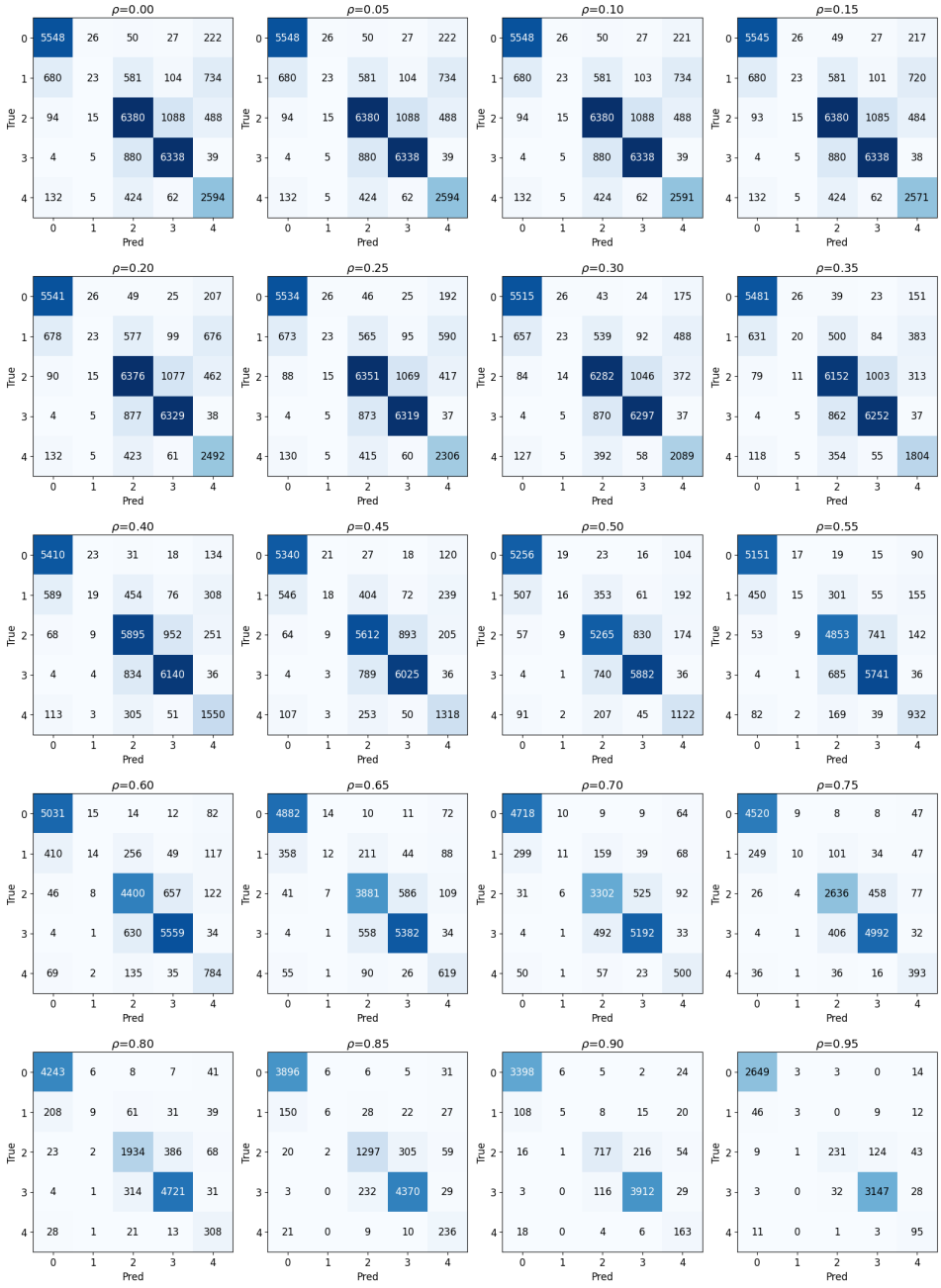

Figure 16: Confusion matrix at different confidence levels without using the Sinkhorn-Knopp algorithm.

J PSEUDOCODE FOR TRAINING PROCEDURE

This section presents the pseudocode of the overall training procedure to facilitate understanding of the workflow. The pseudocode summarizes the key steps involved in external supervision construction, pseudo-label generation, confidence-aware fusion, and parameter update, while omitting implementation-specific details.

Algorithm 1 Training Procedure of SCOPE

Require: Labeled dataset $\mathcal{D}_l = \{(X_i, y_i)\}_{i=1}^{N_l}$ with $y_i \in \{1, \ldots, K\}$, K is the number of class;
Unlabeled dataset $\mathcal{D}_u = \{X_j\}_{j=1}^{N_u}$;
Pretrained foundation model parameters θ_{FM};
Pre-defined confidence threshold ρ;
Initialized task-prior network (TPN) parameters ProAdapter and classifier parameters $\theta_{TPN}, \theta_{PA}, \theta_C$.
Ensure: Optimized parameters $\theta_{PA}^\star, \theta_C^\star$.
(a) Task Prior Network Training for Inter-class Separability
Train θ_{TPN} on $\mathcal{D}_l$ using ETF-regularized cross-entropy loss $\mathcal{L}_{\text{TPN}}$, obtaining $\theta_{\text{TPN}}^\star$
Infer prior labels on $\mathcal{D}_u$: $\hat{y}_j^{\text{prior}} \leftarrow \arg\max_c \, f_{\theta_{\text{TPN}}^\star}(X_j)_c$
(b) Prototype-Based Clustering for Intra-class Variability
Initialize prototype set $\mathcal{P}$ via k-means on unlabeled embeddings $z_j \leftarrow f_{\theta_{TPN}^\star}(X_j) \in \mathbb{R}^d$.
Update $\mathcal{P}$ by minimizing prototype loss $\mathcal{L}_P$ on labeled dataset $\mathcal{D}_l$.
Derive prototype label $\hat{y}_j^{\text{proto}} \leftarrow \arg\max_{p \in \mathcal{P}} \, \text{similarity}(z_j, p)$.
(c) Confidence-Aware Fusion (Parameter-Free)
Set pseudo-labels $\hat{y}_j \leftarrow \hat{y}_j^{\text{prior}}$.
Compute confidence score γ_j using Eq. (8) and set $m_j \leftarrow \mathbb{I}(\hat{y}_j^{\text{prior}} = \hat{y}_j^{\text{proto}})\mathbb{I}(\gamma_j > \rho)$.
(d) ProAdapter Training
Sample labeled and unlabeled mini-batches $\mathcal{B}_l \subseteq \mathcal{D}_l$ and $\mathcal{B}_u \subseteq \mathcal{D}_u$.
$\mathcal{L}_{\text{sup}} \leftarrow \frac{1}{|\mathcal{B}_l|} \sum_{(X_i, y_i) \in \mathcal{B}_l} \text{CE}\big(f_{(\theta_{PA}, \theta_C)}(X_i), y_i\big)$
$\mathcal{L}_{\text{pseudo}} \leftarrow \frac{1}{|\mathcal{B}_u|} \sum_{X_j \in \mathcal{B}_u} m_j \gamma_j \, \text{CE}\big(f_{(\theta_{PA}, \theta_C)}(X_j), \hat{y}_j\big)$
Update (θ_{PA}, θ_C) by minimizing $\mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{pseudo}}$.

K IMPLEMENTATION DETAILS

Our reported full-fine-tuning baselines follow the original backbone-specific training configurations and undergo hyperparameter tuning for the downstream tasks. The reported results therefore characterize tuned full fine-tuning under limited supervision, rather than runs with arbitrarily chosen hyperparameters. The EFM training hyperparameters are summarized in Table 34.

All *SCOPE*-specific hyperparameters are shared across different backbones and datasets. The Task Prior Network (TPN) is implemented as a compact two-block EEG convolutional network that extracts temporal patterns and electrode-wise spatial dependencies, followed by a linear classifier for task-prior prediction. The parameters used in the prototype clustering is shown in Table 33. For the EFM adaptation stage, we follow the original training configurations of each backbone whenever applicable.

Table 33: Configuration of the Prototype Learning

Category	Parameter	Default
Basic Training	Epochs	50
	Batch size	64
	Learning rate	5×10^{-4}
	Min learning rate	1×10^{-6}
	Weight decay	5×10^{-3}
	Optimizer	AdamW
	Optimizer momentum	β_1, β_2 = 0.9, 0.999
	Label smoothing	0.05
	Learning rate schedule	CosineAnnealing
ETF (Equiangular Tight Frame)	ETF λ	0.1
	ETF scale	10.0
Prototype	Num prototypes per class	3
	Prototype learning rate	0.001
	Init method	KMeans
	Kmeans max per class	5000
	Kmeans N_{init}	10
	Prototype τ	10.0
	Prototype temperature	1.0
	Prototype uniform mix	0.05
Sinkhorn–Knopp	Sinkhorn iters	3
	Sinkhorn ϵ	0.05
	Sinkhorn temperature	0.1
Dempster–Shafer Fusion	eps	1×10^{-12}
	Confidence threshold ρ	0.5

Table 34: Training Hyperparameters of Backbone EFMs

Hyperparameter	LaBraM	CBraMod	CSBrain	EEGMamba	CodeBrain
Warm-up Epoch	10	10	10	10	10
Number of Epochs	50	50	50	50	30
Batch Size	64	64	64	64	64
Pseudo Label Batch Size	64	64	64	64	64
Adapter L Layers	3	3	3	3	3
Learning Rate	1×10^{-5}	1×10^{-4}	1×10^{-4}	1×10^{-4}	5×10^{-5}
Minimum Learning Rate	1×10^{-6}	1×10^{-6}	1×10^{-6}	1×10^{-6}	1×10^{-6}
Weight Decay	0.0001	0.001	0.001	0.001	0.001
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Learning Rate Scheduler	CosineAnnealing	CosineAnnealing	CosineAnnealing	CosineAnnealing	CosineAnnealing
Dropout Rate	0.1	0.1	0.1	0.1	0.1

L CODE ACCESS

An anonymized GitHub repository has been used to publicly share the codebase: `https://anonymous.4open.science/r/SCOPE-A0F2`.

M LIMITATIONS

This work focuses on label-limited adaptation of EEG foundation models under cross-subject settings. Several directions remain for extending its scope.

First, *SCOPE*'s advantage over full fine-tuning gradually diminishes as the proportion of labeled training subjects increases. Under full supervision, no unlabeled training subjects remain for its semi-supervised objective, while constrained adaptation may limit the use of the backbone's capacity. Our results suggest that full fine-tuning becomes a more suitable option in this regime. Extending prototype-guided adaptation to full-parameter updates is a direction for future work.

Second, as a plug-in adaptation framework, *SCOPE* builds upon existing EEG foundation models, and its capabilities depend on the pretrained representations and capacity of the underlying backbone. The current framework also assumes that labeled and unlabeled training subjects share the same task label space. Extensions to partially overlapping label spaces, open-set conditions, and cross-task adaptation would broaden its applicability.

Third, our experiments focus on classification tasks under offline subject-level adaptation, matching the label-limited setting studied in this work. Regression and streaming scenarios remain outside the current evaluation scope, rather than established limitations of the plug-in design. Extending prototype-based supervision to continuous targets and supporting online updates are natural directions for broadening its applicability.