BRAINNEST: EVIDENCE-GATED CONSOLIDATION FOR FEDERATED CONTINUAL EEG LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large EEG Models (LEMs) learn transferable representations from diverse datasets, yet their knowledge must keep pace with new tasks, recording protocols, and subject populations. These datasets arrive over time and remain distributed across institutions, calling for a model that can learn collaboratively while retaining earlier capabilities. We present the first systematic study of federated continual adaptation of pretrained LEMs across evolving EEG datasets and task families. Our framework, **BrainNEST**, organizes adaptation around evidence-gated knowledge consolidation. Clients encode multiscale channel and frequency-band structure into compact sketches. The server establishes partial cross-client correspondences and uses this evidence to control both the blockwise contribution of local updates and their allocation to shared or routed state. Routed parameter memory retains updates for subsequent reuse, retrieving offsets by structural and acquisition compatibility to initialize future clients. Experiments on 14 datasets spanning seven BCI task families and two pretrained backbones demonstrate improved adaptation, retention, and transfer to unseen datasets. BrainNEST provides a practical approach to maintaining LEMs as distributed EEG resources continue to grow.

1 INTRODUCTION

Electroencephalography (EEG) supports a broad range of brain–computer interfaces, from motor imagery and emotion recognition to sleep staging and clinical assessment (Wolpaw, 2004; Lotte et al., 2007; 2018). Large EEG Models (LEMs) learn transferable representations through pretraining on extensive EEG recordings (Banville et al., 2021; Kostas et al., 2021; Jiang et al., 2024b; Wang et al., 2024b). Recent models increasingly draw on multiple datasets and tasks, using instruction tuning, contextual learning, and unified pretraining to broaden their capabilities (Jiang et al., 2024a; Liu et al., 2025; Chen et al., 2025c). This progress makes data diversity central to building more capable EEG models, as illustrated in Fig. 1(A).

The next challenge is to keep these models learning as EEG resources grow. New datasets bring additional BCI tasks, electrode montages, recording protocols, and subject populations (Lu et al., 2023; Roy et al., 2019; Chen et al., 2025b; Zhang et al., 2023a). A fixed pretrained checkpoint cannot incorporate knowledge from datasets collected after its release. Independently fine-tuning a copy for each new dataset fragments this knowledge across separate models. Repeated joint training requires revisiting an expanding archive of historical recordings. In practice, recordings are distributed across institutions, access is governed locally, and earlier datasets may become unavailable. A useful LEM must therefore absorb new knowledge where the data reside and carry that knowledge forward as participating datasets change, as shown in Fig. 1(B).

These requirements bring together two dimensions of model maintenance: *collaboration across distributed datasets* and *accumulation across time*. Federated learning enables the former through local training and shared model updates (McMahan et al., 2017); continual learning addresses the latter by adapting to incoming data while retaining earlier capabilities (Parisi et al., 2019). Together, they motivate **federated continual learning for pretrained LEMs**. We present the first systematic study of this setting across evolving EEG datasets and task families: each dataset acts as a client, current clients update a shared backbone, and persistent model and memory states carry knowledge between stages.

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.

The central technical question is: ***What evidence should determine which local updates enter the shared EEG model and which remain available for selective reuse?***

Three coupled challenges shape this decision, as summarized in Fig. 1(C). **Dataset heterogeneity.** Differences in montage, spectral content, task labels, and subject populations make correspondence between local EEG representations difficult to establish (Jayaram & Barachant, 2018; Chen et al., 2025b). Sharing therefore requires evidence about the structure underlying an update. **Scale imbalance.** Sample-weighted aggregation gives larger datasets greater influence, although smaller datasets may contribute complementary task structure. Dataset size and cross-client support play different roles in deciding what to consolidate. **Catastrophic forgetting.** As active datasets change, adaptation can overwrite earlier capabilities (Rajpura et al., 2022). Specialized updates need a persistent path through which compatible future clients can reuse them.

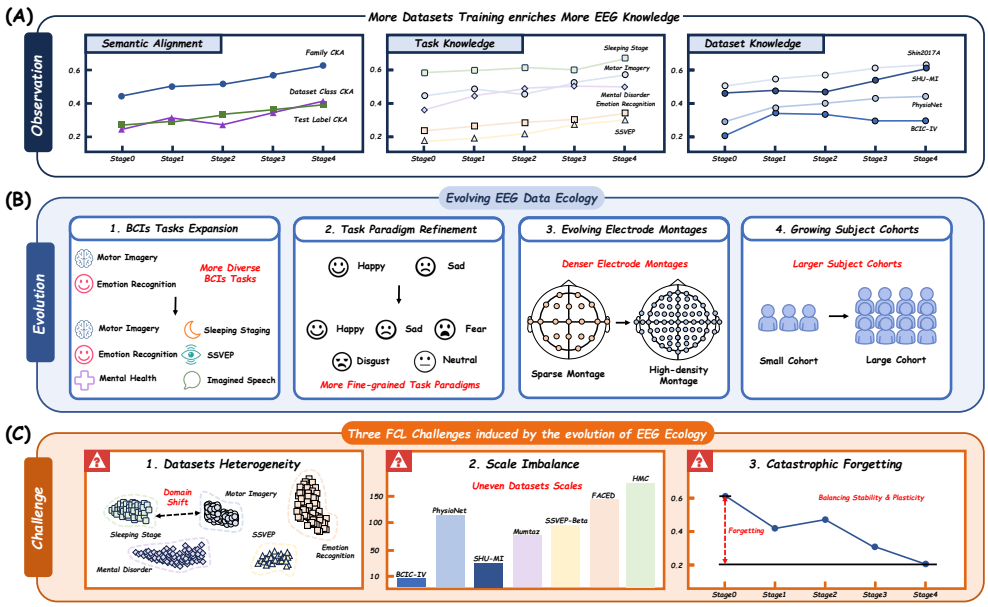


Figure 1: **Motivation of BrainNEST.** (A) Broader EEG data exposure expands the capabilities available to LEMs. (B) New tasks, recording configurations, and subject cohorts motivate continual model maintenance. (C) Federated continual adaptation must jointly address dataset heterogeneity, scale imbalance, and forgetting.

We introduce **BrainNEST**, an *evidence-gated knowledge consolidation* framework. Clients encode channel and frequency-band information at multiple temporal scales into structural sketches linked to parameter-block exposure. Partial cross-client matching forms support groups whose evidence determines both the contribution of each update block and the destination of the group update. Broadly supported groups update the shared backbone; the remaining groups enter routed parameter memory. Structural sketches and acquisition metadata then guide memory retrieval for future client initialization. This links correspondence, consolidation, and reuse through a common evidence interface.

Our contributions are threefold:

- **A new setting for continual LEM maintenance.** We establish the first systematic study of federated continual adaptation of pretrained LEMs across evolving EEG datasets and task families, bringing distributed collaboration and cross-stage knowledge accumulation into one evaluation setting.
- **Evidence-gated knowledge consolidation.** We introduce BrainNEST, which uses cross-client structural support to control blockwise update contributions and their allocation to shared or routed state, preserving parameter offsets for compatibility-based reuse.

- **Evaluation across datasets, tasks, and backbones.** Experiments on 14 EEG datasets, seven BCI task families, and two pretrained backbones demonstrate gains in adaptation and retention over federated and continual learning baselines, together with transfer to unseen datasets.

2 Related Work

EEG model adaptation. NeuroLM, ECHO, and Uni-NTFM broaden EEG pretraining through multi-dataset supervision (Jiang et al., 2024a; Liu et al., 2025; Chen et al., 2025c). EvoBrain extends pretrained EEG models across a sequence of heterogeneous tasks using spectral normalization, distillation, and replay (Zhou et al., 2026). BrainNEST studies adaptation across distributed dataset clients that enter over time, with local prediction heads and persistent server state supporting a shared pretrained backbone.

Federated continual learning. FedWeIT separates global and task-adaptive parameters and transfers task-specific components between clients (Yoon et al., 2021). FOT controls forgetting through orthogonalized updates, while ReFed uses selective replay (Bakman et al., 2023; Li et al., 2024). BrainNEST makes cross-client structural support the basis of consolidation: matched EEG sketches determine blockwise update exposure, support-group evidence governs shared writes, and routed offsets remain addressable by structural and acquisition compatibility.

3 Preliminaries

3.1 Federated Continual EEG Learning

We study federated continual EEG learning where each client is an EEG dataset, such as motor imagery, sleep staging, or emotion recognition. We reserve $\mathcal{S}_t$ for the t-th stage and $\mathcal{C}_t \subseteq \mathcal{C}$ for its active client set. Client i owns trials $x_{in}^t \in \mathbb{R}^{C_i \times S_i \times P_i}$ with dataset-specific labels $y_{in}^t \in \mathcal{Y}_i^t$, where C_i, S_i, and P_i denote channels, temporal patches, and patch length. The server updates a transferable EEG backbone f_Θ, while each dataset keeps its own prediction head h_i. After stage t, an ideal continual learner seeks $\Theta_t^\star \in \arg\min_\Theta \mathcal{J}_{1:t}(\Theta)$, where

$$\mathcal{J}_{1:t}(\Theta) = \frac{1}{t}\sum_{s=1}^{t}\sum_{i\in\mathcal{C}_s}\frac{N_i^s}{\sum_{j\in\mathcal{C}_s}N_j^s}\frac{1}{N_i^s}\sum_{n=1}^{N_i^s}\mathcal{L}_i(h_i(f_\Theta(x_{in}^s)), y_{in}^s)\,. \tag{1}$$

Here N_i^s is the number of training trials on client i at stage s, and $\mathcal{L}_i$ is the dataset-specific supervised loss. Since only $\mathcal{C}_t$ is accessible when stage t arrives, our goal is to adapt to incoming EEG datasets while preserving transferable structure from earlier stages through memory states.

4 Methods

4.1 Framework Overview

BrainNEST couples three operations through structural evidence, as shown in Fig. 2. **Multiscale EEG evidence encoding** summarizes channel and frequency-band structure in persistent latent slots, termed *families*. Each slot carries a sketch and an attribution over transferable parameter blocks. **Negotiated update consolidation** forms cross-client support groups and converts their evidence into blockwise update contributions (Eq. (8)) and shared writes (Eq. (10)). **Compatibility-routed parameter memory** retains the remaining group updates as offsets and retrieves them to initialize future clients (Eq. (11)). Consensus memory provides a complementary reference for local representation alignment.

4.2 Multiscale EEG Evidence Encoding

EEG evidence combines spatial channel information, frequency-band content, and multiple temporal scales. For client i, the personalized backbone produces tokens $H_i^t = f_{\widetilde{\Theta}_i^t}(x_i^t)$, where $\widetilde{\Theta}_i^t$ combines

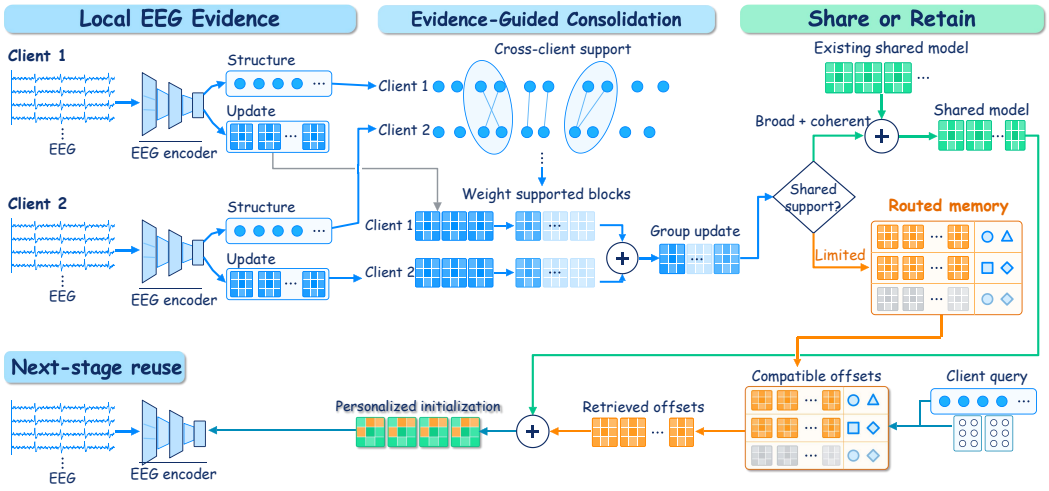


Figure 2: **Evidence-gated knowledge consolidation in BrainNEST.** Current clients encode multi-scale EEG evidence. Cross-client correspondences form support groups and determine the parameter blocks contributed to each group update. The sharing criterion directs these updates to the shared backbone or routed parameter memory. At the next stage, consensus memory guides local alignment and retrieved offsets personalize initialization.

the shared backbone with retrieved parameter offsets. The encoder forms local structural atoms from channel and band streams:

$$\begin{aligned} U_i^t &= P_{\mathrm{ch}}(H_i^t), \qquad V_i^t = P_{\mathrm{band}}(x_i^t), \\ a_{im}^{(r),t} &= P_{\mathrm{mix}}^{(r)}\left[\mathrm{Pool}_r(U_i^t)_m; \mathrm{Pool}_r(V_i^t)_m\right], \\ \gamma_{imj}^t &\propto \alpha_{im}^t \, \mathbf{1}\{j \in \mathrm{TopK}(a_{im}^t, \mathcal{M}_i^{t-1})\} \exp\big(\cos(a_{im}^t, p_{ij}^{t-1})\big). \end{aligned} \tag{2}$$

Here U_i^t and V_i^t are channel and band streams, $a_{im}^{(r),t}$ is a scale-r atom, and a_{im}^t collects all scales. The prototype p_{ij}^{t-1} anchors slot j, α_{im}^t is atom saliency, and γ_{imj}^t is a sparse top-K assignment normalized with a dustbin. Each active family slot is first summarized by its content, support mass, and prototype-relative motion:

$$z_{ij}^t = \frac{\sum_m \gamma_{imj}^t a_{im}^t}{\sum_m \gamma_{imj}^t}, \qquad \mu_{ij}^t = \sum_m \gamma_{imj}^t, \qquad q_{ij}^t = W_q(z_{ij}^t - p_{ij}^{t-1}). \tag{3}$$

Here z_{ij}^t, μ_{ij}^t, and q_{ij}^t denote the family content, support, and motion sketch, respectively. The client then updates the corresponding family prototype to preserve slot identity over time:

$$p_{ij}^t \leftarrow (1-\eta_m) p_{ij}^{t-1} + \eta_m z_{ij}^t, \qquad j \in \mathcal{A}_i^t. \tag{4}$$

Here $\mathcal{A}_i^t$ is the active family set, and η_m controls how quickly the local memory follows the current summary. Finally, the client uploads only family summaries and the transferable-block update:

$$m_{i,\uparrow}^t = (\Delta_i^t, \mathcal{U}_i^t), \qquad \mathcal{U}_i^t = \{(z_{ij}^t, q_{ij}^t, \mu_{ij}^t, \varrho_{ij}^t) : j \in \mathcal{A}_i^t\}. \tag{5}$$

Here Δ_i^t is the transferable update, while ϱ_{ij} is a family-to-block attribution that records which transferable blocks are exposed by family j; unsupported blocks are attenuated before any server write. Dataset heads remain local. Server scoring uses $\omega_{ij}^t = \mu_{ij}^t / \sum_{\ell \in \mathcal{A}_i^t} \mu_{i\ell}^t$ and $\bar{q}_i^t = \sum_{j \in \mathcal{A}_i^t} \omega_{ij}^t q_{ij}^t$.

4.3 Negotiated Update Consolidation

The server turns cross-client correspondence evidence into group updates and storage decisions. It first compares each family pair using content proximity, motion compatibility, and support consistency:

$$C^t_{ii'}(j,\ell) = \frac{1-\cos(z^t_{ij}, z^t_{i'\ell})}{2} + \frac{1-\cos(q^t_{ij}, q^t_{i'\ell})}{2} + |\omega^t_{ij} - \omega^t_{i'\ell}|. \tag{6}$$

The three terms are raw bounded distances in $[0,1]$, measuring content, motion, and support mismatch without pair-specific rescaling. Since many families have no reliable counterpart, the server transports a fixed partial budget $b^t_{ii'} = \rho_{\mathrm{pt}} \min\{\|\boldsymbol{\omega}^t_i\|_1, \|\boldsymbol{\omega}^t_{i'}\|_1\}$:

$$T^t_{ii'} = \arg\min_{T\geq 0} \langle C^t_{ii'}, T\rangle + \epsilon H(T), \qquad T\mathbf{1} \leq \boldsymbol{\omega}^t_i, \quad T^\top \mathbf{1} \leq \boldsymbol{\omega}^t_{i'}, \quad \|T\|_1 = b^t_{ii'}. \tag{7}$$

Here $\boldsymbol{\omega}^t_i = [\omega^t_{ij}]_{j\in\mathcal{A}^t_i}$, $H(T)$ is entropic smoothing, and $0 < \rho_{\mathrm{pt}} < 1$ is fixed. The mass constraint removes the zero-plan degeneracy, while the partial budget leaves unmatched remainder outside negotiation. Retained edges first form candidate components; the server validates them by recursively removing weakest bridge edges, so transitive chains such as A–B–C do not become one sharing group. The validated groups receive client-balanced evidence weights α^G_i, while unvalidated singletons are routed and never update Θ. Before any parameter write, BrainNEST exposes only the part of each client update supported by the families participating in group G. Let π^G_{ij} be the client-normalized transported participation of family j in G. Splitting the transferable update into L blocks, the server forms

$$\varrho^t_{i\to G} = \mathrm{clip}_{[0,1]}\left(\sum_{j\in\mathcal{A}^t_i} \pi^G_{ij}\varrho^t_{ij}\right), \qquad \Delta^{t,(\ell)}_{i\to G} = \varrho^t_{i\to G,\ell}\Delta^{t,(\ell)}_i, \qquad \Delta^t_G = \sum_{i\in G} \alpha^G_i \Delta^t_{i\to G}. \tag{8}$$

This exposure step attenuates unsupported parameter blocks before the group update reaches either shared consolidation or routed memory.

Shareability Gate and Memory Writing. Each cross-client group receives a sharing score:

$$a^t_G = \bar{m}^t_G \frac{|\mathcal{E}^t_G|}{|\mathcal{E}^t|} \exp(-\tilde{\sigma}^t_G). \tag{9}$$

Here $e_i = \phi_{\mathrm{acq}}(m_i)$ is a non-semantic acquisition descriptor that excludes task family, label count, and dataset size; $\bar{m}^t_G$ is quality-adjusted transported support, $|\mathcal{E}^t_G|/|\mathcal{E}^t|$ measures dataset-client coverage rather than sample volume, and $\tilde{\sigma}^t_G$ is normalized sketch dispersion around $\bar{q}^t_G = \sum_{i\in G} \alpha^G_i \bar{q}^t_i$. The score combines transported support, client coverage, and sketch coherence. The shareability gate keeps only groups whose score exceeds a fixed threshold as $\mathcal{G}^t_{\mathrm{sh}} = \{G : a^t_G \geq \tau_{\mathrm{sh}}\}$. Here τ_{sh} controls how conservative the server is when writing to the shared path. Shared groups refresh the consensus memory B^{t+1}, while the remaining candidates are handed to routed memory. Finally, shareable groups update the backbone through score-weighted aggregation:

$$\Theta^{t+1} = \Theta^t + \frac{\sum_{G\in\mathcal{G}^t_{\mathrm{sh}}} a^t_G \Delta^t_G}{\sum_{G\in\mathcal{G}^t_{\mathrm{sh}}} a^t_G}. \tag{10}$$

Here a^t_G weighs each support group, and Δ^t_G is the family-exposed group update from Eq. (8). If $\mathcal{G}^t_{\mathrm{sh}} = \emptyset$, we set $\Theta^{t+1} = \Theta^t$.

Feedback and Personalized Broadcast. At the next stage, the server reads routed memory $\mathcal{R}^{t+1} = \{(\Xi^{t+1}_r, \kappa^{t+1}_r, \tau^{t+1}_r)\}_{r=1}^{K_{t+1}}$ to personalize each active client. Compatibility combines the latest client sketch $\hat{q}^t_i$ with the non-semantic acquisition descriptor e_i, omitting the sketch term when unavailable:

$$\begin{aligned} s^{t+1}_{ir} &= \cos(\hat{q}^t_i, \kappa^{t+1}_r) + \cos(e_i, \tau^{t+1}_r), \\ A^{t+1}_i &= \mathrm{TopK}_r(s^{t+1}_{ir}), \qquad \alpha^{t+1}_{ir} = \mathrm{softmax}_{r\in A^{t+1}_i}(s^{t+1}_{ir}), \\ \widetilde{\Theta}^{t+1}_i &= \Theta^{t+1} + \lambda_{\mathrm{mem}} \sum_{r\in A^{t+1}_i} \alpha^{t+1}_{ir} \Xi^{t+1}_r. \end{aligned} \tag{11}$$

Here A^{t+1}_i is the retrieved memory set, α^{t+1}_{ir} are readout weights, and λ_{mem} controls routed readout strength. The consensus memory B^{t+1} is broadcast separately for local alignment.

4.4 Compatibility-Routed Parameter Memory

Groups below the sharing threshold and singleton groups enter routed parameter memory, preserving their updates for later use. Each entry stores a parameter offset, a sketch key, and an acquisition anchor. For a group G, with sketch $\bar{q}_G^t$ and acquisition centroid $\bar{e}_G^t = \sum_{i \in G} \alpha_i^G e_i$, the server retrieves the closest slot:

$$r^\star = \arg\max_r \left[\cos(\bar{q}_G^t, \kappa_r^t) + \cos(\bar{e}_G^t, \tau_r^t) \right]. \tag{12}$$

Here κ_r^t and τ_r^t are the sketch key and acquisition anchor of routed slot r, so retrieval jointly checks update compatibility and non-semantic acquisition compatibility. The selected slot then absorbs the routed group update through a momentum write:

$$\Xi_{r^\star}^{t+1} \leftarrow (1 - \eta_m)\Xi_{r^\star}^t + \eta_m \Delta_G^t. \tag{13}$$

Here Δ_G^t is the same family-exposed group update, routed. The corresponding key and anchor are updated by the same momentum rule. Thus each routed entry stores (Ξ, κ, τ): the offset to reuse, its structural retrieval key, and its acquisition context.

4.5 Local Objective

Each client trains with a supervised task loss, sketch reconstruction, and consensus alignment:

$$\mathcal{L}_i^t = \mathcal{L}_{\text{task}} + \lambda_{\text{aux}} \left[\left\| g_{\text{dec}}(\{z_{ij}^t, \mu_{ij}^t\}_{j \in \mathcal{A}_i^t}) - U_i^t \right\|_2^2 + \chi_B^t \sum_{j \in \mathcal{A}_i^t} \omega_{ij}^t \Big(1 - \cos\big(z_{ij}^t, r(z_{ij}^t; B^t)\big) \Big) \right]. \tag{14}$$

Here $\mathcal{L}_{\text{task}}$ is the dataset-specific supervised loss, $\chi_B^t = \mathbf{1}\{B^t \neq \emptyset\}$, and λ_{aux} is the single auxiliary weight. The reconstruction term keeps family summaries faithful to the channel-collapsed latent trajectory U_i^t in Eq. (2), while the consensus-alignment term is activated only after B^t exists. The consensus reference is retrieved by attention over the consensus memory:

$$r(z; B^t) = \text{norm}\left(\sum_m \frac{\exp(z^\top b_m^t)}{\sum_\ell \exp(z^\top b_\ell^t)} b_m^t \right). \tag{15}$$

Here $r(z; B^t)$ is the consensus target for family z, and b_m^t denotes the m-th consensus entry. Routed memory affects training through the personalized initialization in Eq. (11). Appendix C characterizes the matching budget, client balance, blockwise exposure, sharing threshold, and retained-memory ancestry. Appendix D.2 gives the communication, memory, and server-compute costs.

5 Experiments and Results

5.1 Datasets and Stage Construction

To evaluate BrainNEST across diverse EEG scenarios, we use 14 downstream datasets spanning seven representative BCI tasks: (1) emotion recognition, (2) motor imagery, (3) sleep staging, (4) SSVEP, (5) imagined speech, (6) mental disorder diagnosis, and (7) mental stress detection. We keep each EEG dataset as one client and organize the 14 datasets into four temporal stages according to release period, as summarized in Table 1. During each stage, only datasets assigned to that stage participate in federated training. Across stages, the active client set changes, creating a federated continual learning stream over evolving EEG datasets. Detailed dataset statistics, task paradigms, preprocessing procedures, and split protocols are provided in Appendix E.3.

Table 1: Summary of different stage data configurations.

Stage	Dataset	Paradigm	Year	Classes	Subjects	Ch.
Stage 1	BCIC IV-2a (Tangermann et al., 2012)	Motor imagery	2008	4	9	22
	PhysioNet-MI (Schalk et al., 2004)	Motor imagery	2009	4	109	64
	DEAP-VA (Koelstra et al., 2011)	Emotion recognition	2012	4	32	32
	SEED (Zheng & Lu, 2015)	Emotion recognition	2013	2	15	62
Stage 2	Mumtaz (Mumtaz et al., 2018)	Mental Disorder Diagnosis	2016	2	64	19
	Shin2017A (Shin et al., 2016)	Motor imagery	2017	2	29	30
	MentalArithmetic (Zyma et al., 2019)	Mental stress detection	2019	2	36	23
Stage 3	BCIC2020-3 (Jeong et al., 2022)	Imagined speech	2020	5	15	64
	BETA (Liu et al., 2020)	SSVEP	2020	40	70	64
	BCIC-UpperLimb (Jeong et al., 2020)	Motor imagery	2020	3	9	22
	SEED-V (Liu et al., 2021)	Emotion recognition	2021	5	16	62
Stage 4	HMC (Alvarez-Estevez & Rijsman, 2021)	Sleep staging	2022	5	151	4
	SHU-MI (Ma et al., 2022)	Motor imagery	2022	2	25	32
	FACED (Chen et al., 2023)	Emotion recognition	2023	9	123	32

5.2 Baselines and Metrics

Large EEG Model backbones. We evaluate BrainNEST with two pretrained Large EEG Model backbones, CBraMod (Wang et al., 2024b) and CodeBrain (Ma et al., 2025). For each model, we initialize from the publicly released pretrained checkpoints. For fair comparison, all methods use the same backbone and communication schedule. More details about LEM backbones are provided in Appendix E.1.

Baselines. For single-stage federated learning, we compare against local training and centralized training, standard FL methods including FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), and SCAFFOLD (Karimireddy et al., 2020), personalized FL methods including FedBABU (Oh et al., 2021), FedPer (Arivazhagan et al., 2019), FedROD (Chen & Chao, 2021), FedAMP (Huang et al., 2021), FedALA (Zhang et al., 2023b), and FeDaL (Chen et al., 2025a). For federated continual learning, we compare with representative continual and federated continual baselines. For Baseline, EWC (Kirkpatrick et al., 2017), LWF (Li & Hoiem, 2017), and OGD (Farajtabar et al., 2020), we use FedAvg as the within-stage federated training procedure and apply the corresponding continual learning mechanism across stages. We further compare with federated continual methods including FedWEIT (Yoon et al., 2021), FOT (Bakman et al., 2023), and ReFed (Li et al., 2024). Additional implementation details are provided in Appendix F.1.

Metrics. For downstream classification, we report Balanced Accuracy (BA) and F1 scores. We additionally report Cohen's κ, AUROC, and PR-AUC. For federated continual learning, we report final Anytime Average Accuracy (AAA), New-Stage Accuracy (New), Average Forgetting (Fgt.), and Backward Transfer (BWT), where higher is better except for Fgt. Detailed splits, preprocessing, implementation settings, and metric definitions are provided in Appendices E.3, F.1, and E.2.

Controlled protocol. Each stage uses 120 communication rounds with one local epoch per round. All methods share the dataset splits, pretrained backbone, local optimizer, and communication schedule. Dataset-specific prediction heads remain local. In BrainNEST, only the server model and memory states persist across stages; historical raw EEG is not revisited after a client leaves the active set. Appendix F.1 specifies the preprocessing, training, and evaluation settings.

5.3 Single-Stage Federated Learning

We evaluate BrainNEST within each stage by treating the datasets in that stage as concurrent federated clients. Table 2 shows that BrainNEST achieves the best results across all datasets and metrics. The improvement is consistent across diverse tasks, suggesting that BrainNEST is not tailored to a single EEG paradigm. Standard FL baselines are sensitive to dataset heterogeneity, while personalized and adaptive FL methods improve selected regimes but remain inconsistent across client ecologies. In contrast, BrainNEST provides a stronger sharing interface by negotiating family-supported evidence across clients, yielding more reliable transfer. Detailed results for each stage are provided in the Appendix.

Table 2: Single-stage federated learning results on representative EEG datasets of each stage.

Method	BCIC IV-2a			PhysioNet-MI			DEAP-VA			SEED		
	BA	WF1	κ	BA	WF1	κ	BA	WF1	κ	BA	AUROC	F1
Single Finetune	0.460	0.431	0.280	0.607	0.605	0.476	0.317	0.325	0.083	0.668	0.751	0.689
Central	0.530	0.528	0.373	0.597	0.598	0.463	0.310	0.323	0.079	0.705	0.760	0.707
FedAvg	0.538	0.530	0.384	0.608	0.609	0.478	0.319	0.324	0.084	0.689	0.766	0.698
FedProx	0.531	0.526	0.375	0.609	0.608	0.479	0.301	0.324	0.067	0.705	0.775	0.692
SCAFFOLD	0.432	0.359	0.243	0.611	0.613	0.482	0.328	0.327	0.098	0.698	0.777	0.687
FedPer	0.516	0.505	0.354	0.613	0.614	0.485	0.310	0.310	0.076	0.690	0.777	0.711
FedBABU	0.487	0.461	0.316	0.605	0.606	0.474	0.339	0.355	0.122	0.696	0.755	0.698
FedROD	0.517	0.510	0.357	0.618	0.617	0.491	0.322	0.343	0.102	0.711	0.769	**0.720**
FedAMP	0.510	0.495	0.347	0.608	0.605	0.478	0.326	0.313	0.097	0.688	0.766	0.705
FedALA	0.530	0.520	0.374	0.617	0.619	0.489	0.313	0.312	0.077	0.689	0.743	0.683
FeDaL	0.532	0.526	0.376	0.611	0.609	0.481	0.336	0.329	0.110	0.719	0.773	0.700
BrainNEST (Ours)	**0.576**	**0.578**	**0.435**	**0.626**	**0.626**	**0.502**	**0.349**	**0.363**	**0.131**	**0.725**	**0.782**	**0.720**

Method	Mumtaz			MentalArithmetic			FACED			HMC		
	BA	AUROC	F1	BA	AUROC	F1	BA	WF1	κ	BA	WF1	κ
Single Finetune	0.878	0.960	0.870	0.708	0.790	0.568	0.527	0.528	0.465	0.718	0.775	0.698
Central	0.884	0.955	0.877	0.719	0.833	0.584	0.483	0.486	0.416	0.720	0.772	0.706
FedAvg	0.892	0.957	0.886	0.698	0.826	0.547	0.498	0.501	0.433	0.723	0.771	0.696
FedProx	0.897	0.959	0.887	0.701	0.803	0.551	0.503	0.438	0.438	0.714	0.759	0.696
SCAFFOLD	0.886	0.924	0.880	0.688	0.821	0.533	0.491	0.493	0.425	0.717	0.758	0.675
FedPer	0.889	0.962	0.885	0.681	0.780	0.522	0.519	0.520	0.456	0.730	0.775	0.690
FedBABU	0.877	0.918	0.867	0.701	0.821	0.549	0.513	0.513	0.449	0.722	0.769	0.686
FedROD	0.885	0.965	0.879	0.708	0.803	0.560	0.520	0.522	0.458	0.726	0.773	0.693
FedAMP	0.882	0.959	0.879	0.712	0.767	0.545	0.536	0.533	0.473	0.716	0.760	0.676
FedALA	0.898	0.968	0.897	0.722	0.838	0.577	0.540	0.540	0.479	0.724	0.759	0.684
FeDaL	0.894	0.967	0.893	0.705	0.788	0.551	0.531	0.531	0.469	0.729	0.767	0.693
BrainNEST (Ours)	**0.918**	**0.982**	**0.916**	**0.729**	**0.841**	**0.600**	**0.544**	**0.544**	**0.484**	**0.745**	**0.791**	**0.721**

5.4 Federated Continual Learning

Table 3 reports full-stage results under CBraMod and CodeBrain. BrainNEST achieves the best performance on every metric, improving new-stage adaptation while reducing forgetting. This indicates that consensus and routed memories help accumulate EEG knowledge without simply overwriting earlier stages.

Table 3: Full-stage federated continual learning results under two Large EEG Model backbones.

Method	CBraMod				CodeBrain			
	AAA ↑	New ↑	Fgt. ↓	BWT ↑	AAA ↑	New ↑	Fgt. ↓	BWT ↑
Baseline	0.5169	0.5369	0.0594	-0.0578	0.5235	0.5410	0.0600	-0.0586
EWC	0.5237	0.5383	0.0481	-0.0477	0.5306	0.5439	0.0488	-0.0442
LWF	0.5271	0.5377	0.0468	-0.0428	0.5317	0.5430	0.0474	-0.0424
OGD	0.5258	0.5372	0.0415	-0.0400	0.5271	0.5377	0.0468	-0.0428
FedWEIT	0.5390	0.5472	0.0454	-0.0387	0.5425	0.5578	0.0459	-0.0442
FOT	0.5425	0.5491	0.0447	-0.0374	0.5533	0.5674	0.0505	-0.0466
ReFed	0.5478	0.5530	0.0412	-0.0365	0.5583	0.5681	0.0441	-0.0390
BrainNEST (Ours)	**0.5711**	**0.5685**	**0.0353**	**-0.0241**	**0.5832**	**0.5817**	**0.0380**	**-0.0327**

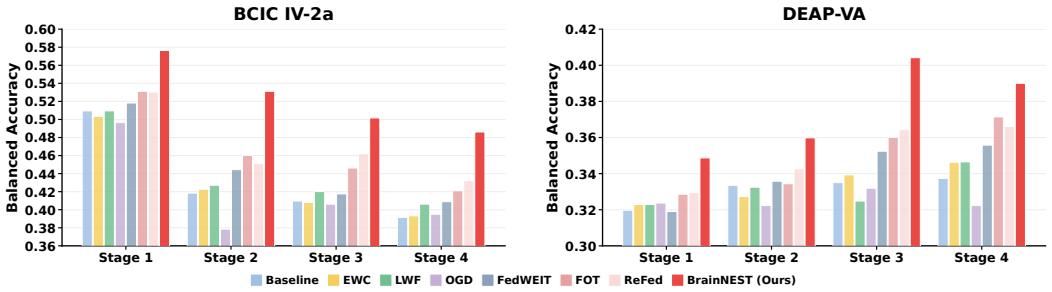


Figure 3: Balanced accuracy on BCIC IV-2a and DEAP-VA across continual stages.

Figure 3 shows that BrainNEST preserves BCIC IV-2a performance and improves DEAP-VA over later stages, indicating stable retention with progressive knowledge accumulation.

5.5 Ablation Study

Table 4: Ablation study of BrainNEST. Left: single-stage ablation. Right: continual ablation.

Variant	Single-stage ablation						Continual ablation			
	BCIC-2a			FACED			AAA ↑	New ↑	Fgt. ↓	BWT ↑
	BA	WF1	κ	BA	WF1	κ				
BrainNEST	**0.5764**	**0.5775**	**0.4352**	**0.5443**	**0.5441**	**0.4842**	**0.5711**	**0.5685**	**0.0353**	**-0.0241**
w/o Family	0.5095	0.4988	0.3461	0.4956	0.4947	0.4290	0.5237	0.5330	0.0417	-0.0414
w/o Sparse	0.5243	0.5174	0.3657	0.5223	0.5204	0.4586	0.5369	0.5441	0.0395	-0.0376
w/o Gate	0.5191	0.5111	0.3588	0.5007	0.5009	0.4343	0.5421	0.5486	0.0533	-0.0401
w/o Routed	0.5252	0.5143	0.3669	0.5256	0.5255	0.4630	0.5411	0.5514	0.0414	-0.0395
w/o Feedback	0.5365	0.5320	0.3819	0.5337	0.5307	0.4700	0.5534	0.5568	0.0407	-0.0336

Table 4 shows that removing family sketching causes the largest drop, confirming that family summaries are the key interface for exposing supported update components. Sparse transport and the shareability gate make transfer selective, while routed memory and consensus feedback improve adaptation and retention. The full model achieves the best plasticity–stability trade-off, indicating that BrainNEST needs both negotiated sharing and persistent routed memory. With CBraMod, removing family sketching reduces AAA from 0.5711 to 0.5237, a decrease of 4.74 percentage points. Removing the shareability gate raises forgetting from 0.0353 to 0.0533, while removing routed memory reduces AAA to 0.5411. These changes separate the roles of evidence construction, selective shared writing, and reusable specialization, and support keeping all three within the same consolidation interface.

5.6 Unseen Dataset Transfer

Table 5: Results of Unseen Dataset Transfer.

Method	eldBETA			TUAB		
	BA	WF1	κ	BA	PR-AUC	ROC-AUC
EEGNet (Lawhern et al., 2018)	0.5841	0.5865	0.5321	0.7618	0.8121	0.8340
CNN-Transformer (Peh et al., 2022)	0.5437	0.5437	0.4866	0.7724	0.8233	0.8380
EEGPT (Wang et al., 2024a)	0.5706	0.5728	0.5170	0.7695	0.8173	0.8407
LaBraM (Jiang et al., 2024b)	0.5365	0.5356	0.5365	0.7742	0.8109	0.8416
CBraMod (Wang et al., 2024b)	0.5897	0.5896	0.5384	0.7886	0.8675	0.8591
CodeBrain (Ma et al., 2025)	0.5992	0.5991	0.5491	0.7919	0.8765	0.8740
Uni-NTFM (Chen et al., 2025c)	0.6175	0.6169	0.5696	0.7965	0.8755	0.8687
NeuroLM (Jiang et al., 2024a)	0.5841	0.5860	0.5321	0.7810	0.8266	0.8449
ECHO (Liu et al., 2025)	0.6190	0.6190	0.5714	0.7986	0.8838	**0.8772**
BrainNEST (CBraMod)	0.6238	0.6239	0.5768	0.8016	**0.8967**	0.8741
BrainNEST (CodeBrain)	**0.6254**	**0.6251**	**0.5786**	**0.8031**	0.8959	0.8771

Table 5 evaluates two unseen datasets, eldBETA (Liu et al., 2022) and TUAB (Obeid & Picone, 2016). All methods use the same unseen split, 100-epoch full fine-tuning budget, and evaluation-head protocol; BrainNEST differs only by initializing from its final continual backbone. BrainNEST leads all eldBETA metrics and obtains the best BA and PR-AUC on TUAB, suggesting that stage-wise learning accumulates reusable EEG knowledge beyond the training stream.

6 Conclusion

We introduce federated continual adaptation of pretrained Large EEG Models across evolving datasets and task families. BrainNEST uses cross-client structural evidence to control blockwise update contributions, consolidate shared knowledge, and retain routed parameter offsets for future reuse. Across 14 datasets and two pretrained backbones, it improves adaptation, retention, and transfer to unseen datasets. These results support continual model maintenance as a practical way to accumulate knowledge from growing, distributed EEG resources.

AI USE STATEMENT

LLMs were used for writing assistance, wording and presentation revision, and LaTeX template migration and formatting. LLMs are not used as an important, original, or non-standard component of the proposed method or experiments. BrainNEST operates on EEG backbones, family summaries, consensus negotiation, and routed memory. The authors take full responsibility for all text, analyses, results, and conclusions presented in this paper.

REFERENCES

Diego Alvarez-Estevez and Roselyne M Rijsman. Inter-database validation of a deep learning approach for automatic sleep scoring. *PloS one*, 16(8):e0256111, 2021.

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Yahya H Ezzeldin, and Salman Avestimehr. Federated orthogonal training: Mitigating global catastrophic forgetting in continual federated learning. *arXiv preprint arXiv:2309.01289*, 2023.

Hubert Banville, Omar Chehab, Aapo Hyvärinen, Denis-Alexander Engemann, and Alexandre Gramfort. Uncovering the structure of clinical eeg signals with self-supervised learning. *Journal of Neural Engineering*, 18(4):046020, 2021.

Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. *arXiv preprint arXiv:2107.00778*, 2021.

Jingjing Chen, Xiaobin Wang, Chen Huang, Xin Hu, Xinke Shen, and Dan Zhang. A large finer-grained affective computing eeg dataset. *Scientific Data*, 10(1):740, 2023.

Shengchao Chen, Guodong Long, and Jing Jiang. Fedal: Federated dataset learning for time series foundation models. *arXiv preprint arXiv:2508.04045*, 2025a.

Zhige Chen, Chengxuan Qin, Wenlong You, Rui Liu, Congying Chu, Rui Yang, Kay Chen Tan, and Jibin Wu. Hear: An eeg foundation model with heterogeneous electrode adaptive representation. *arXiv preprint arXiv:2510.12515*, 2025b.

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. *arXiv preprint arXiv:2509.24222*, 2025c.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pp. 3762–3773. PMLR, 2020.

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

Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 7865–7873, 2021.

Vinay Jayaram and Alexandre Barachant. Moabb: trustworthy algorithm benchmarking for bcis. *Journal of neural engineering*, 15(6):066011, 2018.

Ji-Hoon Jeong, Jeong-Hyun Cho, Kyung-Hwan Shim, Byoung-Hee Kwon, Byeong-Hoo Lee, Do-Yeun Lee, Dae-Hyeok Lee, and Seong-Whan Lee. Multimodal signal dataset for 11 intuitive movement tasks from single upper extremity during multiple recording sessions. *GigaScience*, 9 (10):giaa098, 2020.

Ji-Hoon Jeong, Jeong-Hyun Cho, Young-Eun Lee, Seo-Hyun Lee, Gi-Hwan Shin, Young-Seok Kweon, José del R Millán, Klaus-Robert Müller, and Seong-Whan Lee. 2020 international brain–computer interface competition: A review. *Frontiers in human neuroscience*, 16:898300, 2022.

Wei-Bang 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. *arXiv preprint arXiv:2409.00101*, 2024a.

Wei-Bang Jiang, Li-Ming Zhao, and Bao-Liang Lu. Large brain model for learning generic representations with tremendous eeg data in bci. *arXiv preprint arXiv:2405.18765*, 2024b.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

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.

Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. Deap: A database for emotion analysis; using physiological signals. *IEEE transactions on affective computing*, 3(1):18–31, 2011.

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.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Yichen Li, Qunwei Li, Haozhao Wang, Ruixuan Li, Wenliang Zhong, and Guannan Zhang. Towards efficient replay in federated incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12820–12829, 2024.

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

Bingchuan Liu, Xiaoshan Huang, Yijun Wang, Xiaogang Chen, and Xiaorong Gao. Beta: A large benchmark database toward ssvep-bci application. *Frontiers in neuroscience*, 14:627, 2020.

Bingchuan Liu, Yijun Wang, Xiaorong Gao, and Xiaogang Chen. eldbeta: A large eldercare-oriented benchmark database of ssvep-bci for the aging population. *Scientific Data*, 9, 2022. URL `https://api.semanticscholar.org/CorpusID:249233408`.

Chenyu Liu, Yuqiu Deng, Tianyu Liu, Jinan Zhou, Xinliang Zhou, Ziyu Jia, and Yi Ding. Echo: Toward contextual seq2seq paradigms in large eeg models. *arXiv preprint arXiv:2509.22556*, 2025.

Wei Liu, Jie-Lin Qiu, Wei-Long Zheng, and Bao-Liang Lu. Comparing recognition performance and robustness of multimodal deep learning models for multimodal emotion recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 2021.

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

Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. A review of classification algorithms for eeg-based brain–computer interfaces: a 10 year update. *Journal of neural engineering*, 15(3):031005, 2018.

Wei Lu, Haiyan Liu, Hua Ma, Tien-Ping Tan, and Lingnan Xia. Hybrid transfer learning strategy for cross-subject eeg emotion recognition. *Frontiers in Human Neuroscience*, 17:1280241, 2023.

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*, 2025.

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(1):531, 2022.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. Pmlr, 2017.

Wajid Mumtaz, Syed Saad Azhar Ali, Mohd Azhar Mohd Yasin, and Aamir Saeed Malik. A machine learning framework involving eeg-based functional connectivity to diagnose major depressive disorder (mdd). *Medical & biological engineering & computing*, 56(2):233–246, 2018.

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

Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. *arXiv preprint arXiv:2106.06042*, 2021.

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. *arXiv preprint arXiv:2510.21585*, 2025.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Wei Yan Peh, Yuanyuan Yao, and Justin Dauwels. Transformer convolutional neural networks for automated artifact detection in scalp eeg. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 3599–3602. IEEE, 2022.

Param Rajpura, Pankaj Pandey, and Krishna Miyapuram. Continual learning for eeg based brain computer interfaces. In *Continual Lifelong Learning Workshop at ACML 2022*, 2022.

Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. Deep learning-based electroencephalography analysis: a systematic review. *Journal of neural engineering*, 16(5):051001, 2019.

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.

Jaeyoung Shin, Alexander von Lühmann, Benjamin Blankertz, Do-Won Kim, Jichai Jeong, Han-Jeong Hwang, and Klaus-Robert Müller. Open access dataset for eeg+ nirs single-trial classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(10):1735–1745, 2016.

Michael Tangermann, Klaus-Robert Müller, Ad Aertsen, Niels Birbaumer, Christoph Braun, Clemens Brunner, Robert Leeb, Carsten Mehring, Kai J Miller, Gernot R Müller-Putz, et al. Review of the bci competition iv. *Frontiers in neuroscience*, 6:55, 2012.

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

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. *arXiv preprint arXiv:2412.07236*, 2024b.

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.

Jonathan R Wolpaw. Brain-computer interfaces (bcis) for communication and control: a mini-review. *Supplements to Clinical neurophysiology*, 57:607–613, 2004.

Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *International conference on machine learning*, pp. 12073–12086. PMLR, 2021.

Daoze Zhang, Zhizhang Yuan, Yang Yang, Junru Chen, Jingjing Wang, and Yafeng Li. Brant: Foundation model for intracranial neural signal. *Advances in Neural Information Processing Systems*, 36:26304–26321, 2023a.

Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11237–11244, 2023b.

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.

Yangxuan Zhou, Sha Zhao, Jiquan Wang, Shijian Li, and Gang Pan. EvoBrain: Continual learning of EEG foundation models across heterogeneous BCI tasks. *arXiv preprint arXiv:2606.01767*, 2026. doi: 10.48550/arXiv.2606.01767. URL https://arxiv.org/abs/2606.01767.

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. *arXiv preprint arXiv:2506.23075*, 2025.

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 • Additional Related Work . . . 14

B • Method Details . . . 15

B.1 • *EEG Evidence Encoding and Local Slot Maintenance* . . . 16

B.2 • *Server Negotiation, Routing, and Stage Readout* . . . 18

C • Operator-Level Properties . . . 23

C.1 • *Partial Transport Avoids Forced Misalignment* . . . 23

C.2 • *Shareability Gate Has an Evidence–Coherence Threshold* . . . 24

C.3 • *Client-Balanced Writing Limits Scale-Dominant Aggregation* . . . 26

C.4 • *Family-Conditioned Exposure Is Non-Expansive* . . . 27

C.5 • *Routed Specialization Has Positive Finite-Ancestry Retention* . . . 27

D • Design Rationale and Practical Cost . . . 29

D.1 • *Evidence-Gated Consolidation and Routed Reuse* . . . 29

D.2 • *Communication, Memory, and Scalability* . . . 30

E • Experimental Details . . . 33

E.1 • *Large EEG Model Backbones* . . . 33

E.2 • *Evaluation Metrics* . . . 33

E.3 • *Dataset Details, Preprocessing, and Label Semantics* . . . 34

E.4 • *Single-Stage Federated Learning Baselines* . . . 37

E.5 • *Federated Continual Learning Baselines* . . . 38

F • Additional Experimental Details and Results . . . 39

F.1 • *Implementation Details and Hyperparameters* . . . 39

F.2 • *Stage-wise Single-Stage Federated Learning Results* . . . 40

F.3 • *Federated Continual Learning Dynamics* . . . 42

G • Broader Impact . . . 42

H • Future Work . . . 43

A Additional Related Work

Large EEG foundation models. Recent Large EEG Models move EEG modeling from small task-specific decoders toward reusable neural representations learned from broad neural recordings. Early self-supervised EEG representation learning demonstrated that temporal context, masking, and contrastive objectives can produce transferable features before downstream supervision (Banville et al., 2021; Kostas et al., 2021; Weng et al., 2025). More recent LEMs further incorporate EEG-specific architectural biases: CBraMod uses criss-cross representations for spatial–temporal modeling (Wang et al., 2024b), CSBrain and CodeBrain emphasize cross-scale or multi-scale tokenization for transferable decoding (Zhou et al., 2025; Ma et al., 2025), and EEGMamba explores state-space modeling for long-range EEG dynamics (Gui et al., 2024). Large-corpus efforts such as LaBraM, EEGPT, and REVE further highlight the importance of scale and heterogeneity in neural representation learning (Jiang et al., 2024b; Wang et al., 2024a; Ouahidi et al., 2025). These works establish the backbone side of the problem: stronger architectures and broader pretraining improve EEG transfer. BrainNEST is complementary, asking how such backbones should be maintained when new EEG datasets arrive over time as distributed clients.

Multi-dataset EEG modeling. Multi-dataset EEG models broaden the data ecology seen during training. NeuroLM uses instruction-style supervision to connect EEG signals with task descriptions (Jiang et al., 2024a), ECHO formulates EEG decoding as contextual sequence-to-sequence learning across datasets (Liu et al., 2025), and Uni-NTFM scales unified pretraining across diverse EEG data sources (Chen et al., 2025c). These studies support a central motivation of our paper: LEM

knowledge is shaped not only by model design, but also by the diversity of tasks, label semantics, recording protocols, and subject populations encountered during training. However, these methods are primarily formulated as centralized or jointly visible multi-dataset training. BrainNEST studies the complementary regime in which datasets arrive stage by stage, raw EEG is not pooled, and useful structure must be accumulated through persistent shared and routed memory.

Federated and federated continual learning. Classical federated learning methods such as FedAvg, FedProx, and SCAFFOLD address distributed optimization under non-IID clients through averaging, proximal control, or variance reduction (McMahan et al., 2017; Li et al., 2020; Karimireddy et al., 2020). Personalized FL methods further separate shared representation learning from client-specific adaptation, which is useful when local distributions differ substantially. Continual and federated continual methods, including EWC, LWF, OGD, FedWEIT, FOT, and ReFed, target stability–plasticity through regularization, distillation, orthogonalization, weighted transfer, or replay-style mechanisms (Kirkpatrick et al., 2017; Li & Hoiem, 2017; Farajtabar et al., 2020; Yoon et al., 2021; Bakman et al., 2023; Li et al., 2024). BrainNEST builds on this line but changes the sharing interface for EEG. Instead of deciding only at the level of full-client parameters or task identities, it constructs latent family evidence, validates cross-client support, exposes only family-supported update components, and stores reusable non-shared directions as routed specialization.

B Method Details

Appendix B specifies the operators used in the main method. Server-facing parameter vectors occupy the transferable trainable block; dataset-specific prediction heads remain local. A *family* is a persistent latent slot that summarizes recurring EEG structure for matching and parameter-block attribution. Table 6 maps the main-text equations to their detailed definitions.

Table 6: Mapping from compact main-text equations to appendix-level operator details.

Main component	Main equations	Appendix expansion
EEG encoding and upload	Eqs. (2)–(5)	Sec. B.1
Negotiation and shared writing	Eqs. (6)–(10)	Sec. B.2
Routed memory write and read	Eqs. (11)–(13)	Sec. B.2
Local feedback objective	Eqs. (14)–(15)	Sec. B.3

Table 7: Core notation for evidence encoding and server-side routing.

Symbol	Shape	Meaning
H_i^t	$C_i \times S_i \times d$	personalized backbone latent tokens
U_i^t	$S_i \times d$	channel-collapsed latent trajectory reconstructed by $\mathcal{L}_{\mathrm{str}}$
V_i^t	$S_i \times d$	band-aware latent stream in the shared backbone basis
a_{im}^t, z_{ij}^t	d	local atom and uploaded family content carrier
q_{ij}^t	d_q	prototype-relative motion sketch from shared W_q
$\mu_{ij}^t, \omega_{ij}^t$	scalar	family support mass and normalized support
ϱ_{ij}^t	L	family-to-block exposure over transferable update blocks
Δ_i^t, Δ_G^t	p	client update and family-exposed group update

Acquisition descriptor. BrainNEST uses a lightweight non-semantic descriptor e_i whenever a module needs acquisition compatibility. Client i registers static acquisition/format metadata

$$m_i = (C_i, S_i, P_i, \upsilon_i, f_i^{\mathrm{sr}}),$$

where C_i, S_i, P_i denote channels, temporal patch count, and patch length; υ_i is the montage/acquisition template and f_i^{sr} the sampling-rate tag after preprocessing. Task family, label count, and training-set size are deliberately excluded from e_i, so routing uses acquisition compatibility independently of task labels. Training-set size can still appear in the standard supervised objective or reporting protocol shared by all methods, and label count remains local to the dataset-specific head.

Continuous fields are encoded as log-scaled z-scores using constants fixed from registration metadata, categorical fields are encoded by learned embeddings, and unseen categorical values use a reserved new-category embedding at registration. The concatenated vector is then projected and normalized:

$$e_i = \phi_{\mathrm{acq}}(m_i) = \mathrm{norm}\left(W_e[\mathrm{zscore}(\log(1+v_i)); E_v(v_i); E_{\mathrm{sr}}(f_i^{\mathrm{sr}})]\right) \in \mathbb{R}^{d_e},$$

with $v_i = (C_i, S_i, P_i)$. The descriptor is registered once when a client enters and contains no raw EEG, trial-level statistics, task-family tag, label semantics, or dataset-size cue. It conditions local saliency and routed-memory retrieval, while the coverage term is computed from client identifiers rather than descriptor values. Dataset-specific prediction heads remain local.

B.1 EEG Evidence Encoding and Local Slot Maintenance

This subsection makes explicit the client-side map from personalized latent state to compact family summary. The encoder takes the current latent H_i^t and the previous family memory $\mathcal{M}_i^{t-1}$, preserves stable family identities, and outputs the updated memory $\mathcal{M}_i^t$ together with the uploaded summary $\mathcal{U}_i^t$ and family-conditioned update exposure. All quantities uploaded for server comparison or writing, including z_{ij}^t, q_{ij}^t, and ϱ_{ij}^t, are defined in the shared transferable latent basis; only g_{dec} and prediction heads are client-local.

Given a local sample $x \sim \mathcal{D}_i^t$, the personalized backbone produces

$$H_i^t = f_{\widetilde{\Theta}_i^t}(x) \in \mathbb{R}^{C_i \times S_i \times d},$$

where C_i is the channel count, S_i the temporal patch count, and d the latent width. We write the encoder at the per-sample level, while in practice the bank is updated across local training and $\mathcal{U}_i^t$ is uploaded at stage end. Candidate local atoms are obtained by combining a channel-collapsed stream with a band-aware stream. Let

$$\bar{h}_{ic}^t = \frac{1}{S_i}\sum_{s=1}^{S_i} H_i^t[c,s], \qquad \beta_{ic}^t = \frac{\exp(\nu^\top \bar{h}_{ic}^t)}{\sum_{c'=1}^{C_i}\exp(\nu^\top \bar{h}_{ic'}^t)}, \qquad U_i^t[s] = \sum_{c=1}^{C_i} \beta_{ic}^t H_i^t[c,s],$$

and let $V_i^t = g_{\mathrm{band}}(x) \in \mathbb{R}^{S_i \times d}$ denote the band-aware stream. Candidate atoms are then formed over a predefined multi-scale set $\mathcal{S}$:

$$\begin{aligned} a_{im}^{(r),t} &= \mathrm{Mix}\Big(\mathrm{Pool}_r(U_i^t)_m,\ \mathrm{Pool}_r(V_i^t)_m\Big), \\ r \in \mathcal{S}, &\qquad m = 1, \dots, M_i^{(r)}, \end{aligned} \tag{1}$$

In our implementation, the filter-bank view inside g_{band} is fixed and public, and the projections that produce uploaded family content, including the band latent projection and Mix, are shared-initialized modules in the transferable backbone coordinate system. They are either frozen during summary formation or updated only through the transferable block Δ_i^t, so all uploaded family contents z_{ij}^t live in a common latent basis. Pool_r is a scale-r temporal pooling operator applied to the channel-collapsed and band-aware streams, so larger r produces coarser temporal candidate units. The sketch projection W_q is likewise stage-shared and fixed while current-round summaries are formed, so uploaded motion sketches q_{ij}^t are comparable across clients. Only the decoder g_{dec} and dataset-specific prediction heads are client-local; the former supports the structural sufficiency loss and is never used for server matching. Collecting all scales yields the candidate atom set $\mathcal{Z}_i^t = \{a_{im}^t\}_{m=1}^{M_i}$, with $M_i = \sum_{r\in\mathcal{S}} M_i^{(r)}$; these atoms remain local and are never uploaded.

The channel stream summarizes spatially organized backbone activity, the band stream retains frequency evidence, and multiscale pooling represents short transients and longer temporal context. Saliency weighting, sparse assignment, and slot maintenance organize these atoms into persistent structural summaries. Reconstruction in Eq. (5) encourages the summaries to retain the channel-collapsed trajectory. Together, these operations provide compact EEG evidence for server-side correspondence matching; Table 4 evaluates the complete sketching interface.

The encoder organizes $\mathcal{Z}_i^t$ into persistent family units by assigning atoms to the previous family prototypes. For exposition, we write the part of local memory used directly by this step as

$$\mathcal{M}_i^{t-1} = \{p_{ij}^{t-1}\}_{j=1}^{J_i^{t-1}},$$

where $p_{ij}^{t-1} \in \mathbb{R}^d$ is the previous prototype of family j. Each atom first receives a descriptor-conditioned saliency weight

$$\alpha_{im}^t = \sigma\big(w_a^\top [a_{im}^t; e_i] + b_a\big) \in [0,1],$$

and the memory-conditioned assignment weight is

$$\begin{aligned} c_{imj}^t &= \cos(a_{im}^t, p_{ij}^{t-1}), \qquad \mathcal{K}_{im}^t = \mathrm{TopK}_{\min(K_a, J_i^{t-1})}\{c_{imj}^t\}_{j=1}^{J_i^{t-1}}, \\ Z_{im}^t &= 1 + \sum_{\ell \in \mathcal{K}_{im}^t} \exp(c_{im\ell}^t), \\ \gamma_{imj}^t &= \alpha_{im}^t \, \mathbf{1}\{j \in \mathcal{K}_{im}^t\} \frac{\exp(c_{imj}^t)}{Z_{im}^t}, \qquad \zeta_{im}^t = \alpha_{im}^t \frac{1}{Z_{im}^t}. \end{aligned} \tag{2}$$

where K_a is a fixed local assignment neighborhood and the constant 1 in Z_{im}^t is a zero-logit dustbin. Thus non-neighbor prototypes receive exactly zero assignment, while ζ_{im}^t is the residual saliency assigned to the dustbin rather than to an existing family. This residual is important because the encoder assigns residual atoms to the dustbin. If $J_i^{t-1} = 0$, the assignment step is skipped, γ_{imj}^t is vacuous, and the current family bank is initialized directly from the residual atoms using the same slot-creation rule described below. When $J_i^{t-1} > 0$, summaries are formed only for the retained active index set $\mathcal{J}_{i,+}^t$, obtained after saliency weighting and the local prune/retain rule; slots with negligible effective mass are kept only for local maintenance and are not uploaded. For $j \in \mathcal{J}_{i,+}^t$ with $\mu_{ij}^t > 0$, the family summaries used in Sec. 4.2 are

$$z_{ij}^t = \frac{\sum_m \gamma_{imj}^t a_{im}^t}{\sum_m \gamma_{imj}^t}, \qquad \mu_{ij}^t = \sum_m \gamma_{imj}^t, \qquad q_{ij}^t = W_q(z_{ij}^t - p_{ij}^{t-1}). \tag{3}$$

Equation (3) defines the core family statistics on active retained families, so the denominator in the content average is positive by construction: z_{ij}^t captures family content, μ_{ij}^t captures family activity, and q_{ij}^t captures motion relative to the previous prototype. The final client sketch $\bar{q}_i^t$ is formed only after slot editing has fixed the active family set.

To ensure that family evidence controls what is written in parameter space, each family also exposes the transferable blocks it supports. Let the communicated trainable block be partitioned as $\Theta = (\Theta^{(1)}, \ldots, \Theta^{(L)})$ and $\Delta_i^t = (\Delta_i^{t,(1)}, \ldots, \Delta_i^{t,(L)})$. During local optimization, client i records the ordinary mini-batch gradient $g_{i\ell,b}^t = \nabla_{\Theta^{(\ell)}} \mathcal{L}_i^t(b)$ for each transferable block and accumulates a nonnegative family-to-block attribution

$$\tilde{\varrho}_{ij\ell}^t = \sum_{b \in \mathcal{B}_i^t} \omega_{ij,b}^t \left\| g_{i\ell,b}^t \right\|_2, \qquad \varrho_{ij\ell}^t = \frac{\tilde{\varrho}_{ij\ell}^t}{\max_{\ell'} \tilde{\varrho}_{ij\ell'}^t + \varepsilon_\varrho}. \tag{4}$$

Here b indexes local mini-batches, $\omega_{ij,b}^t$ is the normalized family mass in that mini-batch, and ε_ϱ is a numerical stabilizer. The attribution uses blockwise gradients already produced during local training and treats family weights as stop-gradient statistics, so it is a lightweight attribution proxy rather than an additional per-family backward pass. The resulting $\varrho_{ij}^t \in [0,1]^L$ is not an exact decomposition of Δ_i^t; it is a compact exposure vector indicating which transferable blocks are supported by family j.

Local family maintenance then updates retained identities, prunes low-activity slots, and creates new slots only when the residual mass indicates that the current latent is not captured by the existing bank. For retained families, the prototype update is

$$p_{ij}^t \leftarrow (1 - \eta_m) p_{ij}^{t-1} + \eta_m z_{ij}^t.$$

Consistent with Sec. 4.2, slot editing is deterministic rather than threshold-heavy. Let $S_i^t = \sum_{j=1}^{J_i^{t-1}} \mu_{ij}^t + \sum_m \zeta_{im}^t$ be the effective support before editing. The pruning and creation ratios below are evaluated only when $J_i^{t-1} > 0$ and $S_i^t > 0$. In that case, existing families with μ_{ij}^t / S_i^t below the uniform-support baseline $1/J_i^{t-1}$ are pruned, and a new slot is instantiated only when the normalized residual support $\sum_m \zeta_{im}^t / S_i^t$ exceeds the average share $1/(J_i^{t-1}+1)$ that a newly added family would receive. If $J_i^{t-1} = 0$, the first bank is initialized from the residual centroid whenever

the residual support is positive. If $S_i^t = 0$, no slot is edited or uploaded at that stage; the previous memory is retained locally and the server receives an empty family-summary set. When a new slot is created, its prototype is initialized from the residual centroid

$$\tilde{z}_i^t = \frac{\sum_m \zeta_{im}^t a_{im}^t}{\sum_m \zeta_{im}^t},$$

where the denominator is positive by the slot-creation condition. Its sketch is set to $q_{ij}^t = W_q(\tilde{z}_i^t)$, the zero-history counterpart of Eq. (3). After pruning and optional slot creation, let $\mathcal{A}_i^t$ denote the final active family set. It determines

$$\omega_{ij}^t = \frac{\mu_{ij}^t}{\sum_{\ell \in \mathcal{A}_i^t} \mu_{i\ell}^t}, \qquad \bar{q}_i^t = \sum_{j \in \mathcal{A}_i^t} \omega_{ij}^t q_{ij}^t,$$

where the normalization is taken over the final active families and is used only when $\sum_{\ell \in \mathcal{A}_i^t} \mu_{i\ell}^t > 0$. If this denominator is zero after maintenance, we set $\mathcal{U}_i^t = \emptyset$ and omit client i from the server-side family-matching graph for that stage.

The final component of evidence encoding is the structural sufficiency constraint from the main-text local objective, which keeps the extracted families information-bearing rather than purely symbolic routing tokens. Since families are constructed from the channel-collapsed stream, the decoder reconstructs $U_i^t \in \mathbb{R}^{S_i \times d}$, not the variable-channel tensor H_i^t:

$$\hat{U}_i^t = g_{\mathrm{dec}}\big(\{z_{ij}^t, \mu_{ij}^t\}_{j \in \mathcal{A}_i^t}\big), \qquad \mathcal{L}_{\mathrm{str}}^t = \big\|\hat{U}_i^t - U_i^t\big\|_2^2, \tag{5}$$

which is the appendix-level expansion of the structural term in Sec. 4.5. The decoder is client-local and uses the client's temporal grid, so it tests whether active families preserve the latent trajectory actually used for family construction without requiring a cross-montage reconstruction of raw EEG or H_i^t. With this constraint, the uploaded set

$$\mathcal{U}_i^t = \{(z_{ij}^t, q_{ij}^t, \mu_{ij}^t, \varrho_{ij}^t) : j \in \mathcal{A}_i^t\}$$

remains compact, history-aware, and directly usable by the matching and routing operators of Sec. B.2.

B.2 Server Negotiation, Routing, and Stage Readout

This subsection makes explicit the server-side negotiation mechanism that turns uploaded family summaries into consensus-supported group evidence. The server lifts pairwise candidate evidence into support groups, consensus carriers, and a final share/route decision. Given the uploaded active family summaries $\mathcal{U}_i^t = \{(z_{ij}^t, q_{ij}^t, \mu_{ij}^t, \varrho_{ij}^t) : j \in \mathcal{A}_i^t\}$, we index uploaded families as $j = 1, \dots, J_i^t$ for notational compactness. The server must decide whether locally observed family motions provide cross-client evidence for a shared structural change or only dataset-specific variation. Negotiated consolidation proceeds in two stages: it first constructs partial correspondence evidence across clients, and it then consolidates validated evidence into support groups that either update the shared backbone or remain available for routed reuse.

For each pair of active clients $i, i' \in \mathcal{C}_t$, the server compares families through content, motion, and relative activity. All cosine terms below use ℓ_2-normalized vectors with the same small numerical stabilizer used in implementation, so zero-norm sketches are mapped to the zero direction before comparison:

$$\begin{aligned}
d_z(z_{ij}^t, z_{i'\ell}^t) &= \frac{1 - \cos(z_{ij}^t, z_{i'\ell}^t)}{2}, \qquad d_q(q_{ij}^t, q_{i'\ell}^t) = \frac{1 - \cos(q_{ij}^t, q_{i'\ell}^t)}{2}, \\
d_\mu(\omega_{ij}^t, \omega_{i'\ell}^t) &= |\omega_{ij}^t - \omega_{i'\ell}^t|, \\
C_{ii'}^t(j, \ell) &= d_z(z_{ij}^t, z_{i'\ell}^t) + d_q(q_{ij}^t, q_{i'\ell}^t) + d_\mu(\omega_{ij}^t, \omega_{i'\ell}^t), \\
b_{ii'}^t &= \rho_{\mathrm{pt}} \min\{\|\boldsymbol{\omega}_i^t\|_1, \|\boldsymbol{\omega}_{i'}^t\|_1\}, \\
T_{ii'}^t &= \arg\min_{T \in \Pi_{ii'}^t} \langle C_{ii'}^t, T \rangle + \epsilon H(T), \\
\Pi_{ii'}^t &= \big\{T \geq 0 :\ T\mathbf{1} \leq \boldsymbol{\omega}_i^t,\ T^\top \mathbf{1} \leq \boldsymbol{\omega}_{i'}^t,\ \|T\|_1 = b_{ii'}^t\big\},
\end{aligned} \tag{6}$$

where $\boldsymbol{\omega}_i^t = [\omega_{i1}^t, \ldots, \omega_{iJ_i^t}^t]^\top$, $H(T) = \sum_{j,\ell} T_{j\ell}(\log T_{j\ell} - 1)$, and $0 < \rho_{\mathrm{pt}} < 1$ is a fixed partial-mass ratio. Each cost term is a raw bounded distance in $[0, 1]$, so $C_{ii'}^t(j, \ell) \in [0, 3]$ has an absolute meaning shared across client pairs. No pair-specific normalization is applied. This is important in three edge cases: a 1×1 matrix keeps its raw cost, uniformly incompatible matrices remain high-cost and fail the τ_c filter, and uniformly compatible matrices remain low-cost rather than being stretched into artificial contrast. The equality $\|T\|_1 = b_{ii'}^t$ removes the zero-plan degeneracy, while $b_{ii'}^t < \min\{\|\boldsymbol{\omega}_i^t\|_1, \|\boldsymbol{\omega}_{i'}^t\|_1\}$ keeps the coupling partial. Thus transport is candidate evidence, not a compatibility certificate by itself.

Pairwise transport is then lifted to support structure at the client level. Let $\mathcal{P}_t = \{(u, v) : u < v,\ u, v \in \mathcal{C}_t\}$ be the unordered active client-pair set. If $|\mathcal{P}_t| = 0$, no pairwise negotiation is possible and we set $\mathcal{E}_{\mathrm{neg}}^t = \emptyset$. Otherwise, we define the negotiated mass between two clients as

$$
\begin{aligned}
w_{ii'}^t &= \|T_{ii'}^t\|_1, & \bar{c}_{ii'}^t &= \frac{\langle C_{ii'}^t, T_{ii'}^t \rangle}{w_{ii'}^t + \varepsilon_{\mathrm{num}}}, \\
\widehat{w}_{ii'}^t &= w_{ii'}^t \exp(-\bar{c}_{ii'}^t), \\
\widetilde{w}_{ii'}^t &= \widehat{w}_{ii'}^t \mathbf{1}\{\widehat{w}_{ii'}^t > \tau_m\} \mathbf{1}\{\bar{c}_{ii'}^t < \tau_c\}, \\
\mathcal{E}_{\mathrm{neg}}^t &= \big\{\{i, i'\} :\ i < i',\ \widetilde{w}_{ii'}^t > 0\big\}.
\end{aligned}
$$

Here $\bar{c}_{ii'}^t$ is the transported average cost, $\widehat{w}_{ii'}^t$ is the quality-adjusted transported mass, $\varepsilon_{\mathrm{num}}$ is a numerical stabilizer, and τ_m, τ_c are global graph-filtering constants fixed before the main runs and kept unchanged across datasets, seeds, and baselines. Since $w_{ii'}^t$ uses normalized family masses and $\bar{c}_{ii'}^t$ is computed from raw bounded costs in $[0, 3]$, both filters operate on comparable scales across client pairs. The role of τ_m is to impose a quality-adjusted evidence floor, while τ_c imposes an absolute compatibility ceiling. Thus high-cost budgeted plans cannot become retained edges merely because they transport the fixed budget. Table 8 reports the global search grid and the default values used in all reported experiments.

We then form the negotiation graph $\mathcal{G}_{\mathrm{neg}}^t = (\mathcal{C}_t, \mathcal{E}_{\mathrm{neg}}^t)$. The sparse object used by BrainNEST is the retained-edge graph induced by quality-adjusted transported mass and low transported cost, rather than the entropic transport plan itself. Connected components are only candidate groups, because a single bridge client can otherwise merge incompatible endpoints. We therefore validate each candidate component before sharing. For a retained edge $e = \{i, i'\}$, define its edge coherence by

$$
\chi_e^t = \widetilde{w}_{ii'}^t.
$$

For any candidate component H, if $|H| = 1$, it is deferred to singleton routing; if $|H| = 2$, we keep it as a direct support pair. If $|H| > 2$ and the induced retained-edge subgraph contains no bridge edge, we keep H. Otherwise, we remove the weakest bridge

$$
e^\star = \arg \min_{e \in \mathrm{Bridge}(H)} \chi_e^t
$$

and recursively apply the same validation rule to the resulting connected pieces. This bridge-validation step converts candidate components into validated cross-client groups $\mathcal{G}_{\mathrm{cc}}^t$. It prevents transitive sharing in chains such as A–B–C, where A and C have no direct negotiated support, while preserving groups whose internal support is not held together by a single bridge. Clients that upload nonempty family summaries but do not belong to a validated non-singleton group are not allowed to update the shared backbone; instead, they instantiate singleton routed candidates

$$
\mathcal{G}_{\mathrm{sg}}^t = \big\{\{i\} : \mathcal{U}_i^t \neq \emptyset,\ i \text{ is not in any } G \in \mathcal{G}_{\mathrm{cc}}^t\big\}.
$$

We use $\mathcal{G}^t = \mathcal{G}_{\mathrm{cc}}^t \cup \mathcal{G}_{\mathrm{sg}}^t$ for routed-memory accounting. If $\mathcal{E}_{\mathrm{neg}}^t = \emptyset$, then $\mathcal{G}_{\mathrm{cc}}^t = \emptyset$, no consensus or shared write is performed, and all valid uploaded clients enter only through singleton routed candidates. For a validated non-singleton $G \in \mathcal{G}_{\mathrm{cc}}^t$ and $i \in G$, let

$$
s_i^G = \sum_{i' \in G:\ \{i, i'\} \in \mathcal{E}_{\mathrm{neg}}^t} \widetilde{w}_{\min(i,i'),\max(i,i')}^t
$$

Table 8: Global retained-edge filtering hyperparameters. The default values are fixed across datasets, stages, seeds, backbones, and baselines; the listed ranges indicate the robustness interval considered during method development.

Parameter	Role	Robustness interval	Default
ρ_{pt}	partial transport budget ratio	$\{0.10, 0.20, 0.30\}$	0.20
τ_m	quality-adjusted evidence floor	$\{0.02, 0.05, 0.10\}$	0.05
τ_c	transported average-cost ceiling	$\{0.75, 1.00, 1.25\}$	1.00
λ_{ev}	evidence–balance mixing	fixed	0.50

denote the negotiated evidence incident on client i. The group summaries are then

$$\begin{aligned} \beta_i^G &= \frac{s_i^G + \varepsilon_{\mathrm{num}}}{\sum_{k\in G}(s_k^G + \varepsilon_{\mathrm{num}})}, \\ \alpha_i^G &= \frac{1-\lambda_{\mathrm{ev}}}{|G|} + \lambda_{\mathrm{ev}}\beta_i^G, \\ \bar{q}_G^t &= \sum_{i\in G} \alpha_i^G \bar{q}_i^t, \qquad\qquad \bar{e}_G^t = \sum_{i\in G} \alpha_i^G e_i. \end{aligned} \tag{7}$$

Here β_i^G is evidence-based but independent of the number of raw trials, subjects, or sessions on client i, and $\lambda_{\mathrm{ev}} \in [0,1]$ controls the evidence–balance trade-off. The uniform component gives every validated client a positive share, while the evidence component still rewards stronger negotiated support. Thus a sample-rich dataset cannot dominate a shared write by volume alone, and a smaller validated dataset is not erased merely because it contributes fewer trials. For a singleton routed candidate $G = \{i\} \in \mathcal{G}_{\mathrm{sg}}^t$, we set $\alpha_i^G = 1$, $\bar{q}_G^t = \bar{q}_i^t$, and $\bar{e}_G^t = e_i$. Such candidates preserve isolated but valid specialization in $\mathcal{R}^{t+1}$, but they never refresh B^{t+1} or Θ^{t+1} because they lack cross-client support.

For each non-singleton support group $G \in \mathcal{G}_{\mathrm{cc}}^t$, we also form a candidate consensus carrier from its negotiated family evidence, but this carrier is not written into B^t until the group passes the shareability gate. For an incident retained edge $\{i, i'\}$, let $T_{i\to i'}^t = T_{ii'}^t$ when $i < i'$, and $T_{i\to i'}^t = (T_{i'i}^t)^\top$ otherwise, so rows always index families of client i. We convert transported family-level evidence into a normalized participation weight,

$$\rho_{ij}^G = \sum_{i'\in G:\, \{i,i'\}\in\mathcal{E}_{\mathrm{neg}}^t} \sum_{\ell=1}^{J_{i'}^t} T_{i\to i'}^t(j,\ell), \qquad \pi_{ij}^G = \frac{\rho_{ij}^G}{\sum_{m=1}^{J_i^t} \rho_{im}^G + \varepsilon_{\mathrm{num}}},$$

and form the group carrier

$$\tilde{b}_G^t = \mathrm{norm}\left(\sum_{i\in G} \alpha_i^G \sum_{j=1}^{J_i^t} \pi_{ij}^G z_{ij}^t \right).$$

Here π_{ij}^G measures how strongly family j participates within client i's negotiated evidence, while α_i^G controls the cross-client contribution to the group. The carrier is treated as a candidate at this point: below-threshold groups may still be reusable, but they are routed rather than used as future consensus feedback.

The same participation weights connect negotiated families to parameter writing. For singleton routed candidates, which have no cross-client transport, we set $\pi_{ij}^{\{i\}} = \omega_{ij}^t$. For any $G \in \mathcal{G}^t$, client i's group-conditioned exposure and exposed update are

$$\varrho_{i\to G}^t = \mathrm{clip}_{[0,1]}\left(\sum_{j=1}^{J_i^t} \pi_{ij}^G \varrho_{ij}^t \right), \qquad \Delta_{i\to G}^{t,(\ell)} = \varrho_{i\to G,\ell}^t \Delta_i^{t,(\ell)}. \tag{8}$$

The resulting group update is

$$\Delta_G^t = \sum_{i\in G} \alpha_i^G \Delta_{i\to G}^t. \tag{9}$$

Thus π_{ij}^G determines the family-supported exposure of each parameter block, while α_i^G controls the client's contribution to the group. The resulting group update is the common parameter object used by shared consolidation and routed memory.

After group formation, the server still has to decide whether each cross-client direction is genuinely shareable. Following Sec. 4.3, we compute the shareability score only for $G \in \mathcal{G}_{\mathrm{cc}}^t$, because singleton candidates have no cross-client evidence and are routed by construction:

$$\begin{aligned}
\bar{m}_G^t &= \frac{1}{|\mathcal{E}_{\mathrm{neg}}^t(G)|} \sum_{\{i,i'\} \in \mathcal{E}_{\mathrm{neg}}^t(G)} \widetilde{w}_{\min(i,i'),\max(i,i')}^t, \\
\sigma_G^t &= \sum_{i \in G} \alpha_i^G \|\bar{q}_i^t - \bar{q}_G^t\|_2^2, \\
\tilde{\sigma}_G^t &= \frac{\sigma_G^t}{\frac{1}{|\mathcal{G}_{\mathrm{cc}}^t|} \sum_{H \in \mathcal{G}_{\mathrm{cc}}^t} \sigma_H^t} \quad \text{with } \tilde{\sigma}_G^t = 0 \text{ if the denominator is zero,} \\
a_G^t &= \bar{m}_G^t \cdot \frac{|\mathcal{E}_G^t|}{|\mathcal{E}^t|} \cdot \exp\!\left(-\tilde{\sigma}_G^t\right),
\end{aligned} \tag{10}$$

where $\mathcal{E}_{\mathrm{neg}}^t(G) \subseteq \mathcal{E}_{\mathrm{neg}}^t$ is the set of retained negotiation edges internal to G. The coverage sets $\mathcal{E}_G^t$ and $\mathcal{E}^t$ count active client identifiers rather than task labels or descriptor values; because each active client is one dataset, $|\mathcal{E}_G^t| = |G|$ and $|\mathcal{E}^t| = |\mathcal{C}_t|$ in our benchmark. The score is conservative: a direction is shared only when negotiated support is strong, client coverage is nontrivial, and group sketches remain concentrated.

The subset of cross-client groups satisfying $a_G^t \geq \tau_{\mathrm{sh}}$ is denoted by $\mathcal{G}_{\mathrm{sh}}^t = \{G \in \mathcal{G}_{\mathrm{cc}}^t : a_G^t \geq \tau_{\mathrm{sh}}\}$. Only those groups are allowed to modify the shared backbone:

$$\Theta^{t+1} = \Theta^t + \frac{\sum_{G \in \mathcal{G}_{\mathrm{sh}}^t} a_G^t \Delta_G^t}{\sum_{G \in \mathcal{G}_{\mathrm{sh}}^t} a_G^t}. \tag{11}$$

If $\mathcal{G}_{\mathrm{sh}}^t = \emptyset$, we set $\Theta^{t+1} = \Theta^t$. Consensus memory is refreshed by the same shared groups and by no routed groups. For each $G \in \mathcal{G}_{\mathrm{sh}}^t$ with $B^t \neq \emptyset$, the candidate carrier $\tilde{b}_G^t$ is merged into the nearest consensus entry through similarity-based momentum updating:

$$m_G^\star = \arg\max_m \cos(\tilde{b}_G^t, b_m^t), \qquad b_{m_G^\star}^{t+1} \leftarrow \mathrm{norm}\big((1-\eta_m) b_{m_G^\star}^t + \eta_m \tilde{b}_G^t\big). \tag{12}$$

The stage-average candidate–bank compatibility is the mean similarity between current shared candidates and existing consensus entries,

$$\bar{s}_B^t = \frac{1}{|\mathcal{G}_{\mathrm{sh}}^t|\,|B^t|} \sum_{G \in \mathcal{G}_{\mathrm{sh}}^t} \sum_{m=1}^{|B^t|} \cos(\tilde{b}_G^t, b_m^t),$$

whenever both sets are nonempty. If $B^t = \emptyset$, or if the nearest entry falls below $\bar{s}_B^t$, $\tilde{b}_G^t$ is appended as a new slot instead. If $\mathcal{G}_{\mathrm{sh}}^t = \emptyset$, then $B^{t+1} = B^t$. Groups below the sharing threshold are routed rather than discarded. This closes the server-side decision map: transport builds support groups, B^{t+1} records only shareability-approved consensus carriers, and coherent below-threshold group directions are reserved for routed memory.

Routed memory update and stage readout. This subsection makes explicit the routed-memory mechanism that realizes the second transfer path of BrainNEST. Whereas B^t stores consensus carriers for feedback alignment, $\mathcal{R}^t = \{(\Xi_r^t, \kappa_r^t, \tau_r^t)\}_{r=1}^{K_t}$ stores reusable parameter offsets together with a sketch key and a non-semantic acquisition anchor.

Let $\mathcal{G}^t = \mathcal{G}_{\mathrm{cc}}^t \cup \mathcal{G}_{\mathrm{sg}}^t$ denote the cross-client support groups and singleton routed candidates from Sec. B.2, and define the routed subset by $\mathcal{G}_{\mathrm{rt}}^t = \mathcal{G}^t \setminus \mathcal{G}_{\mathrm{sh}}^t$. For each $G \in \mathcal{G}_{\mathrm{rt}}^t$, the server forms the routed write candidate

$$\Xi_G^t = \Delta_G^t, \qquad \kappa_G^t = \bar{q}_G^t, \qquad \tau_G^t = \bar{e}_G^t, \tag{13}$$

where Ξ_G^t is the family-exposed reusable parameter direction, κ_G^t the group sketch used for motion-based addressing, and τ_G^t the associated acquisition anchor. This is the routed counterpart of Eq. (11): the available group evidence defines both the candidate direction and its destination.

To avoid treating every stage-specific variation as a new memory slot, the server queries the current routed bank with the same two-factor compatibility used in the main text whenever $\mathcal{R}^t \neq \emptyset$:

$$u_G^t(r) = \cos(\kappa_G^t, \kappa_r^t) + \cos(\tau_G^t, \tau_r^t), \qquad r_G^\star = \arg\max_r u_G^t(r). \tag{14}$$

When both the routed candidate set and $\mathcal{R}^t$ are nonempty, the routed candidate–bank compatibility is

$$\bar{s}_R^t = \frac{1}{|\mathcal{G}_{\mathrm{rt}}^t|\,|\mathcal{R}^t|} \sum_{G \in \mathcal{G}_{\mathrm{rt}}^t} \sum_{r=1}^{|\mathcal{R}^t|} u_G^t(r),$$

computed over the routed candidates of the current stage and the existing routed bank. If $\mathcal{R}^t = \emptyset$, or if the best routed score falls below $\bar{s}_R^t$, a new slot is appended. Otherwise, the matched entry is updated by the same fixed EMA rule:

$$\begin{aligned} \Xi_{r_G^\star}^{t+1} &\leftarrow (1-\eta_m)\Xi_{r_G^\star}^t + \eta_m \Xi_G^t, \\ \kappa_{r_G^\star}^{t+1} &\leftarrow (1-\eta_m)\kappa_{r_G^\star}^t + \eta_m \kappa_G^t, \\ \tau_{r_G^\star}^{t+1} &\leftarrow (1-\eta_m)\tau_{r_G^\star}^t + \eta_m \tau_G^t. \end{aligned} \tag{15}$$

In the append case, the new entry is initialized directly by $(\Xi_G^t, \kappa_G^t, \tau_G^t)$. The same operator is applied to every $G \in \mathcal{G}_{\mathrm{rt}}^t$, so a direction that is too narrow to enter Θ^{t+1} is still preserved as addressable specialization.

If a hard routed-bank budget $K_{\max}$ is imposed, budget maintenance is an explicit step applied after all routed writes of the stage. When $\mathcal{G}_{\mathrm{rt}}^t \neq \emptyset$, entries are ranked by their best current-stage compatibility $\max_{G \in \mathcal{G}_{\mathrm{rt}}^t} u_G^t(r)$, with ties broken in favor of the most recently updated entry, and the lowest-priority entries are removed until $|\mathcal{R}^{t+1}| \leq K_{\max}$. Setting $K_{\max} = \infty$ recovers the unbudgeted bank. The retention property in Appendix C.5 is stated for routed slots that remain after this explicit budget-maintenance step; it is not a claim of infinite storage under arbitrary eviction.

Once the routed writes from stage t are committed, the server forms the stage-$(t+1)$ initialization. The next active set is partitioned into continuing and newly joining clients:

$$\mathcal{C}_{t+1}^{\mathrm{stay}} = \mathcal{C}_t \cap \mathcal{C}_{t+1}, \qquad \mathcal{C}_{t+1}^{\mathrm{new}} = \mathcal{C}_{t+1} \setminus \mathcal{C}_t.$$

The sketch reference used for retrieval is then

$$\hat{q}_i^t = \begin{cases} \bar{q}_i^t, & i \in \mathcal{C}_{t+1}^{\mathrm{stay}}, \\ \emptyset, & i \in \mathcal{C}_{t+1}^{\mathrm{new}}. \end{cases}$$

The routed readout is carried out in three steps. We first score each routed entry by

$$s_i^{t+1}(r) = \begin{cases} \cos(\hat{q}_i^t, \kappa_r^{t+1}) + \cos(e_i, \tau_r^{t+1}), & \hat{q}_i^t \neq \emptyset, \\ \cos(e_i, \tau_r^{t+1}), & \hat{q}_i^t = \emptyset, \end{cases} \tag{16}$$

then select and normalize

$$A_i^{t+1} = \mathrm{TopK}_r\big(s_i^{t+1}(r)\big), \qquad \alpha_{ir}^{t+1} = \frac{\exp(s_i^{t+1}(r))}{\sum_{r' \in A_i^{t+1}} \exp(s_i^{t+1}(r'))}, \tag{17}$$

and finally form the personalized warm start

$$\widetilde{\Theta}_i^{t+1} = \Theta^{t+1} + \lambda_{\mathrm{mem}} \sum_{r \in A_i^{t+1}} \alpha_{ir}^{t+1} \Xi_r^{t+1}. \tag{18}$$

If $\mathcal{R}^{t+1} = \emptyset$, we set $A_i^{t+1} = \emptyset$ and $\widetilde{\Theta}_i^{t+1} = \Theta^{t+1}$. Continuing clients use both sketch and acquisition information, whereas newly joining clients use acquisition-aware retrieval until their own sketches are formed by local training. If fewer than K_r routed entries are available, TopK_r returns all available entries.

This write–read mechanism completes the second transfer path of BrainNEST. B^{t+1} regularizes the next local representation update, while $\mathcal{R}^{t+1}$ restores reusable specialization through a personalized warm start. Together with evidence encoding and negotiated consolidation, it defines the closed stage recursion used in the main text.

B.3 Local Objective and Closed Stage Recursion

This subsection completes the read side of BrainNEST. At the beginning of stage t, client i receives the personalized backbone $\widetilde{\Theta}_i^t$ produced by the readout rule in Eq. (18) with indices shifted from $t-1$ to t, together with the consensus bank B^t. It then trains only on its local EEG data and local prediction head, while the transferable block is regularized by the family summaries constructed in Sec. B.1. The local objective used in the main text is

$$\mathcal{L}_i^t = \mathcal{L}_{\text{task}}^i + \lambda_{\text{aux}} \left(\mathcal{L}_{\text{str}}^i + \chi_B^t \mathcal{L}_{\text{fb}}^i \right), \tag{19}$$

where $\mathcal{L}_{\text{task}}^i$ is the dataset-specific supervised loss on the local head, λ_{aux} is shared by both auxiliary terms, and $\chi_B^t = \mathbf{1}\{B^t \neq \emptyset\}$. The structural term is the sufficiency loss already expanded in Eq. (5). The feedback term is

$$\mathcal{L}_{\text{fb}}^i = \sum_{j \in \mathcal{A}_i^t} \omega_{ij}^t \Big(1 - \cos\big(z_{ij}^t, r(z_{ij}^t; B^t)\big) \Big). \tag{20}$$

Here the sum is omitted when $\mathcal{A}_i^t = \emptyset$, and the term is inactive when B^t is empty. Thus the first stage does not force alignment to a non-existent consensus, while later stages can use consolidated cross-client structure as a soft reference.

The consensus reference in Eq. (20) is a normalized attention read from B^t:

$$r(z; B^t) = \text{norm} \left(\sum_{m=1}^{|B^t|} \frac{\exp(z^\top b_m^t)}{\sum_{\ell=1}^{|B^t|} \exp(z^\top b_\ell^t)} b_m^t \right). \tag{21}$$

Here b_m^t is a consensus carrier written only by shareability-approved groups in Sec. B.2. This feedback does not change the dataset-specific label space or prediction head; it only encourages the family summaries extracted from the transferable backbone to remain compatible with previously negotiated EEG structure.

Combining the pieces above yields the closed stage transition. Let the server-side state be

$$\Phi^t = (\Theta^t, B^t, \mathcal{R}^t), \qquad \mathcal{R}^t = \{(\Xi_r^t, \kappa_r^t, \tau_r^t)\}_{r=1}^{K_t}. \tag{22}$$

Given Φ^t, the server first reads $\mathcal{R}^t$ to produce $\widetilde{\Theta}_i^t$ for each active client. The client then encodes EEG evidence, optimizes Eq. (19), updates its local family memory, and uploads $m_{i,\uparrow}^t = (\Delta_i^t, \mathcal{U}_i^t)$. The server applies negotiated consolidation to obtain support groups, writes shareable groups into (Θ^{t+1}, B^{t+1}), and writes below-threshold groups into $\mathcal{R}^{t+1}$. Therefore the next state

$$\Phi^{t+1} = (\Theta^{t+1}, B^{t+1}, \mathcal{R}^{t+1})$$

contains exactly the two forms of knowledge used in the next stage: consensus structure for local alignment and routed specialization for personalized initialization. No raw EEG, previous-stage samples, or dataset-specific heads are stored on the server.

C Operator-Level Properties

The following results characterize BrainNEST's matching, balancing, exposure, gating, and retention operators for the bridge-validated support groups in Sec. B.2. They establish a fixed partial matching budget, client-balanced weight bounds, non-expansive blockwise exposure, an explicit evidence–coherence threshold, and positive finite-ancestry weights in retained memory slots. These statements concern the update operators; adaptation and forgetting are evaluated empirically in Sec. 5.

C.1 Partial Transport Avoids Forced Misalignment

The transport step in Eq. (6) allocates a positive partial mass budget. Candidate correspondences subsequently pass quality-adjusted edge filters and support-group validation before entering a shared write.

Fix an active client pair (i, i') at stage t, and abbreviate $C = C_{ii'}^t$, $\mu = \boldsymbol{\omega}_i^t$, $\nu = \boldsymbol{\omega}_{i'}^t$, $b = b_{ii'}^t$, and $\Pi_b = \Pi_{ii'}^t$. Let

$$\Pi_{\leq} = \{T \geq 0 : \; T\mathbf{1} \leq \mu, \; T^\top \mathbf{1} \leq \nu\}$$

be the upper-marginal partial-coupling polytope before the budget constraint, and let $M_{\text{full}} = \min\{\|\mu\|_1, \|\nu\|_1\}$. Because $b = \rho_{\text{pt}} M_{\text{full}}$ with fixed $0 < \rho_{\text{pt}} < 1$, the zero matrix is infeasible whenever both clients upload nonempty family summaries, while $b < M_{\text{full}}$ leaves a non-transported remainder by construction.

Let $\mathcal{K}^t_{ii'} \subseteq [J^t_i] \times [J^t_{i'}]$ denote an analysis-only set of genuinely compatible family pairs under the bounded family cost C. For any plan T, define $\|T_{\mathcal{K}}\|_1 = \sum_{(j,\ell)\in\mathcal{K}^t_{ii'}} T_{j\ell}$ and $\|T_{\mathcal{K}^c}\|_1 = \sum_{(j,\ell)\notin\mathcal{K}^t_{ii'}} T_{j\ell}$. The largest mass transportable using only compatible pairs is

$$M_{\text{good}} = \max_{\substack{T\in\Pi_{\le} \\ \text{supp}(T)\subseteq\mathcal{K}^t_{ii'}}} \|T\|_1. \tag{23}$$

The discrepancy

$$\delta^t_{ii'} = M_{\text{full}} - M_{\text{good}} \ge 0 \tag{24}$$

is the amount of mass that a full matching must place outside the compatible set when $M_{\text{good}} < M_{\text{full}}$.

Lemma C.1 (compulsory leakage under full matching). Any $\bar{T} \in \Pi_{\le}$ satisfying $\|\bar{T}\|_1 = M_{\text{full}}$ obeys

$$\|\bar{T}_{\mathcal{K}^c}\|_1 \ge \delta^t_{ii'}. \tag{25}$$

Proof. If $\|\bar{T}_{\mathcal{K}^c}\|_1 < \delta^t_{ii'}$, then $\|\bar{T}_{\mathcal{K}}\|_1 > M_{\text{full}} - \delta^t_{ii'} = M_{\text{good}}$. Zeroing the entries of $\bar{T}$ outside $\mathcal{K}^t_{ii'}$ preserves the upper marginal constraints, producing a feasible plan in $\Pi_{\le}$ supported on $\mathcal{K}^t_{ii'}$ with mass larger than M_{good}, contradicting Eq. (23).

Proposition C.1 (budgeted transport and retained-edge filtering). If $b \le M_{\text{good}}$, then there exists a feasible budgeted plan $T^{\mathcal{K}} \in \Pi_b$ with $\text{supp}(T^{\mathcal{K}}) \subseteq \mathcal{K}^t_{ii'}$. In contrast, every full plan of mass M_{full} leaks at least $\delta^t_{ii'}$ mass outside $\mathcal{K}^t_{ii'}$. Moreover, suppose every incompatible pair has cost at least $c_{\text{out}} > 0$. Any retained budgeted plan $T \in \Pi_b$ with transported average cost $\bar{c}(T) = \langle C, T\rangle / b < \tau_c$ satisfies

$$\frac{\|T_{\mathcal{K}^c}\|_1}{b} \le \frac{\tau_c}{c_{\text{out}}}. \tag{26}$$

If all candidate pairs have cost at least $c_{\text{bad}} > \tau_c$, no retained edge can be formed, including the degenerate 1×1 case. This relies on the raw bounded cost scale in Sec. B.2; pair-specific normalization would destroy this absolute rejection property.

Proof. Let S attain the maximum in Eq. (23). If $b \le M_{\text{good}}$, then $T^{\mathcal{K}} = (b/M_{\text{good}})S$ belongs to Π_b and remains supported on $\mathcal{K}^t_{ii'}$. The full-plan leakage statement is Lemma C.1. For the retained-edge bound, nonnegativity of C and the lower bound $C(j,\ell) \ge c_{\text{out}}$ on $\mathcal{K}^c$ give

$$\langle C, T\rangle \ge c_{\text{out}} \|T_{\mathcal{K}^c}\|_1.$$

Since retained edges satisfy $\langle C, T\rangle / b < \tau_c$, Eq. (26) follows. If all entries have cost at least $c_{\text{bad}} > \tau_c$, then every feasible budgeted plan has $\bar{c}(T) \ge c_{\text{bad}} > \tau_c$ and is rejected by the retained-edge rule.

This result is intentionally an operator-level sanity property. The entropic minimizer may still allocate some budget to imperfect pairs when $\epsilon > 0$, and the analysis-only set $\mathcal{K}^t_{ii'}$ is not observed by the algorithm. The additional filter $\widehat{w}^t_{ii'} > \tau_m$ in Sec. B.2 is complementary to the cost bound: it rejects budgeted plans whose transported mass becomes weak after the cost penalty, even when the fixed budget is feasible. The operational property is therefore narrow and reproducible: BrainNEST compares a nonzero but partial amount of candidate mass, never treats the budgeted plan alone as a compatibility certificate, and allows only quality-adjusted, low-cost, bridge-validated evidence to enter the shareability gate.

C.2 Shareability Gate Has an Evidence–Coherence Threshold

This subsection shows that the shareability score from Sec. B.2 has an exact threshold form, monotone dependence on its components, and a local margin-stability property. The analysis is carried out only for cross-client support groups $G \in \mathcal{G}^t_{\text{cc}}$; singleton candidates are routed by construction and are not evaluated by the shareability gate.

For brevity, define the client-coverage ratio

$$\rho_G^t := \frac{|\mathcal{E}_G^t|}{|\mathcal{E}^t|} \in (0, 1].$$

Then Eq. (10) can be written as

$$a_G^t = \bar{m}_G^t \, \rho_G^t \, \exp(-\tilde{\sigma}_G^t), \qquad \tilde{\sigma}_G^t \geq 0. \tag{27}$$

Here $\tilde{\sigma}_G^t$ is the stage-normalized dispersion defined in Sec. B.2. For every cross-client support group from Sec. B.2, $\rho_G^t > 0$, $\bar{m}_G^t > 0$, and $\tau_{\text{sh}} > 0$.

Lemma C.2 (shareability monotonicity). Fix a stage t and a support group G. Over the domain $\bar{m}_G^t > 0,\ \rho_G^t > 0,\ \tilde{\sigma}_G^t \geq 0$, the score a_G^t is strictly increasing in $\bar{m}_G^t$, strictly increasing in ρ_G^t, and strictly decreasing in $\tilde{\sigma}_G^t$. Equivalently,

$$\log a_G^t = \log \bar{m}_G^t + \log \rho_G^t - \tilde{\sigma}_G^t. \tag{28}$$

Hence, for any two support groups G and H,

$$a_G^t \geq a_H^t \iff \log \frac{\bar{m}_G^t}{\bar{m}_H^t} + \log \frac{\rho_G^t}{\rho_H^t} \geq \tilde{\sigma}_G^t - \tilde{\sigma}_H^t. \tag{29}$$

Proof. From Eq. (27),

$$\frac{\partial a_G^t}{\partial \bar{m}_G^t} = \rho_G^t \exp(-\tilde{\sigma}_G^t) > 0, \qquad \frac{\partial a_G^t}{\partial \rho_G^t} = \bar{m}_G^t \exp(-\tilde{\sigma}_G^t) > 0,$$

and

$$\frac{\partial a_G^t}{\partial \tilde{\sigma}_G^t} = -a_G^t < 0.$$

This proves strict monotonicity. Taking logarithms of Eq. (27) yields Eq. (28). Subtracting the corresponding identities for G and H gives

$$\log a_G^t - \log a_H^t = \log \frac{\bar{m}_G^t}{\bar{m}_H^t} + \log \frac{\rho_G^t}{\rho_H^t} - \big(\tilde{\sigma}_G^t - \tilde{\sigma}_H^t\big),$$

and Eq. (29) follows because $\log(\cdot)$ is strictly increasing.

Proposition C.2 (threshold and margin form of the gate). Let

$$\beta_G^t := \bar{m}_G^t \rho_G^t. \tag{30}$$

Then the shareability condition $a_G^t \geq \tau_{\text{sh}}$ admits the following exact characterization:

$$a_G^t \geq \tau_{\text{sh}} \iff \beta_G^t \geq \tau_{\text{sh}} \quad \text{and} \quad \tilde{\sigma}_G^t \leq \log\big(\beta_G^t / \tau_{\text{sh}}\big). \tag{31}$$

Equivalently, if $\beta_G^t < \tau_{\text{sh}}$, then no nonnegative dispersion can make G shareable; if $\beta_G^t = \tau_{\text{sh}}$, then G is shareable if and only if $\tilde{\sigma}_G^t = 0$; and if $\beta_G^t > \tau_{\text{sh}}$, then

$$\tilde{\sigma}_G^t \leq \log\big(\beta_G^t / \tau_{\text{sh}}\big) \tag{32}$$

is the exact coherence budget allowed by the threshold.

Furthermore, let $(\widetilde{m}_G^t, \widetilde{\rho}_G^t, \widehat{\sigma}_G^t)$ be perturbed score components with

$$\left|\log \frac{\widetilde{m}_G^t}{\bar{m}_G^t}\right| \leq \varepsilon_m, \qquad \left|\log \frac{\widetilde{\rho}_G^t}{\rho_G^t}\right| \leq \varepsilon_\rho, \qquad |\widehat{\sigma}_G^t - \tilde{\sigma}_G^t| \leq \varepsilon_\sigma. \tag{33}$$

Define the perturbed score $\widetilde{a}_G^t = \widetilde{m}_G^t \, \widetilde{\rho}_G^t \exp(-\widehat{\sigma}_G^t)$. Then

$$\big|\log \widetilde{a}_G^t - \log a_G^t\big| \leq \varepsilon_m + \varepsilon_\rho + \varepsilon_\sigma. \tag{34}$$

Consequently, if

$$\big|\log a_G^t - \log \tau_{\text{sh}}\big| > \varepsilon_m + \varepsilon_\rho + \varepsilon_\sigma, \tag{35}$$

then the decision induced by the threshold is unchanged under the perturbation, i.e., $\mathbf{1}\{\widetilde{a}_G^t \geq \tau_{\text{sh}}\} = \mathbf{1}\{a_G^t \geq \tau_{\text{sh}}\}$.

Proof. Starting from Eq. (27),

$$a_G^t \geq \tau_{\mathrm{sh}} \iff \beta_G^t \exp(-\tilde{\sigma}_G^t) \geq \tau_{\mathrm{sh}}.$$

If $\beta_G^t < \tau_{\mathrm{sh}}$, then using $\tilde{\sigma}_G^t \geq 0$,

$$a_G^t = \beta_G^t \exp(-\tilde{\sigma}_G^t) \leq \beta_G^t < \tau_{\mathrm{sh}},$$

so the shareability event is impossible. If $\beta_G^t \geq \tau_{\mathrm{sh}}$, taking logarithms yields

$$\begin{aligned} a_G^t \geq \tau_{\mathrm{sh}} &\iff \log \beta_G^t - \tilde{\sigma}_G^t \geq \log \tau_{\mathrm{sh}} \\ &\iff \tilde{\sigma}_G^t \leq \log\big(\beta_G^t/\tau_{\mathrm{sh}}\big), \end{aligned}$$

which proves Eq. (31). When $\beta_G^t = \tau_{\mathrm{sh}}$, the right-hand side is zero, so the shareability event reduces exactly to $\tilde{\sigma}_G^t = 0$. When $\beta_G^t > \tau_{\mathrm{sh}}$, the same inequality yields the coherence-budget interpretation in Eq. (32).

For the perturbation bound, write

$$\log \widetilde{a}_G^t - \log a_G^t = \log \frac{\widetilde{m}_G^t}{\bar{m}_G^t} + \log \frac{\widetilde{\rho}_G^t}{\rho_G^t} - \big(\widehat{\sigma}_G^t - \tilde{\sigma}_G^t\big).$$

Taking absolute values and using Eq. (33) gives Eq. (34). Finally, if Eq. (35) holds, then $\log a_G^t$ lies at a distance strictly larger than the perturbation radius from the threshold $\log \tau_{\mathrm{sh}}$. Therefore $\log \widetilde{a}_G^t$ cannot cross $\log \tau_{\mathrm{sh}}$, and the binary decision is unchanged.

The gate combines the evidence term $\beta_G^t = \bar{m}_G^t \rho_G^t$ with the coherence penalty $\exp(-\tilde{\sigma}_G^t)$. Proposition C.2 expresses the sharing decision as an evidence floor $\beta_G^t \geq \tau_{\mathrm{sh}}$ and a conditional coherence budget from Eq. (32).

C.3 Client-Balanced Writing Limits Scale-Dominant Aggregation

The client-balanced weights in Eq. (7) combine a uniform contribution with normalized structural evidence. The following result bounds each validated client's coefficient in the group update.

For a validated non-singleton group G, recall

$$\beta_i^G = \frac{s_i^G + \varepsilon_{\mathrm{num}}}{\sum_{k \in G}(s_k^G + \varepsilon_{\mathrm{num}})}, \qquad \alpha_i^G = \frac{1 - \lambda_{\mathrm{ev}}}{|G|} + \lambda_{\mathrm{ev}} \beta_i^G, \tag{36}$$

where $s_i^G \geq 0$ is the negotiated edge evidence incident on client i, $\varepsilon_{\mathrm{num}} > 0$, and $\lambda_{\mathrm{ev}} \in [0, 1]$. The quantity s_i^G is computed from retained family evidence, not from the number of trials, subjects, or sessions.

Proposition C.3 (client-balanced dominance bound). For every $i \in G$,

$$\frac{1 - \lambda_{\mathrm{ev}}}{|G|} \leq \alpha_i^G \leq \frac{1 - \lambda_{\mathrm{ev}}}{|G|} + \lambda_{\mathrm{ev}}, \qquad \sum_{i \in G} \alpha_i^G = 1. \tag{37}$$

Consequently, for any block-level group write $\Delta_G^{t,(\ell)} = \sum_{i \in G} \alpha_i^G \Delta_{i \to G}^{t,(\ell)}$, no validated client can receive zero weight when $\lambda_{\mathrm{ev}} < 1$, and no single client can receive more than $(1 - \lambda_{\mathrm{ev}})/|G| + \lambda_{\mathrm{ev}}$ of the group coefficient.

Proof. Since β_i^G is a normalized nonnegative evidence weight, $0 \leq \beta_i^G \leq 1$ and $\sum_{i \in G} \beta_i^G = 1$. Substituting these two facts into Eq. (36) immediately gives the lower bound, upper bound, and normalization in Eq. (37). The block-write statement follows by applying the same coefficients to $\Delta_{i \to G}^{t,(\ell)}$.

Client influence is bounded by group size and the evidence-mixing coefficient. With the default $\lambda_{\mathrm{ev}} = 0.5$, each validated client receives at least $1/(2|G|)$ of the group coefficient, and a maximally supported client receives at most $1/2 + 1/(2|G|)$.

C.4 Family-Conditioned Exposure Is Non-Expansive

The exposure vector $\varrho^t_{i\to G}$ maps family participation to parameter-block attribution. Its entries lie in $[0,1]$, so the resulting operator attenuates transferable blocks before shared consolidation or routed-memory writing.

For every group G, client $i \in G$, and transferable block ℓ, Sec. B.2 defines

$$\Delta^{t,(\ell)}_{i\to G} = \varrho^t_{i\to G,\ell}\Delta^{t,(\ell)}_i, \qquad \varrho^t_{i\to G,\ell} \in [0,1]. \tag{38}$$

The group update is $\Delta^t_G = \sum_{i\in G} \alpha^G_i \Delta^t_{i\to G}$, with $\alpha^G_i \geq 0$ and $\sum_i \alpha^G_i = 1$.

Proposition C.4 (non-expansive family exposure). For every $i \in G$ and block ℓ,

$$\left\|\Delta^{t,(\ell)}_{i\to G}\right\|_2 \leq \left\|\Delta^{t,(\ell)}_i\right\|_2. \tag{39}$$

Moreover,

$$\left\|\Delta^{t,(\ell)}_G\right\|_2 \leq \sum_{i\in G} \alpha^G_i \left\|\Delta^{t,(\ell)}_i\right\|_2. \tag{40}$$

If $\varrho^t_{i\to G,\ell} = 0$, then client i's block ℓ contributes no mass to the group write; if $\varrho^t_{i\to G,\ell} = 0$ for all $i \in G$, then block ℓ is absent from both the shared and routed write induced by G.

Proof. Equation (38) gives

$$\left\|\Delta^{t,(\ell)}_{i\to G}\right\|_2 = \varrho^t_{i\to G,\ell}\left\|\Delta^{t,(\ell)}_i\right\|_2,$$

and Eq. (39) follows because $\varrho^t_{i\to G,\ell} \in [0,1]$. For the group update, the triangle inequality and $\alpha^G_i \geq 0$ give

$$\left\|\Delta^{t,(\ell)}_G\right\|_2 \leq \sum_{i\in G} \alpha^G_i \left\|\Delta^{t,(\ell)}_{i\to G}\right\|_2 \leq \sum_{i\in G} \alpha^G_i \left\|\Delta^{t,(\ell)}_i\right\|_2.$$

The zero-exposure statements follow directly from Eq. (38).

Family participation gates transferable blocks before group aggregation. The exposure operator attenuates each block, and the nonnegative client weights bound its contribution to the final group update.

C.5 Routed Specialization Has Positive Finite-Ancestry Retention

Non-shareable groups enter routed memory and contribute to subsequent personalized initialization when their slots are retrieved. The results below characterize this retention and conditional reuse. With a finite bank budget $K_{\max}$, they apply to slots retained after the maintenance step in Sec. B.2.

For notational symmetry, define for every support group $G \in \mathcal{G}^t$

$$\Xi^t_G := \Delta^t_G.$$

When $G \in \mathcal{G}^t_{\mathrm{rt}}$, this is exactly the routed write candidate already defined in Eq. (13); when $G \in \mathcal{G}^t_{\mathrm{sh}}$, it is the family-exposed group direction appearing in Eq. (11). Throughout this subsection, we focus on the persistent-memory regime $\eta_m \in (0,1)$. This is the regime in which routed memory accumulates reusable specialization over time; the endpoint $\eta_m = 1$ reduces Eq. (15) to hard replacement, which is compatible with the method but does not realize the non-discard accumulation property analyzed here.

Lemma C.5 (finite-ancestry expansion of routed slots). Fix a routed slot r. Consider the totally ordered sequence of routed write events that generate its current content, and denote that ancestry by $(s_1, G_1), \ldots, (s_m, G_m)$, where s_ℓ is the stage index of the ℓ-th write event, so $s_1 \leq \cdots \leq s_m$; ties are allowed because several routed groups from the same stage may be applied sequentially to the same slot. Assume that the first event appends slot r from G_1, and each later event updates the same

slot through Eq. (15). Let $(\Xi_r^{[\ell]}, \kappa_r^{[\ell]}, \tau_r^{[\ell]})$ denote the slot state immediately after the ℓ-th write event in this ancestry. Then

$$\begin{aligned}
\Xi_r^{[m]} &= (1-\eta_m)^{m-1}\Xi_{G_1}^{s_1} + \sum_{\ell=2}^{m} \eta_m (1-\eta_m)^{m-\ell}\Xi_{G_\ell}^{s_\ell}, \\
\kappa_r^{[m]} &= (1-\eta_m)^{m-1}\kappa_{G_1}^{s_1} + \sum_{\ell=2}^{m} \eta_m (1-\eta_m)^{m-\ell}\kappa_{G_\ell}^{s_\ell}, \\
\tau_r^{[m]} &= (1-\eta_m)^{m-1}\tau_{G_1}^{s_1} + \sum_{\ell=2}^{m} \eta_m (1-\eta_m)^{m-\ell}\tau_{G_\ell}^{s_\ell}.
\end{aligned} \tag{41}$$

In particular, at any finite ancestry depth, every routed group that has written into slot r retains a positive coefficient in the resulting slot state.

Proof. We prove the statement for Ξ_r; the arguments for κ_r and τ_r are identical. At the first write event, the append rule in Sec. B.2 gives

$$\Xi_r^{[1]} = \Xi_{G_1}^{s_1}.$$

For $m = 1$, Eq. (41) is therefore immediate. Assume now that the expansion holds after the $(m-1)$-st write, i.e.,

$$\Xi_r^{[m-1]} = (1-\eta_m)^{m-2}\Xi_{G_1}^{s_1} + \sum_{\ell=2}^{m-1} \eta_m (1-\eta_m)^{m-1-\ell}\Xi_{G_\ell}^{s_\ell}.$$

The m-th routed write applies Eq. (15), so

$$\begin{aligned}
\Xi_r^{[m]} &= (1-\eta_m)\Xi_r^{[m-1]} + \eta_m \Xi_{G_m}^{s_m} \\
&= (1-\eta_m)\left[(1-\eta_m)^{m-2}\Xi_{G_1}^{s_1} + \sum_{\ell=2}^{m-1} \eta_m (1-\eta_m)^{m-1-\ell}\Xi_{G_\ell}^{s_\ell}\right] + \eta_m \Xi_{G_m}^{s_m} \\
&= (1-\eta_m)^{m-1}\Xi_{G_1}^{s_1} + \sum_{\ell=2}^{m-1} \eta_m (1-\eta_m)^{m-\ell}\Xi_{G_\ell}^{s_\ell} + \eta_m \Xi_{G_m}^{s_m},
\end{aligned}$$

which is exactly the first line of Eq. (41). Since $\eta_m \in (0,1)$, all coefficients are nonnegative; moreover,

$$(1-\eta_m)^{m-1} + \sum_{\ell=2}^{m} \eta_m (1-\eta_m)^{m-\ell} = 1,$$

so the slot state is a convex combination of the routed groups that have written into it. Because $\eta_m \in (0,1)$, every finite-ancestry coefficient in Eq. (41) is positive, although older coefficients may decay as more writes hit the same slot. Thus no routed write is immediately erased from the slot ancestry once it has been absorbed.

Proposition C.5 (non-discard and conditional readout). Fix a stage t. Then every support group $G \in \mathcal{G}^t$ enters at least one persistent transfer path in the stage transition from t to $t+1$, up to the explicit routed-bank budget rule above:

$$\mathcal{G}^t = \mathcal{G}_{\mathrm{sh}}^t \sqcup \mathcal{G}_{\mathrm{rt}}^t. \tag{42}$$

More precisely:

1. If $G \in \mathcal{G}_{\mathrm{sh}}^t$, then G contributes to the shared backbone update with a positive coefficient. In particular, when $\mathcal{G}_{\mathrm{sh}}^t \neq \emptyset$,

$$\Theta^{t+1} = \Theta^t + \sum_{H \in \mathcal{G}_{\mathrm{sh}}^t} \omega_H^t \Xi_H^t, \qquad \omega_H^t := \frac{a_H^t}{\sum_{J \in \mathcal{G}_{\mathrm{sh}}^t} a_J^t}, \tag{43}$$

and $\omega_G^t > 0$.

2. If $G \in \mathcal{G}^t_{\mathrm{rt}}$, then there exists a slot r in $\mathcal{R}^{t+1}$, a coefficient $c^t_{G,r} \in (0,1]$, and an auxiliary state $(\bar{\Xi}^t_{G,r}, \bar{\kappa}^t_{G,r}, \bar{\tau}^t_{G,r})$ such that

$$\begin{aligned} \Xi^{t+1}_r &= c^t_{G,r}\Xi^t_G + (1-c^t_{G,r})\bar{\Xi}^t_{G,r}, \\ \kappa^{t+1}_r &= c^t_{G,r}\kappa^t_G + (1-c^t_{G,r})\bar{\kappa}^t_{G,r}, \\ \tau^{t+1}_r &= c^t_{G,r}\tau^t_G + (1-c^t_{G,r})\bar{\tau}^t_{G,r}. \end{aligned} \tag{44}$$

Hence G is preserved with positive finite-ancestry weight rather than immediately discarded, provided the corresponding routed slot is retained by the optional budget-maintenance step.

Furthermore, let $u \geq t$ and let $i \in \mathcal{C}_{u+1}$ be any client. For each retrieved slot $r \in A^{u+1}_i$, let its routed ancestry up to stage u be $(t^r_1, G^r_1), \ldots, (t^r_{m_r}, G^r_{m_r})$ ordered by write events as in Lemma C.5, with ties in the stage indices allowed. Then the warm start in Eq. (18) admits the decomposition

$$\widetilde{\Theta}^{u+1}_i = \Theta^{u+1} + \lambda_{\mathrm{mem}} \sum_{r \in A^{u+1}_i} \alpha^{u+1}_{ir} \left[(1-\eta_m)^{m_r-1} \Xi^{t^r_1}_{G^r_1} + \sum_{\ell=2}^{m_r} \eta_m (1-\eta_m)^{m_r-\ell} \Xi^{t^r_\ell}_{G^r_\ell} \right], \tag{45}$$

where m_r is the number of routed writes in the ancestry of slot r. Since $\alpha^{u+1}_{ir} > 0$ for every $r \in A^{u+1}_i$, each retrieved slot re-injects a positively weighted mixture of historically non-shareable directions into local initialization.

Proof. Equation (42) is immediate from the definition $\mathcal{G}^t_{\mathrm{rt}} = \mathcal{G}^t \setminus \mathcal{G}^t_{\mathrm{sh}}$ in Sec. B.2. For the shared path, rewrite Eq. (11) as Eq. (43). If $G \in \mathcal{G}^t_{\mathrm{sh}}$, then by definition $a^t_G \geq \tau_{\mathrm{sh}} > 0$, so $\omega^t_G > 0$. Thus every shared group enters the persistent state Θ^{t+1} with nonzero weight.

For the routed path, fix $G \in \mathcal{G}^t_{\mathrm{rt}}$. During stage t, the operator of Sec. B.2 applies routed writes sequentially to all groups in $\mathcal{G}^t_{\mathrm{rt}}$. Hence G participates in exactly one routed write event in the ancestry of some final slot $r \in \mathcal{R}^{t+1}$. If that event appends a fresh slot and no later routed write in the same stage hits that slot, then $c^t_{G,r} = 1$ and Eq. (44) is immediate. In the general case, let the final ancestry of slot r up to the end of stage t be $(s_1, H_1), \ldots, (s_m, H_m)$, ordered by write events, and suppose $(s_j, H_j) = (t, G)$. Applying Lemma C.5 to that ancestry shows that the coefficient of Ξ^t_G in Ξ^{t+1}_r is

$$c^t_{G,r} = \begin{cases} (1-\eta_m)^{m-1}, & j = 1, \\ \eta_m (1-\eta_m)^{m-j}, & j \geq 2, \end{cases}$$

which lies in $(0,1]$ because $\eta_m \in (0,1)$. The same coefficient multiplies κ^t_G and τ^t_G. Collecting all remaining ancestry terms into $(\bar{\Xi}^t_{G,r}, \bar{\kappa}^t_{G,r}, \bar{\tau}^t_{G,r})$ yields Eq. (44). Thus every routed group survives in the finite routed-bank ancestry with positive weight, even if later routed writes in the same stage are applied to the same slot.

It remains to show the read-side statement. Fix a client $i \in \mathcal{C}_{u+1}$. By Eq. (17), every selected slot $r \in A^{u+1}_i$ receives a positive normalized weight α^{u+1}_{ir}. Substituting the ancestry expansion from Lemma C.5 into Eq. (18) yields Eq. (45). Because both the softmax weights and the ancestry coefficients are nonnegative, each retrieved slot contributes a positive mixture of routed historical directions to $\widetilde{\Theta}^{u+1}_i$. Thus, once a negotiated direction fails the sharing test, it is not immediately erased: it survives in persistent routed memory and can influence any later readout that selects the corresponding slot.

D Design Rationale and Practical Cost

This section explains the connection between evidence-gated consolidation and routed parameter memory, then derives the communication, storage, and server-compute costs of the resulting update cycle.

D.1 Evidence-Gated Consolidation and Routed Reuse

Evidence-conditioned parameter exposure. Local updates combine structure with different cross-client reuse potential. A diagnostic decomposition makes this distinction explicit:

$$\sum_{i\in\mathcal{C}_t} \rho_i^t \Delta_i^t = \sum_{i\in\mathcal{C}_t} \rho_i^t \Delta_{i,\mathrm{sh}}^t + \sum_{i\in\mathcal{C}_t} \rho_i^t \Delta_{i,\mathrm{sp}}^t, \tag{46}$$

where the two terms represent broadly reusable and dataset-specific contributions for explanatory purposes. BrainNEST implements selective sharing through blockwise exposure: structural slots supply attribution vectors, and negotiated participation converts them into the group update Δ_G^t in Eq. (9). This operational choice attenuates unsupported parameter blocks before consolidation.

Persistent specialization. A group with limited cross-client support can carry structure useful to future clients. Routed parameter memory preserves such groups as entries (Ξ, κ, τ): a parameter offset, a structural key, and an acquisition anchor. This augments the shared backbone with addressable alternatives for later initialization. Different learning histories can produce similar shared parameters while leaving different offsets available for retrieval, allowing the next stage to reuse specialization according to its current clients.

Client-balanced consolidation. Scale imbalance enters federated EEG when a sample-rich dataset contributes a larger optimization signal simply because it has more trials or subjects. BrainNEST avoids using raw sample counts in server-side group writing. Family masses are normalized within each client, quality-adjusted transported support is filtered on a bounded scale, and α_i^G mixes negotiated evidence with a uniform client share. The method therefore treats dataset scale as a potential source of optimization bias rather than as automatic evidence of transferability. Large clients can still influence the backbone when their family evidence is coherent, but they cannot dominate solely by volume; small clients with valid but narrow evidence are routed rather than discarded.

Connection to the operator properties. The diagnostic decomposition motivates the blockwise exposure rule. Appendix C characterizes the implemented operators through partial-budget matching, client-weight bounds, non-expansive exposure, evidence-based gating, and finite-ancestry retention. Section 5 evaluates their combined effect on adaptation and retention.

D.2 Communication, Memory, and Scalability

We account for the costs of the write–read cycle in terms of communicated parameter blocks, active structural slots, and persistent memory entries. The server retains consensus and routed memories $(B^t, \mathcal{R}^t)$, while client-private slot memories remain local. The following expressions separate per-stage communication from cumulative server storage and retrieval costs.

Let p denote the dimension of the transferable trainable block exposed to federation, not the full Large EEG Model or any dataset-specific prediction head. Let d_z be the dimension of the family carrier z_{ij}^t and consensus entry b_m^t, let d_q be the dimension of the client sketch q_{ij}^t and routed key κ_r^t, and let L be the number of communicated block-exposure entries in ϱ_{ij}^t. For bookkeeping, define

$$M_t := |B^t|, \qquad K_t := |\mathcal{R}^t|, \qquad J_\Sigma^t := \sum_{i\in\mathcal{C}_t} J_i^t.$$

Thus M_t is the number of consensus slots, K_t is the number of routed slots already present at stage t, and J_Σ^t is the total number of active family carriers uploaded at that stage. Throughout this subsection, communication is measured by algorithmic payload size in scalars, abstracting away protocol headers and any possible multicast savings at the systems level.

Recurrent communication payload. For every active client $i \in \mathcal{C}_t$, the exact recurrent stage-t uplink payload of BrainNEST is

$$\mathrm{Up}_i^t = p + J_i^t(d_z + d_q + L + 1), \tag{47}$$

where the additive terms correspond to Δ_i^t, $\{z_{ij}^t\}_{j=1}^{J_i^t}$, $\{q_{ij}^t\}_{j=1}^{J_i^t}$, $\{\varrho_{ij}^t\}_{j=1}^{J_i^t}$, and $\{\mu_{ij}^t\}_{j=1}^{J_i^t}$, respectively. For every client $i \in \mathcal{C}_{t+1}$, the recurrent downlink payload is

$$\mathrm{Down}_i^{t+1} = p + M_{t+1} d_z, \tag{48}$$

corresponding to the personalized warm start $\widetilde{\Theta}_i^{t+1}$ and the broadcast consensus bank B^{t+1}. Consequently, the total cross-stage communication is

$$\mathrm{Comm}^t = \sum_{i\in\mathcal{C}_t} \mathrm{Up}_i^t + \sum_{i\in\mathcal{C}_{t+1}} \mathrm{Down}_i^{t+1} = \sum_{i\in\mathcal{C}_t}\Big[p+J_i^t(d_z+d_q+L+1)\Big] + \sum_{i\in\mathcal{C}_{t+1}}\Big[p+M_{t+1}d_z\Big]. \quad (49)$$

In particular, the client-side communication of BrainNEST is independent of the routed-bank size K_t and of the retrieval cardinality $|A_i^{t+1}|$.

Derivation. By Sec. 4.2 and Eq. (18), client i uploads exactly $m_{i,\uparrow}^t = (\Delta_i^t, \mathcal{U}_i^t)$, where $\Delta_i^t \in \mathbb{R}^p$ is the transferable-block update and $\mathcal{U}_i^t = \{(z_{ij}^t, q_{ij}^t, \mu_{ij}^t, \varrho_{ij}^t)\}_{j=1}^{J_i^t}$. Each z_{ij}^t contributes d_z scalars, each q_{ij}^t contributes d_q, each ϱ_{ij}^t contributes L, and each μ_{ij}^t contributes one scalar, which proves Eq. (47). Static client metadata used to define e_i are not included here, because they are provisioned once at registration time rather than transmitted as recurrent stage-level payload.

For the downlink, the method does not transmit the routed slots themselves. Instead, the server reads $\mathcal{R}^{t+1}$ internally through Eqs. (16)–(18) and sends only the resulting client-specific transferable-block vector $\widetilde{\Theta}_i^{t+1} \in \mathbb{R}^p$ together with the consensus bank $B^{t+1} = \{b_m^{t+1}\}_{m=1}^{M_{t+1}}$, whose entries all lie in $\mathbb{R}^{d_z}$. This proves Eq. (48). Summing the uplink and downlink costs over the corresponding active client sets yields Eq. (49). Since neither Eq. (48) nor Eq. (18) requires transmitting $\mathcal{R}^{t+1}$ itself, the client-side communication has no explicit dependence on K_t or on $|A_i^{t+1}|$.

The server performs routed retrieval internally and transmits the resulting personalized parameter vector. Thus the downlink depends on the transferable-block dimension and consensus-bank size, while routed-bank size affects server storage and retrieval compute.

Server memory growth. The auxiliary server memory of BrainNEST at stage t obeys

$$\mathrm{Mem}_{\mathrm{srv}}^t = M_t d_z + K_t(p + d_q + d_e), \quad (50)$$

excluding the shared backbone Θ^t. Moreover, the bank sizes satisfy the one-step growth bounds

$$M_{t+1} \le M_t + |\mathcal{G}_{\mathrm{sh}}^t|, \qquad K_{t+1} \le K_t + |\mathcal{G}_{\mathrm{rt}}^t|. \quad (51)$$

With an explicit routed-bank budget, the second inequality tightens to $K_{t+1} \le \min\{K_{\max}, K_t + |\mathcal{G}_{\mathrm{rt}}^t|\}$. Hence, for every stage t,

$$\mathrm{Mem}_{\mathrm{srv}}^t \le \left(M_1 + \sum_{s=1}^{t-1} |\mathcal{G}_{\mathrm{sh}}^s|\right) d_z + \left(K_1 + \sum_{s=1}^{t-1} |\mathcal{G}_{\mathrm{rt}}^s|\right)(p + d_q + d_e). \quad (52)$$

In particular, the persistent auxiliary memory grows at most linearly in the cumulative number of shared support-group and routed-group write events.

Derivation. By definition, B^t stores M_t vectors in $\mathbb{R}^{d_z}$, contributing $M_t d_z$ scalars. The routed bank $\mathcal{R}^t$ stores K_t triples $(\Xi_r^t, \kappa_r^t, \tau_r^t)$, where $\Xi_r^t \in \mathbb{R}^p$ is a transferable-block offset and (κ_r^t, τ_r^t) have dimensions d_q and d_e. This contributes $K_t(p + d_q + d_e)$ scalars and proves Eq. (50).

To prove Eq. (51), observe first that every shared support group $G \in \mathcal{G}_{\mathrm{sh}}^t$ generates exactly one consensus carrier $\tilde{b}_G^t$. Under Eq. (12), that carrier either merges into an existing slot of B^t or appends one new slot. Therefore each shared group can increase M_t by at most one, which yields $M_{t+1} \le M_t + |\mathcal{G}_{\mathrm{sh}}^t|$. Similarly, every routed group $G \in \mathcal{G}_{\mathrm{rt}}^t$ produces exactly one routed write candidate through Eq. (13). Under Eqs. (14) and (15), that candidate either updates an existing routed slot or appends one new slot, so each routed group can increase K_t by at most one. Hence $K_{t+1} \le K_t + |\mathcal{G}_{\mathrm{rt}}^t|$. Unrolling both recursions from stage 1 to stage t proves Eq. (52).

Persistent memory grows with negotiated write events: shared groups update the consensus bank, and routed groups update the parameter-offset bank. The bound tracks retained summaries across stages independently of the number of raw EEG trials used to produce them.

Server-side stage cost. Let L_{ot} denote the number of iterations used by the entropic partial-transport solver for one client pair in Eq. (6), and define

$$A_{\Sigma}^{t+1} := \sum_{i \in \mathcal{C}_{t+1}} |A_i^{t+1}|.$$

Then the server-side stage overhead of BrainNEST beyond ordinary local client training is

$$\begin{aligned} \mathrm{Time}_{\mathrm{srv}}^{t} = \mathcal{O}\Big(& J_{\Sigma}^{t}(d_z + d_q + L + 1) + |\mathcal{C}_t|(p + d_q + d_e + L) \\ & + \big(d_z + d_q + 1 + L_{\mathrm{ot}}\big) \sum_{i<i'} J_i^t J_{i'}^t \\ & + |\mathcal{G}^t| M_t d_z + |\mathcal{G}_{\mathrm{rt}}^t| K_t (d_q + d_e) \\ & + |\mathcal{C}_{t+1}| K_{t+1}(d_q + d_e) + p A_{\Sigma}^{t+1} \Big). \end{aligned} \tag{53}$$

In particular, if $J_i^t \le J_{\max}^t$ for all $i \in \mathcal{C}_t$, then

$$\begin{aligned} \mathrm{Time}_{\mathrm{srv}}^{t} = \mathcal{O}\Big(& J_{\Sigma}^{t}(d_z + d_q + L + 1) + |\mathcal{C}_t|(p + d_q + d_e + L) \\ & + \big(d_z + d_q + 1 + L_{\mathrm{ot}}\big) |\mathcal{C}_t|^2 (J_{\max}^t)^2 \\ & + |\mathcal{G}^t| M_t d_z + |\mathcal{G}_{\mathrm{rt}}^t| K_t (d_q + d_e) \\ & + |\mathcal{C}_{t+1}| K_{t+1}(d_q + d_e) + p A_{\Sigma}^{t+1} \Big). \end{aligned} \tag{54}$$

Derivation. The term $J_{\Sigma}^{t}(d_z + d_q + L + 1)$ accounts for reading uploaded family summaries, exposure vectors, and family masses. Bridge validation only removes retained edges and recursively partitions candidate components, so the final group memberships remain disjoint and $\sum_{G \in \mathcal{G}^t} |G| \le |\mathcal{C}_t|$. Consequently, forming $(\Delta_G^t, \bar{q}_G^t, \bar{e}_G^t)$, applying the shared update in Eq. (11), and executing routed updates in Eq. (15) together cost at most $\mathcal{O}(|\mathcal{C}_t|(p + d_q + d_e + L))$, which gives the second term.

For every unordered client pair (i, i'), constructing the cost matrix $C_{ii'}^t$ in Eq. (6) requires evaluating content, sketch, and mass-consistency terms for each pair of family slots, hence $\mathcal{O}((d_z + d_q + 1) J_i^t J_{i'}^t)$ operations. The entropic transport solve for the same pair runs on the same matrix and therefore incurs $\mathcal{O}(L_{\mathrm{ot}} J_i^t J_{i'}^t)$ additional work. Bridge validation scans the retained-edge graph and removes bridge edges recursively, costing at most $\mathcal{O}(|\mathcal{C}_t| + |\mathcal{E}_{\mathrm{neg}}^t|)$, which is dominated by pairwise matching. Summing over all unordered client pairs yields the third term in Eq. (53).

Once the support groups are formed, each group carrier $\tilde{b}_G^t$ must be compared against the current consensus bank in Eq. (12). A cosine scan over M_t entries in $\mathbb{R}^{d_z}$ costs $\mathcal{O}(M_t d_z)$ per group, giving the fourth term. For every routed group $G \in \mathcal{G}_{\mathrm{rt}}^t$, routed write-side addressing in Eq. (14) scans all current routed entries, which costs $\mathcal{O}(K_t(d_q + d_e))$ per routed group and yields the fifth term. Finally, for every future active client $i \in \mathcal{C}_{t+1}$, routed retrieval in Eq. (16) scores that client against all K_{t+1} routed slots using one sketch similarity and one acquisition-descriptor similarity, which costs $\mathcal{O}(K_{t+1}(d_q + d_e))$ per client. After the relevant slots are selected, materializing the warm start in Eq. (18) costs $\mathcal{O}(p|A_i^{t+1}|)$ for that client. Summing these two costs over $\mathcal{C}_{t+1}$ yields the last two terms.

To obtain Eq. (54), note that

$$\sum_{i<i'} J_i^t J_{i'}^t \le \binom{|\mathcal{C}_t|}{2} (J_{\max}^t)^2 = \mathcal{O}\big(|\mathcal{C}_t|^2 (J_{\max}^t)^2\big),$$

and substitute this bound into Eq. (53).

The dominant server costs arise from pairwise family matching, consensus-bank lookup, and routed-memory retrieval. They depend on the active structural summaries and stored memory entries, with local trial-level processing performed by the clients.

Routed memory decouples continuality from traffic. BrainNEST decouples *continuality* from *network traffic* in the following precise sense:

$$\mathrm{Down}_i^{t+1} = p + M_{t+1} d_z \quad \text{for every } i \in \mathcal{C}_{t+1}, \tag{55}$$

and this expression contains no K_t-dependent term, while the routed-memory growth satisfies

$$K_t \le K_1 + \sum_{s=1}^{t-1} |\mathcal{G}_{\mathrm{rt}}^s|.$$

Hence increasing the amount of preserved routed specialization affects server storage and server-side retrieval cost, but does not by itself increase the communicated payload received by each client.

Derivation. The identity for Down_i^{t+1} is exactly Eq. (48), which contains no K_t-dependent term. Therefore routed-bank growth does not enter the closed-form expression for the transmitted payload. The bound on K_t follows from Eq. (52). Combining the two statements proves the claim.

The accounting separates three costs: communication of current updates and sketches, storage of accumulated server memory, and computation for matching and retrieval. Routed memory is read on the server, so each client receives a personalized initialization whose size is set by the transferable parameter block.

E Experimental Details

E.1 Large EEG Model Backbones

BrainNEST is evaluated as a federated continual wrapper around pretrained EEG encoders rather than as a replacement for the encoder itself. We use CBraMod and CodeBrain as representative compatible backbones, covering transformer-style cross-brain representation learning and multi-scale tokenization-based large EEG representation learning. Within each reported comparison, the selected backbone and preprocessing pipeline are fixed for all methods whenever applicable, so performance differences are attributed to the federated continual training strategy rather than to changes in the underlying EEG feature extractor.

Operationally, each active client applies the backbone to its own preprocessed EEG trials and obtains latent sequences H_i^t. BrainNEST then acts on the resulting client-side latent structure through local family slots, compact uploaded summaries, sparse family matching, consensus-supported aggregation, and routed memory retrieval. This separation is important for reproducibility: the backbone determines the trial-level representation space, whereas BrainNEST determines how heterogeneous datasets exchange, preserve, and reuse structure across stages.

E.2 Evaluation Metrics

This subsection gives the formal definitions of the continual-learning metrics reported in the main experiments. Let $\mathcal{N}_t$ be the set of datasets newly introduced at stage t, and let

$$\mathcal{A}_t = \bigcup_{k=1}^{t} \mathcal{N}_k$$

denote all datasets observed up to stage t. After training through stage t, let $R_{t,j}$ denote the balanced accuracy on dataset j, evaluated with the dataset-specific prediction head and fixed test split. The first stage in which dataset j appears is denoted by $\tau(j)$. When a stage-level average involves datasets with different test-set sizes, we use weights n_j proportional to the number of test examples; setting all $n_j = 1$ gives the unweighted version.

The anytime average accuracy measures performance over the entire learning trajectory rather than only at the final checkpoint:

$$\mathrm{AAA}_t = \frac{\sum_{k=1}^{t} \sum_{j \in \mathcal{A}_k} n_j R_{k,j}}{\sum_{k=1}^{t} \sum_{j \in \mathcal{A}_k} n_j}. \tag{56}$$

The weighted new-stage balanced accuracy measures plasticity on the datasets introduced at the current stage:

$$\mathrm{New}_t^{\mathrm{wBA}} = \frac{\sum_{j\in\mathcal{N}_t} n_j R_{t,j}}{\sum_{j\in\mathcal{N}_t} n_j}. \tag{57}$$

Average forgetting is computed on datasets observed before stage t:

$$\mathrm{Forget}_t = \frac{\sum_{j\in\mathcal{A}_{t-1}} n_j \max\big(0,\ \max_{\tau(j)\le k<t} R_{k,j} - R_{t,j}\big)}{\sum_{j\in\mathcal{A}_{t-1}} n_j}, \qquad t>1. \tag{58}$$

Finally, backward transfer compares the current performance on old datasets against the performance immediately after those datasets first appeared:

$$\mathrm{BWT}_t = \frac{\sum_{j\in\mathcal{A}_{t-1}} n_j \big(R_{t,j} - R_{\tau(j),j}\big)}{\sum_{j\in\mathcal{A}_{t-1}} n_j}, \qquad t>1. \tag{59}$$

For $t=1$, forgetting and BWT are reported as zero because no previous dataset exists. In the main tables, we report the final-stage values AAA_T, $\mathrm{New}_T^{\mathrm{wBA}}$, Forget_T, and BWT_T.

Unseen transfer protocol. For Table 5, eldBETA and TUAB are held out from all stage-wise BrainNEST training. Each reported method is then fully fine-tuned on the unseen dataset for 100 epochs using the same train/validation/test split, optimizer, and evaluation-head protocol. BrainNEST uses the final continual backbone corresponding to its base encoder as initialization; no raw data or labels from the unseen datasets are used before this fine-tuning step.

E.3 Dataset Details, Preprocessing, and Label Semantics

Each downstream dataset is treated as one federated client, and the stage construction follows the chronological grouping reported in Table 1. The benchmark covers seven BCI task families: emotion recognition, motor imagery, sleep staging, SSVEP decoding, imagined speech, mental disorder diagnosis, and mental stress detection. This design intentionally exposes the learner to heterogeneous label spaces, subject populations, recording devices, channel montages, trial lengths, and class imbalance patterns. The purpose is not to create an artificially homogeneous multi-dataset benchmark, but to evaluate whether a federated continual method can retain useful cross-dataset structure while avoiding forced transfer between incompatible EEG ecologies.

Table 9 reports the train/validation/test protocol used for each downstream client. We use the same split for all methods on a given dataset. In line with the main text, datasets with sufficiently many participants are evaluated in a cross-subject protocol, where the test set contains held-out subjects. Smaller datasets are evaluated with multi-subject or session-aware splits so that each subject still contributes enough training evidence while held-out trials or sessions are excluded from parameter updates. For clinical and affective datasets, subject-level splits are stratified whenever label metadata permit stratification.

We follow dataset-specific trial definitions when datasets are already segmented, and use task-standard windowing for continuous recordings. Preprocessing is applied only within the corresponding client before federated training. Invalid trials are removed, channels are aligned according to the backbone input convention when required, and normalization statistics are estimated from the training split only. No raw EEG segment is uploaded to the server. After a client leaves a stage, its raw data are no longer accessible; later stages are influenced only through server-persistent consensus and routed memory rather than by raw-data replay.

Instruction and target texts. For transparency, we provide the instruction and target texts associated with each downstream dataset in Table 10. The table follows the common instruction-style reporting format used in recent Large EEG Model evaluations, but its role in this paper is deliberately limited: the texts define a reproducible semantic interface for downstream labels and task descriptions. They do not introduce an additional server-side language model into BrainNEST, nor do they alter the federated communication protocol. Instead, they make label semantics explicit across heterogeneous clients, which is useful when comparing datasets whose class indices alone would obscure the task meaning.

Table 9: Train/validation/test split strategies for downstream EEG datasets. "Train/val" denotes the pool from which the validation subset is selected; validation examples are never used for parameter updates.

Stage	Dataset	Split strategy
1	BCIC-IV-2a	Multi-subject trial-based: for each subject, the first 75% of trials are used for train/val and the last 25% for test; 20% of the train/val pool is used for validation.
1	PhysioNet-MI	Cross-subject: subjects 1–80 are used for train/val and subjects 81–109 for test; 20% of the train/val subjects are used for validation.
1	DEAP-VA	Cross-subject: subjects 1–25 are used for train/val and subjects 26–32 for test; 20% of the train/val subjects are used for validation.
1	SEED	Multi-subject trial-based after removing neutral trials: positive and negative trials are ordered by recording index, with 60% for training, 20% for validation, and 20% for test.
2	Mumtaz	Stratified cross-subject: 80% of subjects are used for train/val and 20% for test; 20% of the train/val subjects are used for validation.
2	Shin2017A	Multi-subject trial-based: for each subject, the first 75% of trials are used for train/val and the last 25% for test; 20% of train/val trials are used for validation.
2	MentalArithmetic	Cross-subject: 80% of subjects are used for train/val and 20% for test; validation is drawn from 20% of the train/val subjects.
3	BCIC2020-3	Multi-subject trial-based: the first 300 trials are used for training and the remaining trials for test; validation is drawn from the training portion.
3	BETA	Cross-subject: subjects 1–46 are used for training, subjects 47–50 for validation, and subjects 51–70 for test.
3	SEED-V	Multi-subject trial-based: for each subject, trials 1–5 are used for training, trials 6–10 for validation, and trials 11–15 for test.
3	BCIC-UpperLimb	Session-aware split: the official day/session protocol is preserved, with early-session data for training, the validation session for monitoring, and the final held-out session for test.
4	HMC	Cross-subject sleep-staging split: recordings from held-out subjects are reserved for test, and 20% of the remaining train/val subjects are used for validation. Sleep epochs from the same recording are not split across train and test.
4	SHU-MI	Multi-session motor-imagery split: earlier sessions are used for train/val and the final session is held out for test; 20% of train/val trials are used for validation.
4	FACED	Cross-subject: subjects 1–100 are used for train/val and the remaining subjects for test; 20% of the train/val subjects are used for validation.

The target texts are intentionally short and label-aligned. For tasks with naturally named categories, such as motor imagery, sleep staging, and discrete emotion recognition, we use the canonical class semantics. For SSVEP, where classes correspond to stimulus frequencies rather than linguistic categories, we describe the frequency grid compactly instead of enumerating all classes. For imagined speech, the target texts follow the five command-like categories used by the dataset. This normalization keeps the downstream interface comparable across clients while preserving the original label meaning of each dataset.

Additional dataset notes. The following notes complement the split strategies above and make the benchmark composition auditable without interrupting the main experimental narrative.

BCIC-IV-2a: BCIC-IV-2a is a standard motor-imagery benchmark from BCI Competition IV. It contains recordings from nine subjects performing four imagery classes: left hand, right hand, feet, and tongue. EEG was recorded with 22 scalp channels, and we use it as a small but canonical Stage 1 motor-imagery client.

PhysioNet-MI: PhysioNet-MI is the EEG Motor Movement/Imagery dataset collected with the BCI2000 system. It includes recordings from 109 healthy subjects with a 64-channel montage. We use the four-class formulation involving left fist, right fist, both fists, and both feet, making it the large-scale motor-imagery client in Stage 1.

DEAP-VA: DEAP-VA contains affective EEG recordings from 32 subjects watching music videos. The dataset provides valence and arousal annotations, and we use the four-quadrant valence–arousal formulation. This client tests whether the model can transfer from motor imagery to affective-state decoding within the same stage.

SEED: SEED is an emotion-recognition dataset with 15 subjects and 62 EEG channels. The original labels correspond to positive, neutral, and negative affective states; in our benchmark, we follow

Table 10: Instruction and target texts for the downstream datasets used in the stage-wise federated continual benchmark. Target texts are written as normalized label semantics rather than raw numeric class identifiers.

Stage	Dataset	Task family	Instruction text	Target texts
1	BCIC-IV-2a	Motor imagery	Decode motor imagery from EEG.	left hand; right hand; feet; tongue
1	PhysioNet-MI	Motor imagery	Decode motor imagery and movement from EEG.	left fist; right fist; both fists; both feet
1	DEAP-VA	Emotion recognition	Decode valence-arousal emotion state from EEG.	low valence low arousal; low valence high arousal; high valence low arousal; high valence high arousal
1	SEED	Emotion recognition	Decode binary discrete emotion state from EEG.	negative; positive
2	Mumtaz	Mental disorder diagnosis	Diagnose depression-related EEG state.	healthy control; depression
2	Shin2017A	Motor imagery	Decode left-versus-right hand motor imagery from EEG.	left hand; right hand
2	MentalArithmetic	Mental stress detection	Identify mental arithmetic stress from EEG.	resting baseline; mental arithmetic stress
3	BCIC2020-3 Speech	Imagined speech	Decode imagined speech content from EEG.	hello; help-me; stop; thank-you; yes
3	BETA	SSVEP	Decode SSVEP stimulus frequency from EEG.	40 frequency classes from 8.0 Hz to 15.8 Hz with 0.2 Hz spacing
3	SEED-V	Emotion recognition	Decode five-way emotion state from EEG.	disgust; fear; sad; neutral; happy
3	BCIC-UpperLimb	Grasp imagery	Decode upper-limb grasp imagery from EEG.	cylindrical grasp; spherical grasp; lumbrical grasp
4	HMC	Sleep staging	Decode sleep stage from EEG.	wake; N1; N2; N3; REM
4	SHU-MI	Motor imagery	Decode hand motor imagery from EEG.	left hand; right hand
4	FACED	Emotion recognition	Decode fine-grained emotion state from EEG.	anger; fear; disgust; sadness; amusement; inspiration; joy; tenderness; neutral

a binary affective setting by retaining only positive and negative trials and excluding neutral trials. SEED therefore introduces discrete positive-versus-negative emotion recognition into Stage 1.

Mumtaz: Mumtaz is used as the depression-screening client. It contains clinical EEG recordings with a 19-channel montage and binary labels distinguishing healthy controls from participants with depression. This dataset introduces a clinical diagnostic shift that differs substantially from both motor imagery and affective recognition.

Shin2017A: Shin2017A is an open EEG/NIRS motor-imagery dataset collected from healthy participants across repeated sessions. We use the EEG stream for the two-class left-versus-right hand motor-imagery task. Its session structure makes it useful for evaluating whether local family structure remains stable under repeated subject-level measurements.

MentalArithmetic: MentalArithmetic contains EEG recordings during resting and arithmetic-task conditions. We use it as a two-class mental stress/workload recognition dataset with 23 EEG channels. This client introduces cognitive-load variation rather than motor, affective, or clinical labels.

BCIC2020-3: BCIC2020-3 is used as the imagined-speech client. The task contains five command-like imagined-speech categories, which differ semantically from motor imagery and emotion labels. Including this dataset in Stage 3 tests whether BrainNEST can route useful specialization when the task family changes sharply.

BETA: BETA is a large-scale SSVEP benchmark with 70 subjects and 40 visual stimulus frequencies. It has the largest label cardinality in our downstream benchmark. This client is important for evaluating robustness under frequency-coded labels and strong class-granularity mismatch.

SEED-V: SEED-V extends discrete EEG emotion recognition to five emotion categories. It contains 16 subjects recorded with a 62-channel montage. Compared with the binary SEED setting, it increases label granularity and therefore tests whether affective structure can be reused without assuming identical class spaces.

BCIC-UpperLimb: BCIC-UpperLimb focuses on upper-limb grasp imagery. We use the three-class benchmark over cylindrical, spherical, and lumbrical grasp. This dataset is a fine-grained motor client whose useful specialization should not be indiscriminately averaged with unrelated clients.

HMC: HMC is used for sleep staging. We evaluate the five standard sleep stages: wake, N1, N2, N3, and REM. Its long temporal dynamics and sleep-specific label semantics make it a difficult Stage 4 client for continual transfer.

SHU-MI: SHU-MI is a motor-imagery dataset used as a later-stage client. We use the two-class left-versus-right hand setting with a 32-channel montage. It evaluates whether motor-related structure learned in earlier stages can still provide useful initialization after several heterogeneous stage transitions.

FACED: FACED is a large-scale emotion-recognition dataset with 123 subjects and 32 EEG channels. We use its nine fine-grained emotion categories. As a late-stage affective client, FACED tests whether BrainNEST can preserve and reuse affective structure while adapting to a larger and more fine-grained label space.

Across all datasets, preprocessing is intentionally client-local. Each client performs channel selection or alignment required by the backbone, trial segmentation according to the dataset protocol, and normalization using training statistics only. The server receives only model updates and compact family summaries. This preserves the federated assumption used by BrainNEST and prevents the downstream benchmark from relying on centralized raw EEG access.

E.4 Single-Stage Federated Learning Baselines

This subsection summarizes the single-stage federated baselines used in Table 2. These methods are evaluated within a fixed stage, so they test whether BrainNEST improves ordinary federated EEG learning before the additional difficulty of stage-wise client turnover is introduced.

Single: The single-client baseline trains one independent model for each dataset without any federated communication. It measures the performance obtained when every dataset is treated as an isolated learning problem. This baseline is important because it separates the benefit of cross-client transfer from the benefit of simply using a strong EEG backbone.

Central: The centralized baseline pools all datasets within a stage and trains one shared encoder with dataset-specific local prediction heads under non-federated access. It is not deployable under federated data-access constraints, but it provides an oracle-style reference for what can be obtained when raw data are co-visible under the same head protocol used by all methods. Comparing against it clarifies whether federated negotiation can approach centralized sharing without raw EEG pooling.

FedAvg: FedAvg is the standard weighted model-averaging baseline for federated learning. Each client trains locally and the server aggregates parameter updates by client weight. It is simple and strong when client objectives are compatible, but it has no mechanism to separate shareable EEG structure from dataset-specific drift.

FedProx: FedProx modifies local training by adding a proximal term that keeps each client model close to the current global model. This reduces client drift under heterogeneous data and unstable local optimization. However, the server still receives and aggregates whole-model updates rather than negotiating which structures should be shared.

SCAFFOLD: SCAFFOLD (Karimireddy et al., 2020) introduces control variates to correct the bias between local client gradients and the global objective. Its main purpose is to stabilize federated optimization under non-IID clients. In our setting, it provides a strong optimization-control baseline but does not explicitly model cross-dataset EEG family correspondences.

FedPer: FedPer separates the model into shared representation layers and private personalization layers. The shared part is aggregated by the server, while the private part remains client-specific. This fixed architectural split is useful for personalization, but it assumes the boundary between shared and private knowledge is predetermined rather than negotiated from stage evidence.

FedBABU: FedBABU aggregates the body of the network while keeping the classifier head local. The method encourages a generally useful representation and relies on client-side heads for personalization. It is a natural comparison for Large EEG Model backbones because it tests whether body-level sharing alone is sufficient under heterogeneous downstream tasks.

FedROD: FedROD bridges generic and personalized federated learning by separating generic and personalized predictors. It is designed to handle label and data heterogeneity through decoupled prediction paths. In our benchmark, it tests whether predictor-level decoupling can match BrainNEST's family-level negotiation and routed specialization.

FedAMP: FedAMP performs personalized federated learning through attentive message passing among clients. Each client receives a similarity-weighted combination of other client models rather than a single uniform global average. This is a strong personalized aggregation baseline, but its similarity is model-level rather than family-structure-level.

FedALA: FedALA performs adaptive local aggregation by learning how much external information each client should absorb. It improves personalization by allowing client-specific aggregation behavior. The comparison is useful because BrainNEST also controls transfer, but it does so through sparse family matching, shareability gating, and routed memory rather than only aggregation weights.

FeDaL: FeDaL is a federated dataset-learning baseline for heterogeneous time-series foundation models. It models dataset-level heterogeneity more explicitly than standard averaging and is therefore a relevant comparison for EEG datasets with different recording ecologies. We include it to test whether dataset-aware federated pretraining alone is sufficient without BrainNEST's stage-persistent consensus memory, sparse family negotiation, and routed specialization.

E.5 Federated Continual Learning Baselines

This subsection summarizes the federated continual learning baselines used in Table 3. These methods are evaluated across all stages, so they test stability, plasticity, and retention under client arrival, departure, and task-family shifts.

Baseline: Baseline is the direct federated continual learning baseline that carries the global model from one stage to the next and continues federated training on newly active clients. It preserves knowledge only through the evolving shared parameters. Therefore, it tests whether simple stage-wise continuation is enough without explicit forgetting control, memory routing, or client-aware readout.

EWC: EWC extends federated stage-wise training with elastic weight consolidation. Important parameters from previous stages are penalized when they change too much during new-stage learning. This provides a stability-oriented baseline, but it protects parameters globally rather than deciding which EEG structures should be shared, routed, or kept local.

LWF: LWF uses learning without forgetting to distill predictions or representations from the previous model while learning the current stage. It preserves old behavior through a teacher-student constraint. This method is useful for testing distillation-based retention, but it does not maintain an addressable memory for non-shareable client specialization.

OGD: OGD applies orthogonal gradient descent in the continual setting. New-stage gradients are constrained to be less interfering with directions associated with previous knowledge. It directly targets forgetting at the optimization level, whereas BrainNEST targets forgetting through negotiated sharing and routed reusable memory.

FedWEIT: FedWEIT decomposes knowledge into global and task-adaptive components and performs weighted inter-client transfer. It is a representative federated continual method that explicitly models transferable and private knowledge. We compare with it because both methods seek reusable specialization, but BrainNEST writes specialization into server-routed memory indexed by family sketches and non-semantic acquisition anchors.

FOT: FOT mitigates global catastrophic forgetting through federated orthogonal training. It constrains the evolution of the shared model so that learning new tasks is less destructive to previous ones. This

provides a strong trajectory-regularization baseline, while BrainNEST additionally decides whether each negotiated direction should be globally consolidated or routed for later personalized reuse.

ReFed: ReFed is the recent reweighting-style federated continual baseline in our comparison, designed to improve retention by controlling how previous and current federated information are reused. It represents methods that refine the global trajectory after stage changes. Unlike BrainNEST, it does not explicitly form family-level support groups or preserve rejected-but-reusable directions in a routed memory bank.

F Additional Experimental Details and Results

This appendix first provides the implementation details needed to reproduce BrainNEST, then reports the full stage-wise single-stage federated learning results that complement Table 2. Each stage is evaluated independently by treating the datasets active in that stage as concurrent federated clients. These results provide a more detailed view of how BrainNEST behaves under different task mixtures, dataset sizes, and EEG acquisition ecologies.

F.1 Implementation Details and Hyperparameters

Training workflow. All methods use the same resampled EEG signals, dataset splits, active-client schedule, LEM backbone, local training budget, and communication schedule. Each public dataset is treated as one client, and no pseudo-clients are created by random partitioning. BrainNEST follows the stage transition described in Sec. 4: the server broadcasts the shared state and client-specific routed warm start, active clients train locally and upload transferable updates with family summaries, and the server writes validated evidence into the shared consensus path or routed memory. After a stage finishes, only the server state $(\Theta, B, \mathcal{R})$ is carried forward; historical raw EEG is not revisited. Dataset-specific prediction heads remain local, while the transferable backbone block is the only parameter component exposed to federation. BrainNEST uses only non-semantic acquisition metadata for routing and retrieval; task-family labels, label counts, dataset names, training-set sizes, and raw EEG statistics are not used as server-side routing features.

Table 11: Default training and evaluation settings used in the main experiments. These settings are fixed across datasets and methods unless required by a baseline's original algorithm.

Category	Setting
Input preprocessing	All EEG signals are resampled to 200 Hz; dataset-specific preprocessing follows Appendix E.3.
Client construction	One public EEG dataset is one client; all active clients participate in each communication round.
Random initialization seed	Seed 42 is used for deterministic split construction, added-module initialization, and data-order shuffling.
Federated communication budget	120 communication rounds for each active stage, both in single-stage FL and in the stage-wise continual stream.
Local training	1 local epoch per communication round, batch size 64, AdamW optimizer, learning rate 2×10^{-4}, weight decay 10^{-4}.
Compute resources	All experiments are conducted using four NVIDIA A100 GPUs.
Optimization control	Gradient norm clipping at 1.5, warm-up for the first 10 rounds, minimum round learning-rate scale 0.35.
Local objective	Dataset-specific supervised loss plus structural sufficiency and consensus alignment with $\lambda_{\text{aux}} = 0.10$.
Stage-local personalization	After each stage update, only currently active clients perform 10 local personalization epochs on their own training split before in-stream evaluation. Historical clients are not retrained after leaving their stage.
In-stream evaluation	Local prediction heads use batch size 64, AdamW learning rate 10^{-4}, weight decay 10^{-4}, label smoothing 0.1, and dropout 0.25.
Unseen transfer	Held-out unseen datasets use full model fine-tuning for 100 epochs under the same split, optimizer, and evaluation-head protocol for all methods.
Validation and test use	Validation splits are used only for method-independent monitoring; held-out test data are evaluated once for final reporting.

BrainNEST hyperparameters. Table 12 lists the configuration used across stages and datasets. Family summaries induce the attribution vector ϱ_{ij}^t, which controls parameter-block exposure before either shared or routed writing.

Table 12: Default BrainNEST hyperparameters aligned with the operators in Sec. 4 and Appendix B. Acquisition descriptors are non-semantic and are used only for compatibility scoring, not for task-label supervision.

Component	Hyperparameter	Value
Acquisition descriptor	Non-semantic descriptor dimension d_e	13
Family atomization	Multi-scale pooling set; shared latent basis	$\{1, 2, 4\}$; backbone latent space
Family slots	Maximum active family slots $J_{\max}$; local assignment neighborhood K_a	8; 3
Family maintenance	Dustbin residual; memory write rate η_m; slot editing rule	enabled; 0.20; uniform-support baseline
Family-conditioned exposure	Transferable-block exposure ϱ_{ij}^t; block partition	enabled; natural backbone blocks
Motion sketch	Motion-sketch dimension d_q; shared sketch projection W_q	128; fixed during each round
Partial transport	Budget ratio ρ_{pt}; entropy coefficient ϵ; solver iterations	0.20; 0.70; 30
Retained edge filter	Quality-adjusted mass floor τ_m; average-cost ceiling τ_c	0.05; 1.00
Group weighting	Client-balanced evidence coefficient λ_{ev}; bridge validation	0.50; enabled
Shareability gate	Shared-write threshold τ_{sh}; singleton rule	0.08; routed only
Consensus memory	Consensus entry update rate; append/update rule	$\eta_m = 0.20$; stage-average compatibility
Routed memory	Readout Top-K; readout strength λ_{mem}; routed write rate	4; 0.50; $\eta_m = 0.20$
Memory budget	Routed-memory capacity $K_{\max}$	64 entries; explicit budget eviction only after routed writes

Baseline protocol and fairness. For baseline methods, we keep the same backbone, local optimizer, communication rounds, downstream evaluation protocol, and active-client schedule. Method-specific hyperparameters follow the recommended values in the corresponding papers; when a method requires a scalar coefficient, we use the same validation-only protocol across methods and freeze the protocol before test evaluation. In particular, FedProx uses $\mu = 10^{-3}$, distillation-style baselines use temperature 4 when applicable, and continual regularization coefficients are taken from $\{10^{-3}, 10^{-2}, 10^{-1}, 1\}$. Continual baselines use the same 120-round within-stage federated budget and the same 10-epoch stage-local personalization budget for currently active clients. No inactive historical client is retrained for any method, and no baseline is given less local training, semantic server metadata, or a different downstream evaluation head than BrainNEST.

F.2 Stage-wise Single-Stage Federated Learning Results

Table 13 shows that BrainNEST obtains the best result on every Stage 1 metric across both motor-imagery and emotion-recognition clients. The gains are especially clear on BCIC-IV-2a and DEAP-VA, where direct averaging and standard personalization are less stable under mixed task families. This supports the role of family-level negotiation: BrainNEST shares only structure with cross-client evidence while avoiding over-aggregation of dataset-specific EEG patterns.

Table 14 shows that BrainNEST remains the strongest overall method under the more heterogeneous Stage 2 mixture. It achieves the best BA and F1 on all three datasets and gives a large gain on the clinical Mumtaz client, while remaining competitive on ROC-AUC where centralized or personalized baselines occasionally peak. These results suggest that negotiated sharing is especially useful when clinical, stress, and motor-imagery updates coexist within the same federated stage.

Table 15 reports the most heterogeneous single-stage mixture, combining affective, motor, speech, and frequency-coded SSVEP tasks. BrainNEST achieves the best BA and WF1 on all four datasets and the strongest overall performance on BCIC-UpperLimb, BCIC2020-3, and BETA, showing that

Table 13: Stage 1 single-stage federated learning results. Stage 1 contains motor-imagery and emotion-recognition clients.

Method	BCIC-IV-2a			PhysioNet-MI			DEAP-VA			SEED		
	BA	WF1	κ	BA	WF1	κ	BA	WF1	κ	BA	ROC-AUC	F1
Single	0.4601	0.4307	0.2801	0.6068	0.6052	0.4757	0.3174	0.3245	0.0827	0.6679	0.7505	0.6890
Central	0.5295	0.5279	0.3727	0.5973	0.5977	0.4631	0.3099	0.3228	0.0786	0.7049	0.7602	0.7067
FedAvg	0.5382	0.5304	0.3843	0.6084	0.6090	0.4779	0.3190	0.3238	0.0836	0.6889	0.7658	0.6978
FedProx	0.5312	0.5264	0.3750	0.6091	0.6081	0.4787	0.3009	0.3235	0.0665	0.7049	0.7746	0.6924
SCAFFOLD	0.4323	0.3586	0.2431	0.6113	0.6128	0.4816	0.3277	0.3269	0.0976	0.6975	0.7768	0.6871
FedBABU	0.4870	0.4610	0.3161	0.6052	0.6058	0.4735	0.3392	0.3546	0.1222	0.6963	0.7545	0.6978
FedPer	0.5156	0.5053	0.3542	0.6134	0.6140	0.4846	0.3104	0.3097	0.0758	0.6901	0.7767	0.7105
FedROD	0.5174	0.5100	0.3565	0.6184	0.6170	0.4912	0.3223	0.3427	0.1019	0.7111	0.7692	0.7201
FedAMP	0.5104	0.4951	0.3472	0.6084	0.6046	0.4778	0.3264	0.3129	0.0970	0.6877	0.7663	0.7047
FedALA	0.5304	0.5204	0.3738	0.6167	0.6187	0.4890	0.3128	0.3121	0.0765	0.6889	0.7427	0.6834
FeDaL	0.5321	0.5261	0.3762	0.6107	0.6090	0.4809	0.3358	0.3287	0.1096	0.7185	0.7731	0.7000
BrainNEST	**0.5764**	**0.5775**	**0.4352**	**0.6263**	**0.6264**	**0.5016**	**0.3487**	**0.3627**	**0.1313**	**0.7247**	**0.7823**	**0.7202**

Table 14: Stage 2 single-stage federated learning results. Stage 2 contains mental disorder diagnosis, mental stress detection, and motor-imagery clients.

Method	Mumtaz			MentalArithmetic			Shin2017A		
	BA	ROC-AUC	F1	BA	ROC-AUC	F1	BA	ROC-AUC	F1
Single	0.8775	0.9601	0.8701	0.7083	0.7895	0.5682	0.6889	0.7639	0.6512
Central	0.8844	0.9549	0.8772	0.7188	0.8326	0.5843	0.6843	0.7640	0.6423
FedAvg	0.8923	0.9568	0.8859	0.6979	0.8264	0.5474	0.6889	0.7605	0.6475
FedProx	0.8966	0.9588	0.8873	0.7014	0.8025	0.5510	0.6827	0.7660	0.6776
SCAFFOLD	0.8862	0.9244	0.8795	0.6875	0.8212	0.5333	0.6726	0.7575	0.6183
FedBABU	0.8773	0.9184	0.8673	0.7014	0.8207	0.5490	0.6913	0.7651	0.6599
FedPer	0.8891	0.9615	0.8852	0.6808	0.7795	0.5217	0.6827	0.7652	0.6313
FedROD	0.8853	0.9645	0.8794	0.7083	0.8027	0.5600	0.6800	0.7617	0.6534
FedAMP	0.8820	0.9588	0.8791	0.7118	0.7672	0.5454	0.6846	**0.7686**	0.6634
FedALA	0.8984	0.9677	0.8966	0.7222	0.8378	0.5769	0.6796	0.7660	0.6776
FeDaL	0.8940	0.9669	0.8932	0.7049	0.7878	0.5505	0.6867	0.7635	0.6548
BrainNEST	**0.9175**	**0.9815**	**0.9162**	**0.7292**	**0.8411**	**0.6000**	**0.7008**	0.7677	**0.6860**

Table 15: Stage 3 single-stage federated learning results. Stage 3 contains emotion recognition, upper-limb motor imagery, imagined speech, and SSVEP clients.

Method	SEED-V			BCIC-UpperLimb			BCIC2020-3			BETA		
	BA	WF1	κ	BA	WF1	κ	BA	WF1	κ	BA	WF1	κ
Single	0.3996	0.4011	0.2473	0.4200	0.4191	0.1300	0.3133	0.3140	0.1417	0.6069	0.6066	0.5968
Central	0.3961	0.4009	0.2469	0.4411	0.4367	0.1617	0.2813	0.2812	0.1017	0.6206	0.6197	0.6109
FedAvg	0.3970	0.3983	**0.2580**	0.4189	0.4188	0.1283	0.2880	0.2883	0.1100	0.6491	0.6487	0.6401
FedProx	0.3925	0.3961	0.2431	0.4078	0.4080	0.1117	0.2867	0.2859	0.1083	0.6519	0.6517	0.6429
SCAFFOLD	0.3986	0.4043	0.2517	0.4300	0.4306	0.1450	0.2707	0.2705	0.0883	0.6172	0.6165	0.6074
FedBABU	0.3973	0.4003	0.2465	0.4322	0.4265	0.1483	0.2813	0.2805	0.1017	0.6338	0.6339	0.6221
FedPer	0.3906	0.3942	0.2384	0.4289	0.4279	0.1433	0.2747	0.2738	0.0933	0.6197	0.6193	0.6099
FedROD	0.3974	0.4004	0.2465	0.4333	0.4052	0.1500	0.2827	0.2825	0.1033	0.6225	0.6231	0.6128
FedAMP	0.3965	0.4033	0.2493	0.4067	0.4073	0.1100	0.2600	0.2584	0.0750	0.6384	0.6385	0.6292
FedALA	0.3953	0.3972	0.2440	0.4500	0.4004	0.1750	0.3267	0.3275	0.1583	0.6362	0.6364	0.6269
FeDaL	0.3959	0.4004	0.2459	0.4367	0.4316	0.1500	0.3453	0.3445	0.1817	0.6378	0.6379	0.6285
BrainNEST	**0.4020**	**0.4072**	0.2541	**0.4611**	**0.4605**	**0.1917**	**0.3667**	**0.3670**	**0.2083**	**0.6634**	**0.6630**	**0.6548**

family negotiation remains effective even when label semantics and temporal structures differ sharply. The only non-best entry is SEED-V κ, where FedAvg is slightly higher, but BrainNEST still gives the best accuracy and F1.

Table 16: Stage 4 single-stage federated learning results. Stage 4 contains fine-grained emotion recognition, motor imagery, and sleep-staging clients.

Method	FACED			SHU-MI			HMC		
	BA	WF1	κ	BA	ROC-AUC	F1	BA	WF1	κ
Single	0.5270	0.5275	0.4651	0.6125	0.6782	0.5864	0.7181	0.7752	0.6984
Central	0.4826	0.4856	0.4164	0.6143	0.6825	0.5283	0.7200	0.7721	0.7058
FedAvg	0.4976	0.5009	0.4327	0.6238	0.6927	0.5746	0.7231	0.7713	0.6963
FedProx	0.5030	0.4382	0.4382	0.6187	0.6840	0.5701	0.7140	0.7590	0.6963
SCAFFOLD	0.4913	0.4927	0.4253	0.5951	0.6570	0.5583	0.7171	0.7575	0.6748
FedBABU	0.5133	0.5131	0.4491	0.5954	0.6495	0.5778	0.7219	0.7690	0.6855
FedPer	0.5187	0.5203	0.4562	0.6101	0.6513	0.5720	0.7297	0.7745	0.6904
FedROD	0.5203	0.5216	0.4577	0.6080	0.6648	0.5591	0.7259	0.7729	0.6930
FedAMP	0.5356	0.5331	0.4729	0.6058	0.6535	0.5685	0.7163	0.7597	0.6760
FedALA	0.5401	0.5396	0.4794	0.6245	0.6990	**0.6758**	0.7239	0.7586	0.6844
FeDaL	0.5309	0.5310	0.4687	0.6119	0.6771	0.6105	0.7286	0.7672	0.6930
BrainNEST	**0.5443**	**0.5441**	**0.4842**	**0.6345**	**0.7026**	0.6672	**0.7454**	**0.7911**	**0.7207**

Table 16 shows that BrainNEST also performs strongly in the final single-stage setting, where large-scale emotion and sleep datasets coexist with a later motor-imagery client. BrainNEST achieves the best results on FACED and HMC across all metrics and leads SHU-MI in BA and ROC-AUC, with only SHU-MI F1 slightly favoring FedALA. This supports that the negotiated memory interface remains beneficial even when the stage contains both large late-stage datasets and task-family revisits.

F.3 Federated Continual Learning Dynamics

Beyond the final continual metrics in Table 3, we visualize the full stage-wise learning trajectory in Fig. 4. The curves track how each method evolves as new EEG datasets arrive, making the stability–plasticity behavior visible across the complete stream. Figure 4 shows that BrainNEST

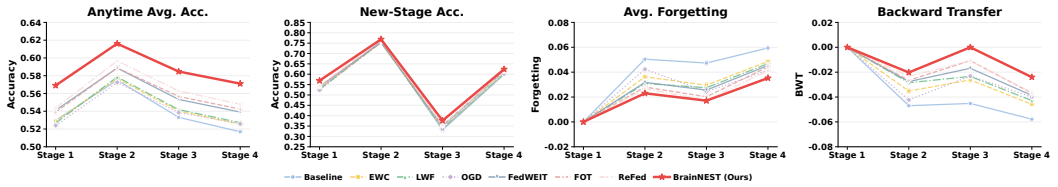


Figure 4: Stage-wise federated continual learning dynamics across the full dataset stream.

maintains the strongest trajectory across stages, rather than improving only at the final checkpoint. The gap becomes more pronounced as the client population changes, suggesting that consensus memory and routed specialization help the model absorb new EEG ecologies while retaining useful structure from earlier stages.

G Broader Impact

BrainNEST is motivated by a broader shift in EEG foundation modeling: useful EEG knowledge is no longer determined only by a single large pretraining corpus, but by how a model is maintained as new datasets, devices, task paradigms, and subject cohorts enter the research ecosystem. By formulating this setting as federated continual learning, our work provides a concrete framework for updating Large EEG Models under realistic data growth without requiring raw EEG pooling. This perspective can help the community move from static model release toward sustainable model maintenance, where new public or institutional datasets can contribute transferable structure while preserving locally specialized information.

The proposed framework also suggests a different way to organize multi-dataset EEG research. Instead of viewing heterogeneous datasets as noise to be averaged away, BrainNEST treats them as a changing ecology in which some structures should be consolidated, some should be routed, and some should remain local. This abstraction can inspire future EEG benchmarks that measure not only single-task accuracy, but also knowledge accumulation, harmful-transfer control, retention of prior structure, and transfer to newly collected datasets. Such evaluation is especially relevant for BCI and computational neuroscience, where data are expensive to collect, institutionally distributed, and frequently tied to changing experimental protocols.

BrainNEST further improves transparency in collaborative EEG learning. The server does not receive raw EEG signals or dataset-specific heads; it receives compact family summaries and family-conditioned exposed updates that make the write decision explicit. This creates a useful interface between local representation learning and server-side model maintenance, making it easier to inspect which evidence supports shared consolidation and which directions are preserved as routed specialization. In the long term, maintenance-oriented modeling may reduce repeated retraining cost, encourage more reusable EEG resources, and support a healthier research ecosystem in which new datasets can improve shared models without requiring permanent data centralization. At the same time, BrainNEST is a research framework rather than a clinical decision system, and any deployment in health-related settings should be accompanied by task-specific validation, governance review, and human oversight.

H Future Work

Future work can extend BrainNEST along several natural directions, including longer asynchronous dataset streams and multimodal neurophysiological settings that combine EEG with related signals such as EOG, EMG, fNIRS, or eye tracking. These directions preserve the central message of this work as Large EEG Models should be maintained as evolving knowledge systems, and federated continual learning offers a practical route for accumulating useful EEG knowledge over time.