HiERA: Hierarchical Domain-aware Reusable Adaptation for Continual Personalization of EEG Foundation Models

Anonymous authors
Paper under double-blind review

Abstract

EEG foundation models (EFMs) generalize well across EEG analysis and brain-computer interface (BCI) tasks, yet continually personalizing EFMs to a growing stream of subjects remains costly: full fine-tuning is computationally expensive, while per-subject LoRA adapters cause storage to grow rapidly with the number of subjects. We observe that shallow EFM layers encode more domain-level variation shared across subjects, whereas deeper layers increasingly capture subject-specific characteristics. Building on this observation, we propose **HiERA**, a hierarchical adaptation framework that employs a domain-aware Tensor Ring transformation in shallow layers and a compact, reusable adapter repository in deeper layers. New subjects can reuse existing adapters when appropriate, avoiding unnecessary parameter growth. Experiments on multiple public EEG datasets and EFMs show that HiERA maintains competitive personalization performance while substantially reducing trainable and stored parameters compared with other PEFT baselines. Code will be released upon acceptance.

1 Introduction

Pre-trained on large-scale and heterogeneous electroencephalogram (EEG) corpora, EEG foundation models (EFMs) have recently shown strong transferability across tasks and datasets (Kuruppu et al., 2026). However, substantial inter-subject variability remains a key challenge for deploying EFMs in practical brain-computer interfaces (BCIs), making subject personalization essential. Hence, LoRA (Hu et al., 2022) becoming a common parameter-efficient fine-tuning (PEFT) solution for this. (Lee et al., 2025; Dai et al., 2026).

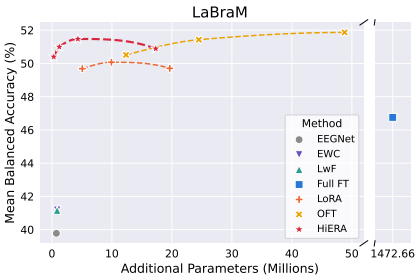


Figure 1: Balanced accuracy vs. additional trainable parameters.

In real-world deployment, users do not arrive all at once: they join *continually* and may come from different domains. For instance, a hospital may continuously receive patients for sleep monitoring or cognitive evaluation using different EEG recording devices. We refer to this setting as **continual personalization**, where an EFM must efficiently adapt to newly arriving subjects while keeping the additional adaptation cost low. A straightforward solution is to train an independent LoRA adapter for each new subject (Lee et al., 2025), but the number of trainable parameters grows rapidly with the subject population. This raises a broader question: *can we maintain high EFM performance while minimizing the trainable parameters required for continual personalization?* Answering this question requires a deeper understanding of how EEG information is represented within EFMs, rather than naively applying PEFT strategies developed for other domains. (Fig. 1 shows that our HiERA outperforms other PEFT methods, particularly under tight parameter budgets.)

We first ask: *does every subject need an individual adapter at every layer?* Our layer-wise analysis reveals a clear hierarchy: shallow layers are more domain-aware, while deeper layers become increasingly subject-specific. This suggests that subject-specific adaptation is largely redundant in shallow layers and can instead be shared within each domain. We then ask *whether such sharing can*

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.

accommodate the growing variation across subjects. Our analysis shows that the required weight update becomes increasingly high-rank as more subjects are incorporated, suggesting that fixed-rank LoRA may lack sufficient capacity for domain-shared adaptation.

Motivated by these findings, we propose **HiERA**, a hierarchical domain-aware and reusable adaptation framework for continual personalization of EFMs. In shallow layers, HiERA employs a *domain-aware transformation* $\boldsymbol{T}$ parameterized by Tensor Ring (TR) decomposition (Zhao et al., 2016). In deeper layers, HiERA maintains a lightweight *adapter repository* that enables adapters to be reused across subjects and domains. When a new subject arrives, only a small amount of additional parameters is required, while a new TR transformation is instantiated only when a new domain appears. As a result, HiERA substantially reduces the per-subject parameter increment while retaining sufficient capacity for domain-level adaptation. Extensive experiments across multiple EFMs and datasets show that HiERA achieves comparable or better performance than per-subject LoRA using only 13% of its trainable parameters on REVE and 24% on LaBraM.

2 Preliminaries and Motivation

2.1 Continual Personalization Problem Setup

We consider a continual personalization setting for an EFM $\mathcal{M}$, where subjects arrive sequentially in a stream and $\mathcal{M}$ continuously adapts to each incoming subject. Specifically, let there be N EEG datasets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N$, where each dataset $\mathcal{D}_i$ comprises N_i individual subjects, denoted as $\mathcal{D}_i = \{S_1^i, S_2^i, \ldots, S_{N_i}^i\}$. Within each dataset, all subjects share the same EEG acquisition configuration and label space.

During continual personalization, subjects from different datasets arrive sequentially in a randomized order to form a cross-dataset subject stream, e.g., $S_{10}^1, S_5^3, \ldots, S_6^2$. Given an incoming subject S_j^i, its data are divided into a training set $\mathcal{D}_{j,\text{train}}^i$, a validation set $\mathcal{D}_{j,\text{val}}^i$, and a test set $\mathcal{D}_{j,\text{test}}^i$. We adapt $\mathcal{M}$ on $\mathcal{D}_{j,\text{train}}^i$ to obtain a personalized model $\mathcal{M}_j^i$, selecting the model checkpoint that achieves the best performance on $\mathcal{D}_{j,\text{val}}^i$. The resulting personalized model is then evaluated on $\mathcal{D}_{j,\text{test}}^i$. After personalization, the subject-specific adapter and classifier are retained so that previously encountered subjects can be directly served when their data are revisited.

2.2 Preliminaries

Notations. The notations of scalars, vectors, matrices, and higher-order tensors basically follow the conventions in the *Deep Learning* textbook (Goodfellow et al., 2016). For a tensor, square brackets specify its slices or entries. For instance, for tensor $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$, $\boldsymbol{X}_{i::} = \mathbf{X}[i, :, :]$ represents a $J \times K$ matrix slice, $\boldsymbol{x}_{ij:} = \mathbf{X}[i, j, :]$ denotes a K-dimensional vector slice, and $x_{ijk} = \mathbf{X}[i, j, k]$ is a scalar element. The identity matrix of size $I \times I$ is written as $\boldsymbol{I}_I$. We use $\otimes$ for the Kronecker product and $\text{tr}(\cdot)$ for the matrix trace.

For indices of tensorization, we follow the little-endian convention (Cichocki et al., 2016). When tensorizing a vector $\boldsymbol{x} \in \mathbb{R}^I$ with $I = \prod_{d=1}^D I_d$, if its tensorized form is $\mathbf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$. The correspondence between the vector and tensor indices is defined as $\boldsymbol{x}[\overline{i_1, \ldots, i_D}] = \mathbf{X}[i_1, \ldots, i_D]$, where the flattened index is $\overline{i_1, \ldots, i_D} = i_1 + (i_2 - 1)I_1 + \cdots + (i_D - 1)\prod_{d=1}^{D-1} I_d$.

Low-rank adaptation. Low-rank adaptation (LoRA) (Hu et al., 2022) is a simple but effective PEFT technique that introduces low-rank updates to pre-trained weight matrices, thereby significantly reducing the number of trainable parameters. For a linear layer $\boldsymbol{y} = \boldsymbol{W}_0\boldsymbol{x}$, where the weight matrix has a shape of $I \times J$. LoRA aims to learn an additive adaptation $\boldsymbol{\Delta}$ of the same dimensions, formulated as:

$$\boldsymbol{y}' = (\boldsymbol{W}_0 + \boldsymbol{\Delta})\boldsymbol{x}, \quad s.t. \, \boldsymbol{\Delta} = \boldsymbol{B}\boldsymbol{A}, \tag{1}$$

where the low-rank matrices $\boldsymbol{B} \in \mathbb{R}^{I \times R}$ and $\boldsymbol{A} \in \mathbb{R}^{R \times J}$ only introduce a total of $2IR$ trainable parameters, which is significantly smaller than the original IJ when $R \ll I, J$. However, recent studies (Borse et al., 2024; Nikdan et al., 2024; Zheng et al., 2025) have pointed out a critical limitation: the ideal fine-tuning weights may not inherently possess a low-rank structure.

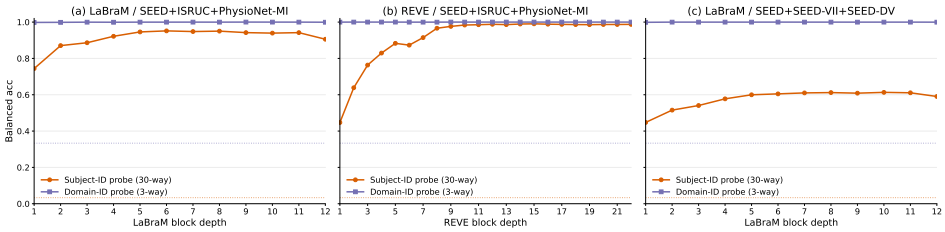


Figure 2: Layer-wise probing of domain and subject identity in frozen EEG foundation models.

Orthogonal Fine-tuning. Orthogonal Fine-tuning (OFT) (Qiu et al., 2023) applies a full-rank orthogonal transformation $\boldsymbol{T} \in \mathbb{R}^{I\times I}$ on the weight matrix $\boldsymbol{W}_0 \in \mathbb{R}^{I\times J}$ instead of adding a low-rank matrix $\boldsymbol{\Delta}$, formulated as:

$$\boldsymbol{y}' = (\boldsymbol{T}\boldsymbol{W}_0)\boldsymbol{x}. \tag{2}$$

To reduce the trainable parameter, OFT represent $\boldsymbol{T}$ with a block-diagonal matrix with r blocks, while keeping the rest part zero. However, recent studies (Liu et al., 2024a; Yuan et al., 2024) have pointed out a critical limitation: the OFT's transformation weight is extremely sparse for small parameter budgets due to the block-diagonal parameterization.

2.3 Our Motivation

Q1: Why EEG foundation models need hierachical adaptation?

Most existing PEFT methods apply a uniform adaptation strategy across the layers of (Hu et al., 2022). and LoRA is adopted to the EFM by treating different layers with the same adaptation mechanism (Lee et al., 2025; Dai et al., 2026). However, it may be suboptimal for EEG foundation models, whose representations exhibit a hierarchical structure across layers.

To investigate this hypothesis, we conduct layer-wise probing on several frozen EFMs. For each dataset, we randomly sample 10 subjects and train linear classifiers on layer-wise representations to predict domain and subject identities. As shown in Figure 2 (a)–(b), domain identity is reliably decoded from the first layer, while subject identity becomes increasingly discriminative in deeper layers (5-th layer for LaBraM and 10-th layer for REVE). This pattern remains consistent across three SEED-series datasets collected with the same EEG device (Figure 2 (c)), indicating that the observed hierarchy is not solely due to device differences.

Unlike natural language or image modalities, EEG representations are strongly shaped by the acquisition process. Shallow layers encode more domain-related variation, which is shared across subjects within the same domain, while deeper layers increasingly capture subject-specific variation (Yi et al., 2023). Therefore, shallow-layer adaptation can be shared within a domain, whereas deeper layers require more personalized adaptation. In summary:

A1: Shallow layers encode domain-level variation (e.g., montage, session and task) that is shared across subjects and can be corrected by a domain-shared adapter, while deeper layers increasingly capture subject-specific characteristics. We thus adopt a hierarchical framework with domain-shared adaptation in shallow layers and reusable subject adapters in deeper layers.

Q2: Why Domain Adapter needs full-rank Transformation $\boldsymbol{T}$ rather than LoRA $\boldsymbol{\Delta}$?

Most PEFT methods for EFMs rely on low-rank updates, motivated by the observation that the fine-tuning update $\Delta\boldsymbol{W}$ for a single downstream task is often low-rank. However, our shallow-layer *domain adapter* is **shared across all subjects within a domain** and must therefore capture the collective variation of an increasing subject population. LoRA with a fixed, small rank r cannot satisfy: once the intrinsic rank of the desired weight shift exceeds r, the adapter becomes capacity-bounded and can no longer reach the optimal weight subspace that full fine-tuning would find, no matter how it is optimized.

To verify this concern empirically, we fully fine-tune the shallow blocks of EFMs, while progressively increasing the number of subjects included in fine-tuning, and track the *effective rank* (eRank)

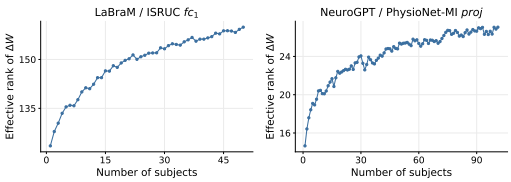


Figure 3: Effective rank of ΔW of block 0. **Left:** LaBraM on ISRUC dataset, W_{fc1}. **Right:** NeuroGPT on PhysioNet-MI, W_{proj}.

Table 1: Average effective rank of ΔW for LaBraM's shallow blocks, averaged across three datasets: ISRUC, SEED, and PhysioNet-MI.

Block	W_Q	W_K	W_V	W_{proj}	W_{fc1}	W_{fc2}
Block 0	106.47	113.01	88.51	93.64	131.60	126.61
Block 1	107.45	111.01	100.38	100.01	137.60	125.93
Block 2	102.43	109.16	100.07	96.45	133.50	124.50
Block 3	95.96	105.97	96.78	95.47	125.56	123.87

(Roy & Vetterli, 2007) of $\Delta \boldsymbol{W} = \boldsymbol{W}_{\text{ft}} - \boldsymbol{W}_{\text{pre}}$ for every weight matrix in the shallow blocks . As shown in Fig. 3, the eRank consistently increases with the number of subjects: for LaBraM, the eRank grows from ∼125 to ∼160, for NeuroGPT, the eRank grows from ∼14 to ∼27 (Note the embedding dimension of NeuroGPT is 40). Moreover, Table 1 shows that the **eRank remains high** even when averaged across the subject-number sweep and three datasets. These results indicate that domain-shared adaptation requires substantially higher-rank updates than those typically assumed by fixed-rank LoRA. In summary:

A2: The weight change induced by full fine-tuning is already high-rank on average, and the effective rank required by a domain adapter further grows with the number of subjects sharing it. A fixed low-rank LoRA update therefore becomes increasingly capacity-limited as the subject population grows. We thus adopt a full-rank transformation $\boldsymbol{T}$ as the domain adapter.

Q3: Why Transformation $\boldsymbol{T}$ needs to be factorized by Tensor Ring Decomposition?

Parameterizing $\boldsymbol{T}$ as a full $I \times I$ matrix requires $\mathcal{O}(I^2)$ parameters, motivating a compact factorization. OFT (Qiu et al., 2023) provides a parameter-efficient orthogonal transformation, but its block-diagonal structure limits interactions across neuron dimensions. BOFT (Liu et al., 2024a) improves expressiveness through dense butterfly compositions, while Tao et al. further shows that dense Tensor Ring (TR) factorizations can provide stronger expressiveness han block-sparse OFT-style constructions under comparable parameter budgets.

However, both OFT and BOFT constrain $\boldsymbol{T}$ to be orthogonal, restricting adaptation to pure rotations. Such a constraint is reasonable for Euclidean feature spaces where preserving geometric relationships is desirable. However, this assumption does not transfer to EEG. A bunch of studies proposed that EEG signals are intrinsically non-Euclidean, and processed EEG data with Riemannian (Hu et al., 2026), hyperbolic (Li et al., 2026), and governed by a probabilistic sparsity prior (Wang et al., 2023b). Therefore, an EEG foundation model pretrained across heterogeneous subjects and domains continually *stretches and warps* its feature space rather than merely rotating it. Adapting such a model to a new domain via a pure rotation cannot undo this warping. In summary:

A3: $\boldsymbol{T}$ is supposed to be simultaneously parameter-efficient, dense, and non-orthogonal, so it can warp and rotate rather than only rotate the pretrained feature space. Tensor Ring Decomposition (TRD) (Zhao et al., 2016; Tao et al., 2025) satisfies all three: it reconstructs a full and dense matrix from a ring of small core tensors at a fraction of the parameter cost, with no orthogonality constraint on the result.

3 Methodology

3.1 Overview of HiERA

As shown in Figure 4 (a), HiERA adapts the shallow blocks L_{dom} with domain-shared transform $\boldsymbol{T}_d$, parameterized by TRD and refined incrementally as more subjects of $\mathcal{D}_d$ arrive (Sec 3.2). The deep blocks L_{sub} are adapted with a repository of lightweight adapters shared across all domains: a new subject first reuses existing adapters, and a new one is added only when reuse is insufficient (Sec. 3.3). A per-domain replay buffer mitigates forgetting in the domain-shared components (Sec. 3.4). The split point follows the observation in Sec. 2.3. Algorithm is written in Appendix C.1.

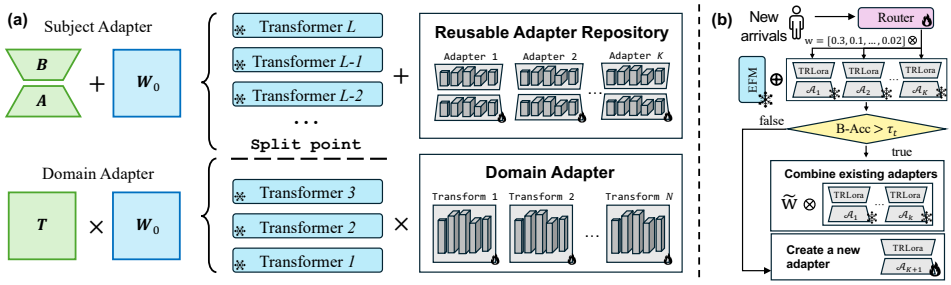


Figure 4: **Overview of HiERA**. **(a)** Shallow blocks share a domain-level transform $\boldsymbol{T}$; deep blocks draw lightweight subject adapters from a growable **Reusable Adapter Repository**. **(b)** Each new subject is routed onto existing adapters via combination weights w; if B-Acc exceeds τ_t the combination is reused, otherwise a new adapter is trained and added to the repository.

3.2 Domain-Aware Adaptation with Tensor Ring Decomposition

For each domain $\mathcal{D}_d$, HiERA maintains a single trainable transform $\boldsymbol{T}_d$ that warps the frozen weight W_0 of each shallow block in L_{dom} across all subjects, $W_0' = \boldsymbol{T}_d W_0$. $\boldsymbol{T}_d$ is shared by every subject in $\mathcal{D}_d$: when a new subject S_i^d arrives, we do not allocate a new transform but instead take a small number of gradient steps on the existing T_d using the subject's data, so its parameter count stays fixed regardless of how many subjects have been absorbed into the domain. This turns the shallow blocks into a slowly-drifting, domain-level representation that is refined incrementally as more subjects of the same domain are seen. As explained before, $\boldsymbol{T}$ is supposed to be simultaneously parameter-efficient, dense, and non-orthogonal. To meet all requirements, TRD is adopt to decompose $\boldsymbol{T}$ (Zhao et al., 2016). Consider a matrix $\boldsymbol{T} \in \mathbb{R}^{I \times I}$ with $I = \prod_{d=1}^{D} I_d$, it can be reshaped (tensorized) into a higher-order array. TRD represents this matrix through the contraction of D fourth-order tensors $\mathbf{A}^d \in \mathbb{R}^{I_d \times J_d \times R_d \times R_{d+1}}, \forall d = 1, \ldots, D$, referred to as core tensors. The TR rank is given by the sequence $[R_1, \ldots, R_{D+1}]$, subject to the cyclic constraint $R_{D+1} = R_1$. If any R_d equals 1, TRD degenerates into the tensor-train decomposition (TTD, Oseledets, 2011) format. To keep the notation simple, we set $R = R_1 = \cdots = R_{D+1}$ in the remainder of this work. Under TRD, every entry of the matrix is given by:

$$\boldsymbol{T}[\overline{i_1 \cdots i_D}, \overline{j_1 \cdots j_D}] = \operatorname{tr}(\mathbf{A}^1[i_1, j_1, :, :] \cdots \mathbf{A}^D[i_D, j_D, :, :]). \tag{3}$$

We write this compactly as $\boldsymbol{T} = \text{TRD}(\mathbf{A}^{1:D})$, with $\mathbf{A}^{1:D}$ standing for the set $\{\mathbf{A}^1, \ldots, \mathbf{A}^D\}$. TRD offers a very compact way to represent dense, full-rank matrices: when the cores $\mathbf{A}^{1:D}$ are dense and full-rank, the resulting $\text{TRD}(\mathbf{A}^{1:D})$ is dense and full-rank as well. In addition, TRD needs only $\mathcal{O}(DI^{2/D}R^2)$ memory, far less than the $\mathcal{O}(I^2)$ of the full matrix when the hyper-parameters D and R are chosen appropriately.

3.3 Shared Reusable Subject Adapter Repository

To adapt the deep blocks L_{sub} without the parameter cost growing rapidly with the number of subjects, HiERA keeps a shared adapter repository $\mathcal{A} = \{\mathcal{A}_k\}_{k=1}^{K}$. Adapters are shared across **all domains**, so an adapter learned from a subject in one dataset can be reused by subjects from other datasets. The repository does not depend on how each adapter is built, so any PEFT method can be used. In this work, we use LoRA and further compress it in the tensor ring (TR) format for higher parameter efficiency. Consider the LoRA $\boldsymbol{\Delta} \in \mathbb{R}^{I \times J}$, where $I = \prod_{d=1}^{D} I_d$ and $J = \prod_{d=1}^{D} J_d$. The TR format expresses the $\boldsymbol{\Delta}$ through a ring of core tensors, but it gives an even more compact, low-rank parameterization. Similar to the TRD above, $\boldsymbol{\Delta}$ is decomposed into $2D$ third-order cores, namely $\mathbf{B}^d \in \mathbb{R}^{I_d \times R \times R}$ and $\mathbf{C}^d \in \mathbb{R}^{J_d \times R \times R}, \forall d = 1, \ldots, D$, and each of its entries is obtained through the contraction:

$$\boldsymbol{\Delta}[\overline{i_1 \cdots i_D}, \overline{j_1 \cdots j_D}] = \operatorname{tr}(\mathbf{B}^1[i_1, :, :] \cdots \mathbf{B}^D[i_D, :, :]\mathbf{C}^1[j_1, :, :] \cdots \mathbf{C}^D[j_D, :, :]). \tag{4}$$

For each new subject S_i, HiERA first tries to reuse the existing adapters instead of allocating a new one (Fig. 4(b)). A subject-specific router $\mathbf{w}_i \in \mathbb{R}^K$ combines the K adapters:

$$\mathbf{z}_i = [\mathcal{M} \oplus \sum_{k=1}^{K} w_{i,k}\mathcal{A}_k](x_i), \tag{5}$$

where $\oplus$ denotes adapting the EFM with $\boldsymbol{T}_d$ and combined adapter. Only $\mathbf{w}_i$, the projection layer and the head are trained on $\mathcal{D}_{train}^{S_i}$, while the EFM and the adapters stay frozen. If the validation B-Acc on $\mathcal{D}_{val}^{S_i}$ exceeds τ_t, no new adapter is added. The threshold τ_t is set separately for each task and EFM, and controls the trade-off between parameter efficiency and performance: a higher τ_t adds adapters more often for better accuracy, while a lower one favors reuse. The router is then gated with $\tau_{\text{gate}} = 0.05$ into sparse weights $\tilde{\mathbf{w}}_i$, and only these weights are stored for S_i. Emprically, about 73% of subjects end up using a single adapter. If the threshold is not reached, a new prototype adapter $\mathcal{A}_{K+1}$ is trained and added to the repository so that later subjects can reuse it. This keeps the repository growing much more slowly than one adapter per subject.

3.4 Replay Buffer for Continual Personalization

Subjects arrive continually and interleaved across domains, so the components shared within a domain (the classification head H_d, the domain transform $\boldsymbol{T}_d$ and the shared subject-level transform) risk overfitting if they are optimized only on the newest one. HiERA counters this with a per-domain replay buffer B_d. After subject S_i is adapted, b class-balanced raw samples from $\mathcal{D}_{\text{train}}^{S_i}$ are appended to $B_{d(S_i)}$, which is capped at capacity C and evicts its oldest entries first. When a later subject of the same domain is adapted, each training step also draws a replay batch from B_d, passes it through the same model, and averages its loss with the current loss, weighted by batch size. The replay loss updates whatever is trainable in the current stage: only the mixture gate and H_d during reuse, and additionally $\boldsymbol{T}_d$, the shared subject-level transform and the new adapter when a new adapter is created. Adapters already stored in $\mathcal{A}$ stay frozen and are never altered. Because only s samples are kept per subject, memory stays bounded by C per domain regardless of how many subjects arrive.

4 Experiment

4.1 Datasets and Experimental Setup

We evaluate our method on three publicly available EEG datasets of three representative decoding tasks: sleep staging, motor imagery, and emotion recognition. **ISRUC** is a five-class sleep staging database involveing all-night 6-channel EEG signals from 100 individuals (Khalighi et al., 2016). **SEED** is an emotion recognition database comprising 62-channel EEG recordings from 15 subjects across 3 sessions (45 sessions in total) (Zheng & Lu, 2015). It involves three emotion categories (positive, neutral, negative). **PhysioNet-MI** (Schalk et al., 2004) is a motor imagery database comprising 64-channel EEG recordings from 109 subjects, covering four motor imagery tasks (left hand, right hand, both fists, both feet). More details are listed in the Appendix D.1.

Three EFMs are adopted to test the generality of our approach. **LaBraM** (Jiang et al., 2024b) tokenizes multi-channel EEG into discrete neural codes via a vector-quantized encoder. **REVE** (Ouahidi et al., 2025) uses rotary position embeddings to jointly encode channel and time information. **NeuroGPT** (Cui et al., 2024) couples a convolutional encoder with a causal GPT-2 decoder. A linear layer is used for classification. More details are listed in the Appendix D.3.

4.2 Main Results

We compare our model with other PEFT methods: LoRA and OFT. Classical continual learning methods are also adopted including EWC (Kirkpatrick et al., 2017) and LwF (Li & Hoiem, 2017), and their backbone is EEGConformer for processing heterogeneous EEG data. The results are summarized in Table 2, from which we draw the following observations.

Foundation-model-based PEFT outperforms classical EEG continual learning. PEFT methods built on pretrained EEG foundation models consistently outperform classical continual learning methods based on EEGConformer.

Table 2: Experiment results(%) with standard deviation in subscripts(%). For #Para, we report the total parameter cost of all subject. The **best** are in bold, and the second best are underlined.

Method	#Param (M)	ISRUC		Physionet-MI		SEED	
		B-Acc	WF1	B-Acc	WF1	B-Acc	WF1
EEGNet	0.75	$47.62_{8.00}$	$57.79_{12.45}$	$25.74_{17.75}$	$25.84_{15.44}$	$45.97_{16.50}$	$44.37_{17.29}$
EWC	0.87	$49.20_{5.97}$	$57.45_{10.81}$	$26.73_{16.29}$	$26.46_{13.98}$	$47.62_{17.06}$	$45.84_{19.01}$
LwF	0.87	$48.44_{8.50}$	$59.44_{12.16}$	$26.40_{19.20}$	$26.54_{16.44}$	$48.63_{17.11}$	$43.39_{16.74}$
LaBraM - Full FT	1472.66	$63.98_{8.27}$	$74.13_{9.52}$	$25.99_{13.44}$	$17.18_{12.91}$	$50.28_{12.45}$	$47.68_{14.04}$
+ LoRA$_{r=2}$	5.07	$69.05_{7.17}$	$80.50_{6.83}$	$25.26_{11.20}$	$11.28_{9.15}$	$54.74_{15.41}$	$51.19_{17.75}$
+ LoRA$_{r=4}$	9.93	$69.97_{6.91}$	$80.99_{6.47}$	$23.74_{11.59}$	$12.31_{10.56}$	$56.50_{14.33}$	$52.97_{17.23}$
+ LoRA$_{r=8}$	19.64	$69.73_{8.56}$	$80.87_{7.81}$	$25.68_{13.40}$	$13.22_{10.83}$	$53.70_{14.80}$	$49.87_{17.57}$
+ OFT$_{r=20}$	12.36	$69.42_{7.63}$	$80.20_{7.75}$	$25.73_{12.37}$	$13.31_{9.57}$	$56.39_{14.99}$	$53.31_{17.58}$
+ OFT$_{r=10}$	24.50	$69.98_{6.88}$	$80.48_{6.20}$	$27.34_{13.70}$	$18.80_{11.56}$	$56.97_{14.68}$	$54.00_{16.98}$
+ OFT$_{r=5}$	48.79	$70.06_{6.53}$	$79.12_{6.63}$	$\mathbf{28.52}_{12.73}$	$\mathbf{21.75}_{\mathbf{12.16}}$	$57.02_{14.60}$	$53.98_{16.90}$
+ HiERA $_{r=(1,2)}$	1.23	$\mathbf{70.59}_{7.83}$	$\mathbf{81.05}_{6.32}$	$26.06_{11.23}$	$20.56_{10.17}$	$56.34_{14.32}$	$53.16_{16.22}$
+ HiERA $_{r=(1,4)}$	4.33	$70.32_{7.83}$	$80.65_{7.41}$	$26.89_{12.71}$	$21.34_{11.93}$	$\mathbf{57.21}_{14.86}$	$\mathbf{54.02}_{17.23}$
REVE - Full FT	17505.52	$71.73_{15.17}$	$79.73_{16.81}$	$33.17_{15.27}$	$29.63_{14.17}$	$65.70_{17.81}$	$63.35_{20.63}$
+ LoRA$_{r=2}$	23.35	$71.59_{6.21}$	$80.43_{4.97}$	$29.79_{12.23}$	$20.67_{13.02}$	$62.04_{14.50}$	$60.58_{15.63}$
+ LoRA$_{r=4}$	46.14	$71.10_{6.34}$	$79.08_{5.15}$	$25.95_{10.53}$	$18.04_{11.73}$	$62.92_{13.00}$	$61.78_{14.31}$
+ LoRA$_{r=8}$	91.74	$71.26_{6.25}$	$79.30_{4.70}$	$25.97_{12.27}$	$18.55_{12.55}$	$61.76_{13.37}$	$60.29_{14.37}$
+ OFT$_{r=16}$	91.74	$71.09_{10.56}$	$79.34_{12.61}$	$31.16_{13.69}$	$26.16_{13.61}$	$\mathbf{65.68}_{13.54}$	$\mathbf{64.21}_{15.04}$
+ OFT$_{r=8}$	182.93	$70.60_{9.29}$	$78.55_{9.78}$	$30.87_{13.18}$	$25.53_{12.88}$	$63.97_{16.92}$	$60.61_{19.55}$
+ OFT$_{r=4}$	365.32	$68.85_{19.46}$	$74.96_{22.72}$	$28.44_{11.43}$	$21.20_{12.42}$	$58.85_{20.12}$	$53.59_{25.25}$
+ HiERA $_{r=(1,2)}$	3.12	$71.82_{7.34}$	$79.58_{4.73}$	$30.19_{11.24}$	$25.70_{12.51}$	$62.24_{12.97}$	$61.28_{14.03}$
+ HiERA $_{r=(1,4)}$	16.35	$\mathbf{72.82}_{7.79}$	$\mathbf{81.10}_{5.39}$	$\mathbf{32.73}_{10.82}$	$\mathbf{27.22}_{10.34}$	$64.35_{14.38}$	$63.54_{13.98}$
NeuroGPT - Full FT	111.91	$46.87_{18.77}$	$48.47_{26.53}$	$30.68_{13.41}$	$23.62_{14.91}$	$46.36_{13.37}$	$38.88_{16.10}$
+ LoRA$_{r=2}$	0.49	$47.08_{8.05}$	$51.46_{10.17}$	$29.35_{11.76}$	$20.77_{11.97}$	$45.44_{12.96}$	$\mathbf{39.76}_{16.30}$
+ LoRA$_{r=8}$	0.97	$\mathbf{47.24}_{8.27}$	$51.15_{10.51}$	$30.62_{12.18}$	$23.62_{12.83}$	$46.01_{13.25}$	$39.57_{16.55}$
+ LoRA$_{r=8}$	1.94	$46.94_{8.98}$	$51.01_{11.05}$	$28.72_{13.13}$	$20.52_{13.47}$	$44.38_{13.23}$	$38.84_{16.76}$
+ OFT$_{r=10}$	0.49	$45.31_{9.28}$	$49.41_{10.81}$	$27.15_{11.22}$	$18.14_{12.71}$	$44.49_{12.91}$	$38.87_{16.14}$
+ OFT$_{r=8}$	0.61	$45.97_{8.88}$	$49.21_{10.72}$	$27.62_{12.94}$	$20.02_{13.72}$	$44.51_{14.02}$	$39.17_{17.22}$
+ OFT$_{r=4}$	1.22	$46.29_{8.68}$	$50.76_{9.86}$	$27.14_{10.97}$	$18.80_{13.87}$	$44.12_{13.47}$	$38.37_{16.92}$
+ HiERA $_{r=(1,2)}$	0.31	$46.71_{8.62}$	$50.80_{10.20}$	$30.07_{12.71}$	$19.73_{12.27}$	$46.45_{13.92}$	$38.24_{16.24}$
+ HiERA $_{r=(1,4)}$	1.12	$47.14_{4.70}$	$\mathbf{51.65}_{11.58}$	$\mathbf{31.13}_{11.40}$	$\mathbf{20.87}_{13.93}$	$\mathbf{47.64}_{14.02}$	$38.62_{17.35}$
NeuroGPT-Dec - Full FT	19856.55	$50.25_{16.69}$	$52.87_{25.02}$	$31.66_{14.43}$	$24.74_{15.10}$	$48.84_{13.85}$	$41.95_{17.24}$
+ LoRA$_{r=2}$	13.76	$50.99_{9.98}$	$72.66_{12.27}$	$25.50_{11.16}$	$14.80_{10.18}$	$50.80_{12.77}$	$45.44_{16.96}$
+ LoRA$_{r=4}$	26.19	$51.22_{9.61}$	$72.80_{10.92}$	$26.85_{9.85}$	$14.34_{10.55}$	$50.67_{12.00}$	$46.15_{15.80}$
+ LoRA$_{r=8}$	51.06	$51.26_{9.87}$	$72.38_{11.72}$	$27.16_{11.53}$	$15.78_{11.62}$	$51.00_{12.61}$	$46.20_{16.77}$
+ OFT$_{r=32}$	100.80	$51.16_{10.77}$	$72.55_{12.59}$	$28.53_{11.30}$	$18.83_{12.15}$	$50.67_{12.30}$	$46.39_{16.16}$
+ OFT$_{r=16}$	200.29	$51.32_{9.74}$	$72.91_{11.70}$	$30.88_{12.51}$	$22.42_{13.22}$	$51.35_{13.09}$	$\mathbf{46.91}_{17.05}$
+ OFT$_{r=8}$	399.26	$49.94_{10.49}$	$71.36_{12.81}$	$30.83_{13.55}$	$22.68_{14.33}$	$50.74_{13.22}$	$46.08_{16.91}$
+ HiERA $_{r=(1,2)}$	3.37	$50.94_{9.82}$	$\mathbf{72.97}_{11.80}$	$30.32_{11.45}$	$22.13_{11.71}$	$50.81_{12.53}$	$45.33_{16.85}$
+ HiERA $_{r=(1,4)}$	18.46	$\mathbf{51.83}_{9.84}$	$72.85_{11.67}$	$\mathbf{31.29}_{12.49}$	$\mathbf{23.15}_{11.65}$	$\mathbf{51.72}_{12.47}$	$46.47_{16.97}$

HiERA achieves strong overall performance with substantially fewer parameters. Across the 24 backbone–dataset–metric combinations, HiERA achieves the best result in 16 cases and the second best in 9 cases, while LoRA and OFT only outperform it in isolated settings. In terms of mean B-Acc across datasets, HiERA$_{r=(1,4)}$ achieves the best performance among PEFT methods on REVE (56.97%), NeuroGPT (41.97%), and NeuroGPT-Dec (44.95%), even slightly surpassing full fine-tuning on all three backbones. Meanwhile, HiERA requires only a small fraction of the parameters used by other PEFT methods. For example, on LaBraM, HiERA$_{r=(1,2)}$ uses only 1.23M parameters, corresponding to 24.3% of LoRA$_{r=2}$ (5.07M) and 10.0% of OFT$_{r=20}$ (12.36M), while achieving the best ISRUC performance (70.59% B-Acc, 81.05% WF1). The reduction is even more substantial relative to full fine-tuning, with HiERA using less than 0.1% of its parameters. The consistent performance across LaBraM, REVE, NeuroGPT, and NeuroGPT-Dec, together with its substantially lower parameter cost, demonstrates the effectiveness and scalability of hierarchical adaptation in continual personalization.

Increasing the number of parameters does not reliably help LoRA and OFT. Larger-rank LoRA and OFT do not consistently improve performance, indicating that their additional parameters can be redundant. In contrast, increasing the hierarchical capacity of HiERA from $r = (1, 2)$ to $r = (1, 4)$ consistently improves performance across all four backbones.

4.3 Parameter Efficiency and Continual Learning Analysis

Figure 5 analyzes parameter growth during continual personalization. As shown in Figure 5 (a), after a short warm-up period, the rolling adapter reuse rate rises to approximately 75–90%, indicating that most later arrivals can be served by reusing existing subject prototypes. This directly leads to the parameter-growth pattern in Figure 5 (b): LoRA and OFT allocate a new adapter for every subject, resulting in rapid parameter growth, whereas our method only introduces parameters when existing prototypes cannot provide a suitable reuse combination. Consequently, the marginal parameter cost of onboarding new subjects decreases as the shared prototype pool grows.

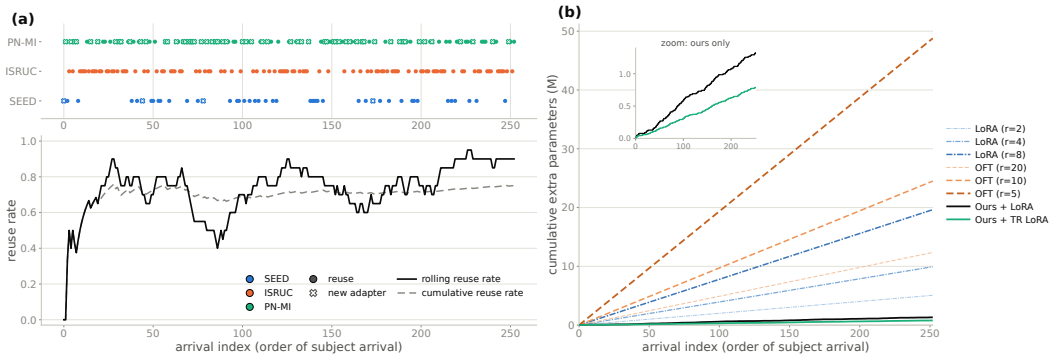


Figure 5: Reuse timeline (a) and parameter growth (b) over the subject arrival stream, condition E/D (reve-base). The reuse rate stabilizes around 75–90% after an initial warm-up; unlike the LoRA/OFT baselines, which grow rapidly, our methods grow only in small, intermittent steps (inset).

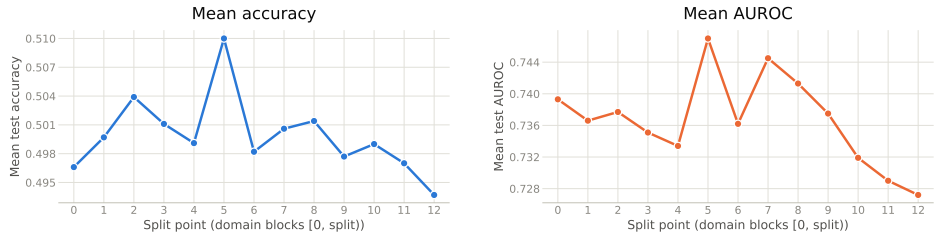


Figure 6: Ablation on the domain/subject split point for LaBraM. **Left:** mean test accuracy; **Right:** mean test AUROC across all continual arrivals from SEED, ISRUC, and PhysioNet-MI.

4.4 Ablation Study

Split Point Ablation. To further validate this layer-wise hypothesis, we ablate the location of the domain/subject boundary itself. We sweep the split point s from 0 to 12 on LaBraM: blocks $[0, s)$ receive the domain (transform-only) adapter and blocks $[s, 12)$ receive the subject (TR-LoRA) adapter, so $s = 0$ degenerates to an all-subject model and $s = 12$ to an all-domain model. As shown in Figure 6, performance peaks when the split lies in around 5-th layer of the network and drops toward both extremes. This peaks is consistent with the layer-wise probing result in Figure 2 (a) and (c), where subject identity classification reaches the highest at layer 5. Hence splitting the domain/subject adapter at the correct layer yields the best downstream performance, whereas splitting too early or too late misaligns the adapter with the information its layers actually carry.

Domain Adapter Ablation. We evaluate different domain adapters for the shallow layers in Table 3. Removing the domain adapter leads to clear performance degradation, demonstrating the importance of explicitly modeling domain-specific variations in the shallow layers. Replacing $\boldsymbol{T}$ with $\text{LoRA}_{r=1}$ even underperforming the *w/o adapter* setting, suggesting that its low-rank constraint limits its capacity to capture the domain-specific transformations required at shallow layers. OFT also provides only marginal improvements over the no-adapter baseline. Although replacing the block-wise OFT with the denser BOFT substantially improves performance, it remains consistently inferior to our TR $\boldsymbol{T}$. This indicates that the orthogonality constraint restricts the expressiveness of the transformation for EFMs. In contrast, our TR $\boldsymbol{T}$ provides a more expressive and flexible parameterization while retaining a compact parameter budget (only 4.56 KB for 5 layers), enabling more effective modeling of domain-specific variations in the shallow layers.

Subject Adapter Ablation. Removing the subject adapter substantially degrades performance, with SEED B-Acc dropping from 56.34% to 42.94%. Without adapter reuse, performance improves but requires 2.32M parameters. Replacing TR-LoRA with LoRA or OFT further improves performance in some cases, but at a much higher parameter cost. In contrast, our TR-LoRA achieves competitive performance with only 1.17M parameters, reducing the parameter cost by 24% compared with LoRA and OFT. These results highlight the parameter efficiency of TR-LoRA for maintaining subject-specific adapters in continual personalization.

Table 3: Ablation Study on Domain Adapter and Subject Adapter on LaBraM. The **best** are in bold, and the second best are underlined.

Method	#Param	ISRUC		Physionet-MI		SEED	
		B-Acc	WF1	B-Acc	WF1	B-Acc	WF1
Ablation on Domain Adapter							
w/o Domain Adapter	0	$69.31_{6.54}$	$78.87_{7.36}$	$25.23_{10.68}$	$14.79_{8.72}$	$49.12_{14.89}$	$46.16_{18.34}$
+ BOFT$_{m=2,\ b=2}$	40 KB	$\underline{70.48}_{7.79}$	$\underline{80.24}_{6.35}$	$\underline{25.63}_{11.85}$	$\underline{18.96}_{9.67}$	$\underline{54.47}_{16.02}$	$\underline{50.88}_{17.11}$
+ OFT$_{r=40}$	40 KB	$69.86_{6.92}$	$79.15_{7.04}$	$25.17_{10.43}$	$15.53_{8.44}$	$51.35_{15.14}$	$49.72_{18.21}$
+ LoRA$_{r=1}$	16 KB	$67.13_{7.45}$	$76.77_{6.29}$	$25.14_{11.71}$	$12.17_{9.81}$	$47.66_{15.90}$	$45.93_{17.03}$
+ TR $\boldsymbol{T}_{r=1}$ (Ours)	**4.56 KB**	$\mathbf{70.59}_{7.83}$	$\mathbf{81.05}_{6.32}$	$\mathbf{26.06}_{11.23}$	$\mathbf{20.56}_{10.17}$	$\mathbf{56.34}_{14.32}$	$\mathbf{53.16}_{16.22}$
Ablation on Subject Adapter							
w/o Subject Adapter	0	$68.74_{6.33}$	$77.59_{7.41}$	$25.10_{10.36}$	$12.41_{8.37}$	$42.94_{14.72}$	$43.28_{18.56}$
w/o Reuse Repository	2.32 M	$\mathbf{72.12}_{7.98}$	$\mathbf{82.41}_{6.47}$	$\underline{27.14}_{11.94}$	$\mathbf{22.84}_{9.53}$	$\mathbf{58.21}_{16.23}$	$\mathbf{55.64}_{17.28}$
+ OFT$_{r=40}$	1.54 M	$\underline{71.89}_{6.76}$	$\underline{81.73}_{7.28}$	$26.37_{10.57}$	$21.65_{8.61}$	$58.01_{14.96}$	$\underline{55.57}_{18.10}$
+ LoRA$_{r=2}$	1.86 M	$71.27_{7.21}$	$81.36_{6.61}$	$\mathbf{27.87}_{11.06}$	$\underline{22.12}_{9.03}$	$\underline{58.58}_{15.33}$	$54.05_{17.62}$
+ TR LoRA$_{r=2}$ (Ours)	**1.17 M**	$70.59_{7.83}$	$81.05_{6.32}$	$26.06_{11.23}$	$20.56_{10.17}$	$56.34_{14.32}$	$53.16_{16.22}$

Table 4: Parameter ablations of our framework. EFM is LaBraM. For (a) and (b), we report a single adapter's parameters. For (c), we report the total parameter cost of all subject.

Rank	# Para	B-Acc	WF1
1	**4.45 KB**	**51.23**	51.43
2	17.8 KB	50.49	**51.77**
4	71.3 KB	50.60	51.24
8	285 KB	49.01	51.29

(a) TR rank for Domain Adapter

Rank	# Para	B-Acc	WF1
1	**6.23 KB**	50.40	51.40
2	24.9 KB	51.23	51.43
4	99.8 KB	**51.47**	**52.03**
8	399 KB	50.88	51.01

(b) TR rank for Subject Adapter

Buffer	# Para	B-Acc	WF1
10	1.35 M	51.21	51.34
20	**1.23 M**	**51.23**	**51.43**
30	1.38 M	51.18	50.49
40	1.42 M	50.30	50.37

Buffer size for each subject. **(c)**

Parameter Ablation. Table 3 studies the key hyperparameters of HiERA. For the domain adapter, $r = 1$ already performs competitively, with only minor variations across ranks, suggesting that a low-rank transform is sufficient to capture domain-level variation. For the subject adapter, $r = 4$ achieves the best performance, while $r = 2$ offers a favorable parameter–performance trade-off. Larger ranks substantially increase the parameter cost and can even hurt performance. For the replay buffer, $b = 20$ yields the best performance. Increasing b further reduces the diversity of subjects represented in the fixed-capacity domain buffer B_d, leading to degraded continual adaptation.

4.5 Visualization of Subject Adapter Repository

To verify that the shared subject adapter repository captures reusable structure, rather than reuse arising from a lenient acceptance threshold, we visualize the stored subject adapters. We flatten each subject adapter's weight matrix into a single vector and project the resulting collection with t-SNE, coloring points by their source dataset. As shown in Fig. 7, despite the substantial difference in EEG characteristics across domains, the resulting subject adapters largely fall within one shared embedding space rather than splitting apart by dataset, and visibly overlap, particularly between SEED and PhysioNet-MI for NeuroGPT and between SEED and ISRUC for NeuroGPT-Dec. This cross-dataset overlap qualitatively supports the reusability of the shared subject adapter repository.

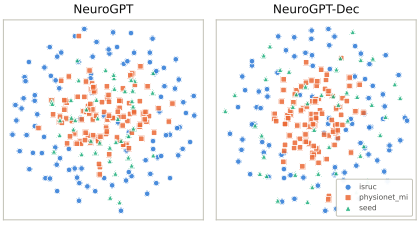


Figure 7: t-SNE projection of the subject adapter weight.

5 Conclusion

We presented HiERA, a hierarchical domain-aware reusable adaptation framework for continual personalization of EEG foundation models. Motivated by the layer-wise differences in domain- and subject-related information, HiERA employs a domain-shared Tensor Ring transformation in shallow layers and a reusable adapter repository in deeper layers. This design enables flexible domain adaptation while avoiding the rapid parameter growth. Experiments across multiple EEG datasets and foundation models demonstrate that HiERA achieves competitive personalization performance with substantially fewer parameters. By reducing the computational and storage costs of continual subject adaptation, HiERA brings EFMs closer to practical BCI deployment and provides a scalable approach for personalized real-world BCI systems.

REPRODUCIBILITY STATEMENT

We provide sufficient details to facilitate the reproducibility of our experiments. The model architecture, adaptation strategy, training procedure, and key hyperparameters are described in the main paper and Appendix. Detailed dataset information, preprocessing procedures, experimental settings, and additional implementation details are provided in the Appendix. We will release the implementation code and configuration files upon acceptance. All experiments are conducted using fixed experimental protocols, and the reported results are obtained following the evaluation procedures described in the paper.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing and debugging software code, designing and refining experimental procedures, interpreting experimental results, and editing parts of the manuscript. We also used generative AI tools to improve the readability, grammar, and organization of the paper.

All AI-assisted code and experimental procedures were reviewed, tested, and verified by the authors. The authors independently conducted the experiments, analyzed and validated the results, and made the final decisions regarding the methodology, experimental design, and interpretation of the findings. AI-generated text was also reviewed and revised by the authors to ensure its accuracy and consistency with the actual work. The authors take full responsibility for the final content of this paper, including all claims, experimental results, code, and other artifacts produced with the assistance of generative AI.

REFERENCES

Afia Anjum, Maksim E Eren, Ismael Boureima, Boian Alexandrov, and Manish Bhattarai. Tensor train low-rank approximation (tt-lora): Democratizing ai with accelerated llms. In *2024 International Conference on Machine Learning and Applications (ICMLA)*, pp. 583–590. IEEE, 2024.

Shubhankar Borse, Shreya Kadambi, Nilesh Prasad Pandey, Kartikeya Bhardwaj, Viswanath Ganapathy, Sweta Priyadarshi, Risheek Garrepalli, Rafael Esteves, Munawar Hayat, and Fatih Porikli. FouRA: Fourier low-rank adaptation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=qCJ1dq5M7N.

Yanlong Chen, Mattia Orlandi, Pierangelo Maria Rapa, Simone Benatti, Luca Benini, and Yawei Li. Physiowave: A multi-scale wavelet-transformer for physiological signal representation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=ayR2JfRYRS.

Yuqi Chen, Kan Ren, Kaitao Song, Yansen Wang, Yifan Wang, Dongsheng Li, and Lili Qiu. EEGFormer: Towards transferable and interpretable large-scale EEG foundation model. In *AAAI 2024 Spring Symposium on Clinical Foundation Models*, 2024. URL https://openreview.net/forum?id=MXRy6bYBfB.

Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo P Mandic. Tensor networks for dimensionality reduction and large-scale optimization part 1 low-rank tensor decompositions. *Foundations and Trends in Machine Learning*, 9(4-5):249–429, 2016.

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

Sicheng Dai, Kai Chen, Hongwang Xiao, Shan Yu, and Qiwei Ye. Towards unified multi-task eeg analysis with low-rank adaptation. *arXiv preprint arXiv:2604.25131*, 2026.

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

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. URL https://openreview.net/forum?id=D4hcJPkJ3y.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Chen Hu, Ziheng Chen, Rui Wang, Yefeng Zheng, and Nicu Sebe. Riemannian high-order pooling for brain foundation models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=66h1sCMm7F.

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 *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Ziyu Jia, Junyi Lin, Pu Wan, Jinxin Pi, Jingying Ma, Peiliang Gong, Xinliang Zhou, Yi Ding, and Chenyu Liu. Embrace: A collective knowledge fusion framework toward unified EEG foundation models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=BJ5rYj8O8W.

Wei-Bang Jiang, Xuan-Hao Liu, Wei-Long Zheng, and Bao-Liang Lu. Seed-vii: A multimodal dataset of six basic emotions with continuous labels for emotion recognition. *IEEE Transactions on Affective Computing*, 16(2):969–985, 2024a.

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*, 2024b. URL https://openreview.net/forum?id=QzTpTRVtrP.

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. URL https://openreview.net/forum?id=Io9yFt7XH7.

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.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Gayal Kuruppu, Neeraj Wagh, Vaclav Kremen, and Yogatheesan Varatharajah. Eeg foundation models: a critical review of current progress and future directions. *Journal of neural engineering*, 23(2):021001, 2026.

Na Lee, Konstantinos Barmpas, Yannis Panagakis, Dimitrios Adamos, Nikolaos Laskaris, and Stefanos Zafeiriou. Are large brainwave foundation models capable yet ? insights from fine-tuning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=J5SbLoq7Uv.

Shanglin Li, Chu Shiwen, Okan Koç, Yi Ding, Qibin Zhao, Motoaki Kawanabe, and Ziheng Chen. HEEGNet: Hyperbolic embeddings for EEG. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=CNDNRjpVIL.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

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. URL https://openreview.net/forum?id=ClLQ6cLkoR.

Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, Yandong Wen, Michael J. Black, Adrian Weller, and Bernhard Schölkopf. Parameter-efficient orthogonal finetuning via butterfly factorization. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=7NzgkEdGyr.

Xuan-Hao Liu, Yan-Kai Liu, Yansen Wang, Kan Ren, Hanwen Shi, Zilong Wang, Dongsheng Li, Bao liang Lu, and Wei-Long Zheng. EEG2video: Towards decoding dynamic visual perception from EEG signals. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=RfsfRn9OFd.

Jingying Ma, Feng Wu, Qika Lin, Yucheng Xing, Chenyu Liu, Ziyu Jia, and Mengling Feng. Codebrain: Bridging decoupled tokenizer and multi-scale architecture for EEG foundation model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=msJgEkjwh5.

Mahdi Nikdan, Soroush Tabesh, Elvir Crnčević, and Dan Alistarh. RoSA: Accurate parameter-efficient fine-tuning via robust adaptation. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=FYvpxyS43U.

Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

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. URL https://openreview.net/forum?id=ZeFMtRBy4Z.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=K30wTdIIYc.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pp. 606–610. IEEE, 2007.

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.

Zerui Tao, Yuhta Takida, Naoki Murata, Qibin Zhao, and Yuki Mitsufuji. Transformed low-rank adaptation via tensor decomposition and its applications to text-to-image models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16333–16344. IEEE, 2025.

Christopher Wang, Vighnesh Subramaniam, Adam Uri Yaari, Gabriel Kreiman, Boris Katz, Ignacio Cases, and Andrei Barbu. BrainBERT: Self-supervised representation learning for intracranial recordings. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=xmcYx_reUn6.

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*, 2025. URL https://openreview.net/forum?id=NPNUHgHF2w.

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.

Wenlong Wang, Feifei Qi, David Paul Wipf, Chang Cai, Tianyou Yu, Yuanqing Li, Yu Zhang, Zhuliang Yu, and Wei Wu. Sparse bayesian learning for end-to-end EEG decoding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15632–15649, 2023b.

Yifan Wang, Yijia Ma, Wen Li, and Chenyu You. Let EEG models learn EEG. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=TP8OuKKmsf.

Qinfan Xiao, Ziyun Cui, Chi Zhang, SiQi Chen, Wen Wu, Andrew Thwaites, Alexandra Woolgar, Bowen Zhou, and Chao Zhang. Brainomni: A brain foundation model for unified EEG and MEG signals. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=cjHQj0tCy6.

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. URL https://openreview.net/forum?id=vGeNaFHdET.

Chaoqi Yang, M Brandon Westover, and Jimeng Sun. BIOT: Biosignal transformer for cross-data learning in the wild. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=c2LZyTyddi.

Liuyin Yang, Qiang Sun, Ang Li, and Marc M. Van Hulle. Are EEG foundation models worth it? comparative evaluation with traditional decoders in diverse BCI tasks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5Xwm8e6vbh.

Shih yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=3d5CIRG1n2.

Ke Yi, Yansen Wang, Kan Ren, and Dongsheng Li. Learning topology-agnostic EEG representations with geometry-aware modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=hiOUySN0ub.

Shen Yuan, Haotian Liu, and Hongteng Xu. Bridging the gap between low-rank and orthogonal adaptation via householder reflection adaptation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=LzLeAscHnj.

Daoze Zhang, Zhizhang Yuan, Yang Yang, Junru Chen, Jingjing Wang, and Yafeng Li. Brant: Foundation model for intracranial neural signal. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=DDkl9vaJyE.

Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.

Jingjing Zheng, Wanglong Lu, Yiming Dong, Chaojie Ji, Yankai Cao, and Zhouchen Lin. AdaMSS: Adaptive multi-subspace approach for parameter-efficient fine-tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=8ZdWmpYxT0.

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.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=agcXjEHmyW.

A Related Work

A.1 EEG Foundation Models

Motivated by the success of self-supervised foundation models in NLP and vision, a growing body of work has developed EEG foundation models (EFMs) that pretrain large transformers on heterogeneous, large-scale EEG corpora before adapting them to downstream decoding tasks. One line of work focuses on tokenization and architecture design for heterogeneous montages and recording protocols, including LaBraM (Jiang et al., 2024b), CBraMod (Wang et al., 2025), NeurIPT (Fang et al., 2025), CSBrain (Zhou et al., 2025), REVE (Ouahidi et al., 2025), BrainOmni (Xiao et al., 2025), EEGFormer (Chen et al., 2024), CodeBrain (Ma et al., 2026), Riemannian high-order pooling (Hu et al., 2026), and a recent study revisiting self-supervised pretraining objectives tailored to EEG signal statistics (Wang et al., 2026b). Another line extends pretraining to intracranial recordings and cross-modality biosignals, such as Brant (Zhang et al., 2023), BrainBERT (Wang et al., 2023a), BIOT (Yang et al., 2023), and PhysioWave (Chen et al., 2025), while DeeperBrain (Wang et al., 2026a) and BrainPro (Ding et al., 2025) study scaling and probing behavior of pretrained brain models. More recently, multi-task and sequence-oriented EFMs such as NeuroLM (Jiang et al., 2025), MTEEG (Dai et al., 2026), ECHO (Liu et al., 2026), which casts EEG decoding as a contextual seq2seq problem, and EmBrace (Jia et al., 2026), which fuses knowledge across heterogeneous pretrained EFMs into a unified backbone, aim to unify heterogeneous decoding objectives within a single backbone.

Despite this progress, recent evidence suggests that the benefits of EFMs over traditional decoders are often marginal under standard full fine-tuning (Yang et al., 2026), a conclusion corroborated by systematic diagnostic benchmarking of existing EFMs (Xiong et al., 2026), which is also costly to store and update per subject or session. This is particularly problematic in continual personalization settings, where a deployed EFM must be repeatedly adapted as new user data arrives, making parameter-efficient fine-tuning (PEFT) methods tailored to EFMs essential for realizing their practical, scalable deployment value.

A.2 Parameter-efficient Fine-tuning

Parameter-efficient fine-tuning (PEFT) has become the standard approach for adapting large pretrained models with minimal trainable parameters. LoRA (Hu et al., 2022) reparameterizes weight updates as a low-rank product of two matrices, and subsequent work has explored alternative structural priors to further improve its expressiveness and efficiency: OFT (Qiu et al., 2023) and BOFT (Liu et al., 2024a) constrain updates to orthogonal transformations for better stability, FouRA (Borse et al., 2024) performs adaptation in the frequency domain, DoRA (yang Liu et al., 2024) decomposes weights into magnitude and direction components, and TLora (Tao et al., 2025) applies a learnable transformation to the low-rank subspace itself. These methods, however, are largely designed for generic language or vision backbones and do not account for EEG-specific structure such as subject variability or multi-task decoding heads. Within the EEG domain, LoRA has been directly applied to EFMs (Lee et al., 2025), and MTEEG (Dai et al., 2026) further adapts LoRA to multi-task EEG fine-tuning by sharing the down-projection matrix A across tasks while keeping a task-specific up-projection matrix B for each task.

Notably, MTEEG (Dai et al., 2026) is designed for multi-task learning However, in continual personalization setting, allocating each subject an up-projection matrix B is expensive due to the large number of subjects. In contrast, our method exploits layer-wise redundancy and reusable cross-subject adapter sharing to achieve substantially higher parameter efficiency without sacrificing decoding accuracy.

A.3 Tensor Decomposition

Tensor decomposition techniques such as tensor-train (TT) decomposition (Oseledets, 2011) and its extension, tensor-ring (TR) decomposition (Zhao et al., 2016), factorize high-dimensional weight tensors into chains of low-rank cores, offering a compact and structured alternative to standard low-rank matrix factorization. These techniques have recently been incorporated into LoRA to further compress adapter parameters: TT-LoRA (Anjum et al., 2024) reparameterizes the LoRA update as a tensor-train decomposition, and TLora (Tao et al., 2025) learns a transformed low-rank subspace

inspired by similar tensor factorization principles. Building on this line of work, we adopt tensor decomposition to jointly factorize adapters across layers and subjects, enabling efficient parameter sharing beyond what standard LoRA variants support.

B Motivation Related Experiments

B.1 Layer-wise Subject- and Domain-Identity Probing

We detail the protocol behind the layer-wise identity-probing results in Figure 2. All experiments use *frozen*, pretrained EEG foundation-model backbones: at no point is any backbone parameter updated by gradient descent. Only lightweight linear probes are fit on top of the frozen features. Two frozen backbones are probed:

- **LaBraM-base**: patch embedding size 200, embedding dimension $d = 200$, 12 transformer blocks, 10 attention heads, MLP ratio 4. Loaded from its official pretrained checkpoint; all parameters set to `requires_grad=False` and the model kept in `eval()` mode for the entire experiment.
- **REVE-base**: embedding dimension $d = 512$, 22 transformer blocks, 8 attention heads (head dimension 64), GEGLU MLP with expansion ratio 2.66, patch size 200 samples with a 20-sample overlap. Loaded from its official pretrained checkpoint (via HuggingFace-style `safetensors` weights) and likewise fully frozen.

At every block depth, each probe is a `StandardScaler` (fit on the probe training features only) followed by multinomial `LogisticRegression` (`max_iter=2000`, `class_weight='balanced'`, fixed random state). Probes are scored on the held-out val+test features by *balanced accuracy*, chosen for robustness to the residual class imbalance across the pooled domains/ subjects. The subject-ID and domain-ID probes at a given depth are trained independently but on the identical set of pooled train/eval features for that depth.

Two pooling settings are used, corresponding to the three panels of Figure 2: **Setting 1** (a, b): SEED (Zheng & Lu, 2015) + ISRUC (Khalighi et al., 2016) + PhysioNet-MI (Schalk et al., 2004), probed with both LaBraM and REVE. **Setting 2** (c): SEED + SEED-VII (Jiang et al., 2024a) + SEED-DV Liu et al. (2024b), probed with LaBraM only. All three share the same 62-channel montage.

In each setting, 10 subjects are sampled *per domain* (30 subjects pooled in total), and at every block depth two independent linear probes are fit on the *same* pooled features: (1) a 30-way **subject-ID probe** (chance $= 1/30 \approx 0.033$); (2) a 3-way **domain-ID probe** (chance $= 1/3 \approx 0.333$), predicting which of the three domains a window came from. No downstream task label (emotion category, sleep stage, motor-imagery class, SEED-DV block id) is used anywhere in this experiment; these datasets have incompatible label spaces and are pooled purely for identity information.

For every sampled subject, that subject's own official `train` split is used to fit both probes, and the `val+test` splits (concatenated) are held out purely for evaluation; probe training windows and probe evaluation windows never overlap. Table 5 summarizes the per-domain convention.

Table 5: Per-domain window/patch convention and train/val/test split rule. All splits are the dataset's own official split, reused unmodified from the main training pipeline.

Domain	#Subj. (pool)	#Ch.	Window/patch	Split rule
SEED	15 (10)	62	4 s / 1 s	sequential 9/3/3 (train/val/test)
SEED-VII	20 (10)	62	4 s / 1 s	positional 2/1/1 within block
SEED-DV	20 (10)	62	4 s / 1 s	sequential 5/1/1 (train/val/test)
ISRUC	99 (10)	6	10 s / 1 s	random 70/15/15, seeded by subject id
PhysioNet-MI	109 (10)	64	4 s / 1 s	random 70/15/15, seeded by subject id

Notes on Table 5: SEED and SEED-DV use a fixed, non-random split along trial/block index (not shuffled), since both corpora already alternate/interleave conditions in a way that keeps each split class-balanced. SEED-VII splits within each 4-trial emotion block by position (positions 0–1 $\rightarrow$ train, 2 $\rightarrow$ val, 3 $\rightarrow$ test), giving 40/20/20 trials per subject overall. ISRUC and PhysioNet-MI

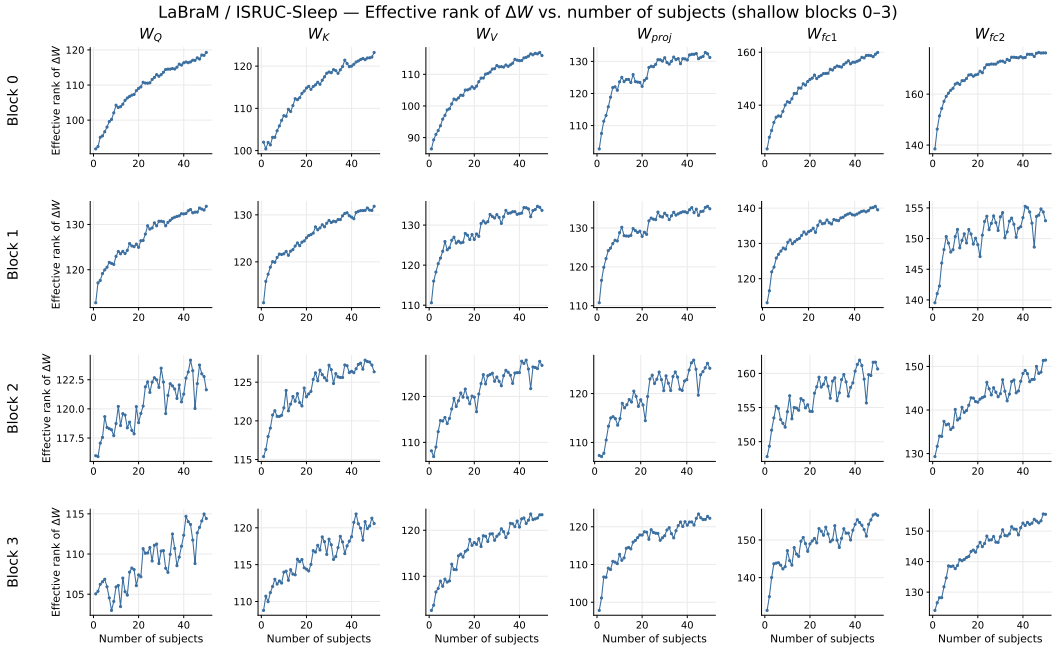


Figure 8: Effective rank of ΔW as a function of the number of training subjects, for LaBraM's trainable shallow blocks (0–3) on ISRUC-Sleep. Each panel corresponds to one weight matrix within one block.

instead use a random 70/15/15 split, seeded deterministically by the integer subject id; chronological splitting was deliberately rejected for ISRUC because sleep-stage proportions drift systematically across the night. ISRUC's raw 30 s AASM epochs are additionally cut into 3 non-overlapping 10 s sub-windows each (all 3 sub-windows of one epoch are kept in the *same* split, to avoid leaking within-epoch information across splits) — this 10 s choice, rather than the native 30 s epoch, is forced by LaBraM's positional-embedding table capping patches-per-channel at 16. ISRUC subject 8 is excluded (missing F3/F4 leads), leaving 99 usable subjects out of 100.

B.2 Effective Rank of the Weight Update

To characterize how the fine-tuning update $\Delta W = W_{\text{finetuned}} - W_{\text{pretrained}}$ evolves with more training data, we measure its *effective rank* (Roy & Vetterli, 2007). This gives a continuous, differentiable proxy for matrix rank: a ΔW dominated by a single singular value has eRank ≈ 1, while one with a flat spectrum approaches $\min(\text{rows}, \text{cols})$.

We fine-tune LaBraM-base on ISRUC-Sleep with only the first four transformer blocks (blocks 0–3) and the classification head trainable, freezing the rest of the backbone. We sweep the number of pooled training subjects from 1 to 50 (fixed random order, seed 42); at each step the model is reset to the pretrained weights and re-trained on the pooled data of the first k subjects. After each step, we extract ΔW for every weight matrix in blocks 0–3 ($W_Q, W_K, W_V, W_{\text{proj}}, W_{fc1}, W_{fc2}$, splitting $Q/K/V$ out of the fused `qkv` projection) and compute its effective rank. Figure 8 shows the resulting trend for each block/matrix pair. It can be observed the shallow layers' ΔW has bigger and bigger eRank when subject number increases.

B.2.1 Calculation of Effective Rank

The effective rank (eRank) (Roy & Vetterli, 2007) of a given matrix $\boldsymbol{W} \in \mathbb{R}^{m \times n}$ is computed via its singular value distribution, utilizing the concept of Shannon entropy to measure its true information-bearing dimension. First, we apply Singular Value Decomposition (SVD) to obtain $\boldsymbol{W} = U\Sigma V^T$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ denote the ordered non-zero singular values, and $r \leq \min(m, n)$ is the classical algebraic rank. To quantify the energy distribution, these singular values are normalized

into a discrete probability vector $p \in \mathbb{R}^r$, where each element p_i is defined as:

$$p_i = \frac{\sigma_i}{\sum_{j=1}^{r} \sigma_j}, \quad \text{for } i = 1, 2, \ldots, r \tag{6}$$

Next, the singular value entropy $H(W)$, which characterizes the uncertainty or concentration of the matrix's energy allocation, is calculated by:

$$H(\boldsymbol{W}) = -\sum_{i=1}^{r} p_i \ln(p_i) \tag{7}$$

Finally, the effective rank eRank(W) is evaluated as the exponential of this entropy:

$$\text{eRank}(\boldsymbol{W}) = \exp\left(H(\boldsymbol{W})\right) = \exp\left(-\sum_{i=1}^{r} p_i \ln(p_i)\right) \tag{8}$$

Unlike the classical rank, which is highly sensitive to noise and can abruptly jump to $\min(m, n)$ under minor numerical perturbations, eRank$(\boldsymbol{W})$ provides a continuous and stable bound satisfying $1 \leq \text{eRank}(\boldsymbol{W}) \leq r$, accurately capturing the true underlying structural dimensionality of $\boldsymbol{W}$. Therefore, we employ the eRank instead of the classical rank to analyze the weight matrices of models processing EEG signals, which are notoriously non-stationary and characterized by a low signal-to-noise ratio (SNR).

C Details of HiERA

C.1 Algorithm of HiERA

Algorithm 1 summarizes the full procedure of HiERA. Each domain $\mathcal{D}_d$ keeps a transform $\boldsymbol{T}_d$, initialized to the identity, together with a head H_d and a replay buffer B_d; the adapter repository $\mathcal{A}$ is shared across all domains and starts empty. For each incoming subject S_i, HiERA first tries to reuse the repository (Stage 1): only the router $\mathbf{w}_i$ and H_d are trained, while $\boldsymbol{T}_d$ and $\mathcal{A}$ stay frozen. The router is then sparsified with τ_{gate}. If the resulting validation B-Acc reaches τ_t, only the sparse router $\tilde{\mathbf{w}}_i$ is stored for S_i. Otherwise, HiERA expands the repository (Stage 2): it trains a new adapter $\mathcal{A}_{K+1}$ together with H_d and a copy $\hat{\boldsymbol{T}}_d$ warm-started from $\boldsymbol{T}_d$. $\hat{\boldsymbol{T}}_d$ is then merged into $\boldsymbol{T}_d$ by a running mean over the TR cores, so $\boldsymbol{T}_d$ aggregates all subjects of the domain that triggered an expansion. Both stages are trained on the current subject's data jointly with replay samples from B_d. Finally, b class-balanced samples of S_i are added to B_d, and the oldest entries are evicted once $|B_d|$ exceeds C.

D Experiment

D.1 Datasets and Experimental Setup

We evaluate our method on three publicly available EEG datasets spanning three representative decoding tasks – sleep staging, motor imagery, and emotion recognition – to test the generality of continual, subject-wise parameter-efficient adaptation across heterogeneous channel counts, sampling rates, and label spaces. All three datasets are organized under the same subject-by-subject continual-adaptation protocol: the pretrained EEG foundation model backbone arrives at one subject (or session) at a time and is adapted with a lightweight adapter on that subject's own train/val/test split before moving on to the next, randomly ordered subject.

ISRUC-Sleep. We use Cohort I of the ISRUC-Sleep dataset (Khalighi et al., 2016), which contains overnight polysomnography (PSG) recordings from 100 subjects. One subject is excluded because two of the six standard EEG leads are missing from its recording, leaving 99 subjects. For each subject, we retain the 6 scalp EEG channels referenced to the contralateral mastoid (F3, C3, O1, F4, C4, O2), sampled at 200 Hz. Sleep stages are scored by two annotators over consecutive 30-second AASM epochs into 5 classes (Wake, N1, N2, N3, REM); we keep an epoch only when both annotators agree (mean inter-annotator agreement 82.1%, range 61.9%–97.7% across subjects). Each kept 30-second epoch is further split into 3 non-overlapping 10-second sub-windows to satisfy the

Algorithm 1 HiERA: Continual Multi-Domain Personalization

Require: Frozen EFM $\mathcal{M}$; subject stream $\{S_i\}_{i=1}^{N}$ with domains $d(S_i)$;
thresholds τ_t, τ_{gate}; buffer size b per subject; capacity C

$\boldsymbol{T}_d \leftarrow \boldsymbol{I}$, $n_d \leftarrow 0$, $B_d \leftarrow \emptyset$ for every domain d
$\mathcal{A} \leftarrow \emptyset$, $K \leftarrow 0$
for $i = 1$ **to** N **do**
  $d \leftarrow d(S_i)$; initialize H_d if S_i is the first subject of $\mathcal{D}_d$
  acc $\leftarrow -\infty$
  if $K > 0$ **then** ▷ Stage 1: reuse
    Train $\mathbf{w}_i$, H_d on $\mathcal{D}_{\text{train}}^{S_i} \cup B_d$; freeze $\boldsymbol{T}_d$ and $\mathcal{A}$
    $\tilde{\mathbf{w}}_i \leftarrow \textsc{Sparsify}(\mathbf{w}_i, \tau_{\text{gate}})$
    acc $\leftarrow$ val B-Acc of $\mathcal{M}$ with $\boldsymbol{T}_d$ and $\sum_k \tilde{w}_{i,k}\mathcal{A}_k$ on $\mathcal{D}_{\text{val}}^{S_i}$
    if acc $< \tau_t$ **then** restore H_d
    end if
  end if
  if acc $< \tau_t$ **then** ▷ Stage 2: expand
    Train $\hat{\boldsymbol{T}}_d$, $\mathcal{A}_{K+1}$, H_d on $\mathcal{D}_{\text{train}}^{S_i} \cup B_d$,
    with $\hat{\boldsymbol{T}}_d$ warm-started from $\boldsymbol{T}_d$
    $n_d \leftarrow n_d + 1$; $\boldsymbol{T}_d \leftarrow \boldsymbol{T}_d + \frac{1}{n_d}(\hat{\boldsymbol{T}}_d - \boldsymbol{T}_d)$ ▷ running mean
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{A}_{K+1}\}$; $\tilde{\mathbf{w}}_i \leftarrow \mathbf{e}_{K+1}$; $K \leftarrow K + 1$
  end if
  Store $\tilde{\mathbf{w}}_i$ for S_i
  $B_d \leftarrow B_d \cup \{b$ class-balanced samples of $S_i\}$
  Evict the oldest entries of B_d while $|B_d| > C$
end for
return $\{\boldsymbol{T}_d, H_d\}$, $\mathcal{A}$, $\{\tilde{\mathbf{w}}_i\}_{i=1}^{N}$

backbone's maximum patch-length constraint, with all sub-windows of the same epoch always kept within the same split. Kept epochs are randomly shuffled per subject (seeded for reproducibility) and split 70%/15%/15% into train/val/test.

PhysioNet-MI. We use the PhysioNet EEG Motor Movement/Imagery Database, which contains 64-channel recordings from 109 subjects (Schalk et al., 2004). All signals are resampled to a common 200 Hz (from a native 160 Hz for 105 subjects and 128 Hz for the remaining four) via exact-ratio polyphase resampling. We use only the motor-*imagery* runs and define a 4-class task – left hand, right hand, both fists, both feet – from the cued T1/T2 imagery events, yielding on average about 90 trials per subject. Each trial is a fixed 4-second window starting at cue onset. Trials are randomly shuffled per subject (seeded) and split 70%/15%/15% into train/val/test, consistent with the protocol used for ISRUC-Sleep.

SEED. We use the SEED emotion dataset (Zheng & Lu, 2015), which records 62-channel EEG at 200 Hz from 15 subjects, each watching emotional video clips across 3 recording sessions (45 session recordings in total). We treat each of the 45 sessions as an independent continual-adaptation task, rather than merging a subject's three sessions together. Each session contains 15 trials with a fixed emotion-label order (positive/neutral/negative), yielding a 3-class classification task. Trials within a session are split sequentially 9/3/3 into train/val/test, which is already class-balanced due to the fixed label ordering. Each trial is further cut into non-overlapping 4-second windows.

D.2 EEG Foundation Models

EEG foundation models (EFMs) are pretrained with self-supervised objectives on large, heterogeneous unlabeled EEG corpora to learn general spatio-temporal representations that transfer to diverse downstream tasks. They differ mainly in how they tokenize EEG, how they encode electrode and time positions, and whether their attention is bidirectional or causal. We adopt three EFMs that differ along all three axes (Table 6).

LaBraM (Jiang et al., 2024b) splits each channel into 1 s patches and adds learnable temporal and spatial embeddings, which lets it handle varying montages. A vector-quantized neural tokenizer, trained to reconstruct each patch's Fourier amplitude and phase, maps patches to discrete codes. A bidirectional transformer is then pretrained to predict the codes of masked patches, using about 2,500 hours of EEG. We use LaBraM-Base (5.8M parameters), the split point is 5. LaBraM's embedding dimension is 200, we use a constant `[2,4,5,5]` for TRD.

REVE (Ouahidi et al., 2025) uses a 4D positional encoding that combines 3D electrode coordinates with time, so it generalizes to unseen montages and sequence lengths. It is pretrained as a masked autoencoder with spatio-temporal block masking on over 60,000 hours of EEG from about 25,000 subjects. The split point is 9. REVE's embedding dimension is 256, we use a constant `[4,4,4,4]` for TRD.

NeuroGPT (Cui et al., 2024) pairs a convolutional-attention encoder, which embeds consecutive EEG chunks, with a causal GPT-2 decoder that is pretrained autoregressively on the TUH EEG corpus to predict each chunk's embedding from the preceding ones. Its unidirectional attention lets us test our method beyond bidirectional EFMs. The split point is 3. NeuroGPT's encoder has embedding dimension of 40, we use a constant `[2,4,5]` for TRD. NeuroGPT's decoder has embedding dimension of 1024, we use a constant `[4,4,8,8]` for TRD.

For all models, we load the released pretrained weights, freeze the backbone for the parameter-efficient methods, and train only the adapters and a classification head.

Table 6: Overview of the adopted EFMs.

	LaBraM	**REVE**	**NeuroGPT**
Attention	Bidirectional	Bidirectional	Causal
Position enc.	Learned temporal + channel	4D (electrode coord. + time)	Learned temporal
Pretraining	VQ + masked code prediction	Masked autoencoding	Autoregressive
Data	∼2,500 h	>60,000 h	TUH EEG

D.3 Implementation Details

For all three datasets, subjects (or, for SEED, sessions) arrive one at a time in random order; the pretrained EEG foundation model backbone is kept frozen and adapted on the current subject's data with our lightweight adapter for 15 epochs before moving to the next subject. We use the AdamW optimizer with learning rate 5×10^{-4}, weight decay 5×10^{-2}, and batch size 64. The adapter uses rank $r = 2$ with scaling $\alpha = 4$ (i.e., $\alpha = 2r$), and its tensor-ring transform uses rank 1 with 3 factors. For replay buffer, the subject buffer size $b = 20$, with capacity of a domain is set to $C = 1000$. Raw EEG signals are rescaled by a factor of $1/100$ and reshaped into the (channel, patch, sample) convention shared across all backbones and datasets, with each patch spanning one second at 200 Hz. All experiments are implemented in Python 3.13 using PyTorch 2.6 (CUDA 12.4) and executed on NVIDIA RTX A5000 GPUs.