ERMoE: Eigen-Reparameterized Mixture-of-Experts for Stable Routing and Interpretable Specialization

Supplementary Material

1. Data and Code Availability

Codes for ERMoE are publicly available at https://github.com/Belis0811/ERMoE. MRI data are from the public cohort ADNI. No relevant accession codes are required to access these data, and the authors had no special access privileges that others would not have to the data obtained from any of these databases.

2. Training Algorithms

2.1. Training ERMoE

We insert ERMoE layers into a ViT backbone by replacing FFNs at fixed blocks and training end-to-end with standard ViT optimization (AdamW, cosine decay), following the patch-embedding and MHSA design in the original ViT paper [10]. Unlike token-choice routers in sparse ViT variants (e.g., V-MoE) that learn free gating logits and then rely on auxiliary balancing losses, our router computes a cosine score in each expert's learned eigenbasis and selects experts via a thresholded top-k rule (Alg. 1). This ties routing decisions directly to an expert's representational subspace and obviates auxiliary load-balancing losses that prior work found necessary for stability in MoE layers [13, 35]. An orthogonality penalty on the expert bases conditions the representation and prevents collapse, and the (rare) fallback path guarantees progress when no score exceeds the threshold.

2.2. Training ERMoE-*ba*

For 3D brain MRI, we adopt a 3D ViT tokenizer (fixed 16^3 cubes) and instantiate a bank of experts that mixes anatomically targeted "region experts" (WM/GM/CSF) with "free experts". Region experts can be warm-started using volumes in which the region of interest is retained (parcellations derived from standard neuroimaging tools, such as FreeSurfer, are a common way to isolate tissue classes). The same eigenbasis score and thresholded top-k gate are used at every MoE block, and a lightweight regression head predicts brain age from the [CLS] token(Alg. 2). This design preserves the content-aware routing of ERMoE while accommodating volumetric context and region-specific specialization.

Our procedures are compatible with common sparse-MoE infrastructure. ERMoE slots into ViT exactly where prior sparse FFN-replacements do (e.g., V-MoE), but removes the need for auxiliary load-balancing terms and router-specific tricks often required for stable training at scale.

3. Visualization for brain router selection

To probe whether ERMoE-*ba* learns anatomically meaningful specializations, we visualize router behavior with *region-isolated* inputs. For each tissue—white matter (WM), gray matter (GM), and cerebrospinal fluid (CSF)—we retain only that region in the volume (visual exemplars in the Appendix) and record the top-2 experts selected by the final MoE layer along with their eigenbasis scores at three training checkpoints (epochs 1, 5, and 300), as shown in Fig. S1. WM/GM/CSF masks are derived from a standard neuroimaging pipeline (FreeSurfer), ensuring that the inputs reflect well-established tissue contrasts.

At early training (epoch 1), selections are diffuse and dominated by free experts with modest eigenbasis scores, consistent with immature expert subspaces. By epoch 300, routing sharpens into tissue-appropriate specializations: `wm_expert` is the primary choice for WM, `gm_expert` dominates GM, and `csf_expert` dominates CSF. Secondary choices (e.g., `wm_expert` for GM at 0.54; `gm_expert` for CSF at 0.47) persist at lower weights, reflecting plausible cross-tissue dependencies rather than collapse. The emergence of such structured, content-aware routing mirrors observations in vision MoEs, where deeper layers exhibit stronger alignment between routing decisions and semantic categories, but here the categories are anatomical tissues rather than object classes.

Methodologically, this region-isolation analysis serves as a targeted perturbation test: by ablating all but one tissue, we assess whether the router's selections remain stable and interpretable under controlled input changes—akin in spirit to occlusion/meaningful-perturbation probes widely used in interpretability. The consolidation of high-confidence selections by epoch 300 indicates that ERMoE's eigenbasis-guided routing learns anatomically grounded expert subspaces rather than relying on spurious correlations.

4. More Results on ADNI

We examined ERMoE-*ba* on the ADNI *train* set to verify that the model learns a stable and well-calibrated mapping from chronological age (CA) to predicted brain age (BA). Figure S2 shows sex-stratified BA–CA scatter plots for males (panel a) and females (panel b). In both groups, points cluster tightly around the identity line (BA = CA), with only modest dispersion across the full age range. The resulting train MAEs are **2.29** years for males and **2.27** years for females, which are close to the corresponding test

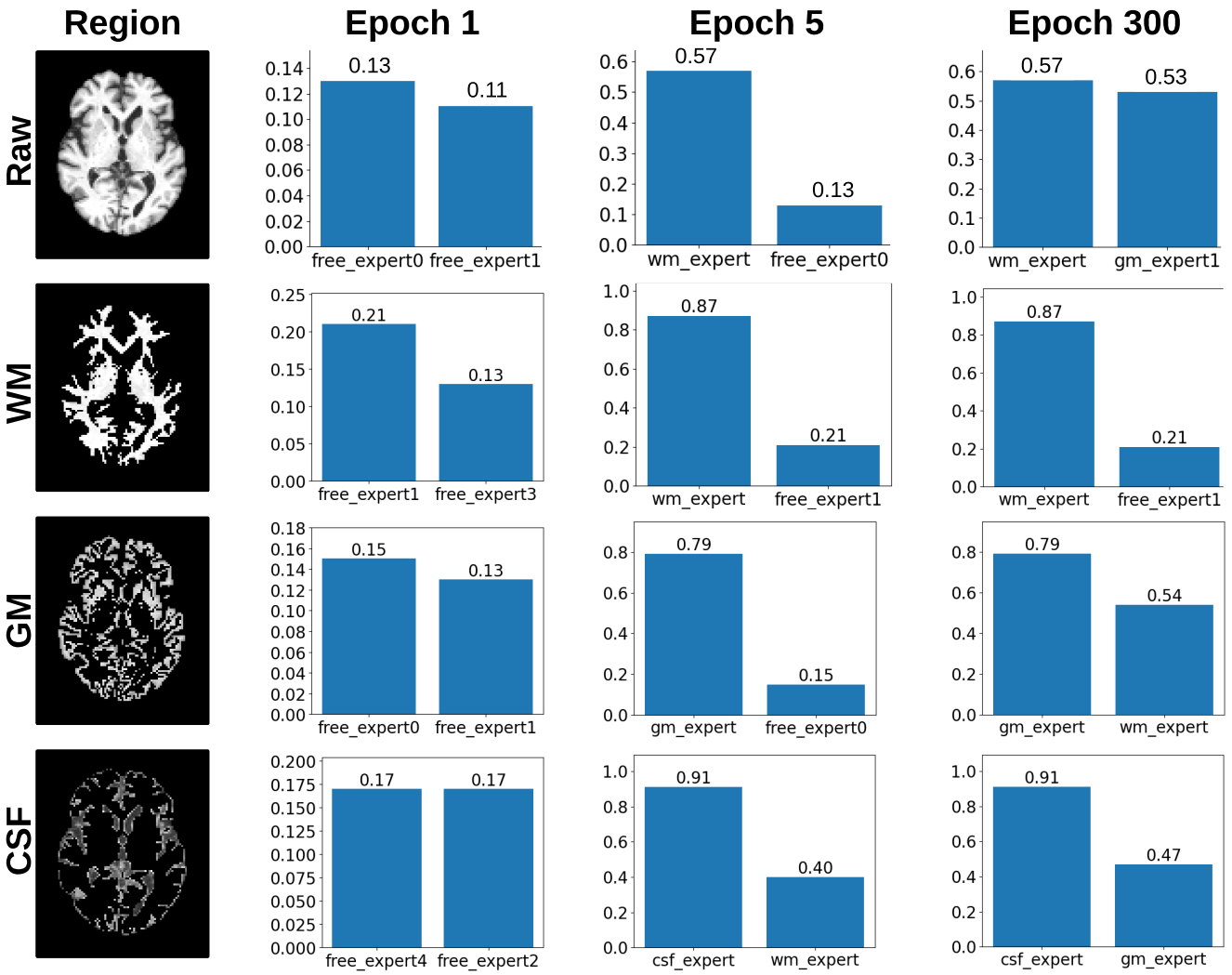

Figure S1. **Router selections on region-isolated brain inputs.** For each region (WM, GM, CSF), we feed a volume in which only that region is retained and log the experts chosen by the final MoE layer over training. Columns show epochs 1, 5, and 300; each cell lists the top–2 selected experts with their eigenbasis scores.

MAEs reported in Table 5.

The near-unit slope and small intercept of the best-fit trends, together with the absence of obvious curvature at younger or older ages, suggest that ERMoE-*ba* attains good calibration on ADNI without relying on aggressive regularization or post-hoc correction. Moreover, the similar spread between male and female panels indicates that the router learns age-sensitive morphometric patterns that generalize across sex, rather than overfitting to sex-specific shortcuts. These training results support that ERMoE-*ba* fits ADNI without evident overfitting and maintains consistent calibration across demographic subgroups.

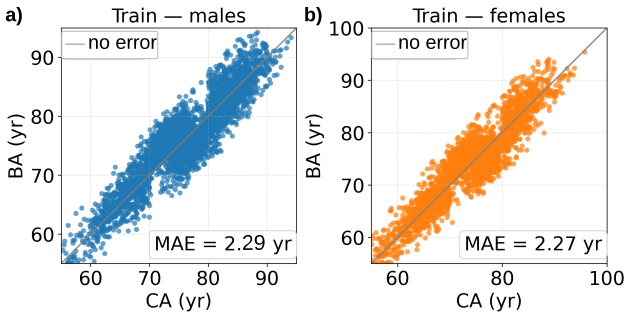

Figure S2. **Brain-age (BA) estimation on ADNI train set.** Scatter plots show ERMoE-*ba* predicted BA versus chronological age (CA) for a) males and b) females on the test set. Points are colored by sex (male: blue; female: orange). The solid diagonal denotes the "no error" line (BA = CA).

5. Post-hoc age-level calibration

Brain-age residuals (BA–CA) typically exhibit age dependence rather than Independent and Identically Distributed(IID), mean-zero Gaussian noise. Following best practice, we evaluate a simple post-hoc linear calibration learned on the training split only: we fit $\hat{y} = a + by$ on the training data and correct test predictions via $\hat{y}_{corr} = (\hat{y} - a)/b$. As shown in Tab. S1, we report MAE, $corr(\hat{y} - y, y)$, and the calibration slope/intercept from regressing $\hat{y}$ on y. This addresses the characteristic "regression-to-the-mean" bend without inflating metrics, aligning with recommendations to report both raw and bias-corrected results, and avoiding over-corrections that artificially boost accuracy.

On the ADNI held-out test set (n=797; M=460, F=337), pooled calibration reduces age-dependence (corr from -0.287 to -0.125) and improves calibration (slope 0.887 to 0.949) with a small MAE gain (2.307 to 2.287 years). Sex-specific calibration yields nearly identical calibration and no further MAE benefit (2.295 y). We therefore report both raw and pooled-calibrated metrics in the main text. Recent age-level bias work further motivates reporting such diagnostics alongside primary accuracy.

Method	MAE	corr	Cal. slope	Cal. intercept
Raw	2.307	−0.287	0.887	9.026
pooled	2.287	−0.125	0.949	4.221
sex-specific	2.295	−0.126	0.949	4.244

Table S1. **Age-level calibration ablation on ADNI test.** Calibration parameters are learned on the training split only (no leakage).

6. Different Thresholds

To show the effectiveness of our router, we ablate the router threshold T that filters experts by eigenbasis score before top-k selection. At test time, we freeze all weights and sweep $T \in [0, 0.9]$ on Tiny-ImageNet (val), counting a fallback whenever no expert satisfies $\max_e \text{score}_e(x) > T$, in which case the router reverts to global top-k. The fallback curve (Fig. S3) remains near zero for $T<0.5$ and rises once $T \geq 0.5$. This shows that when we set a low T, the model will admit low-confidence experts and reintroduce the noisy, content-misaligned assignments characteristic of token-choice routing. This willsignificantly affect accuracy and the utilization of expertsn. When T is too high(higher than 0.5), the router will fall back to global top-k selections, which overloads a small subset of experts and recreates long-tail utilization.. Based on this behavior, we fix $T = 0.5$ for all main results.

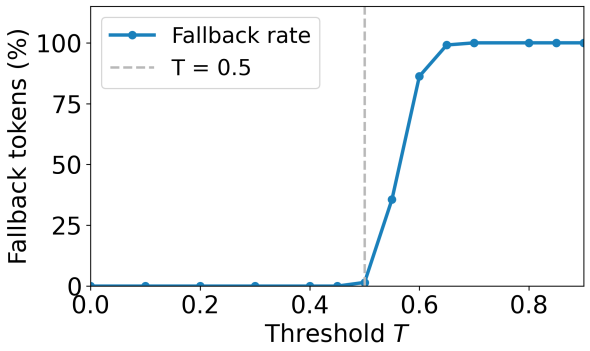

Figure S3. **Threshold T controls fallback in ERMoE (Tiny-ImageNet, test only).** A fallback occurs only when no expert's eigenbasis score exceeds T, in which case the router reverts to global top-k (k=2). The curve stays near zero for $T<0.5$ and rises sharply for $T \geq 0.5$, indicating that $T=0.5$ prunes noisy routes without starving eligibility (lower is better).

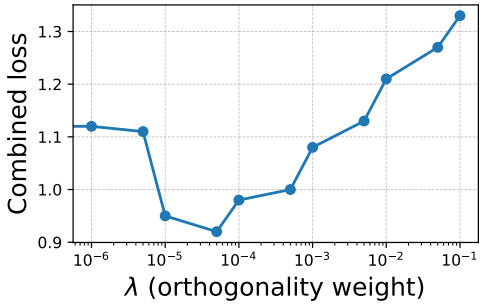

Figure S4. **Effect of the orthogonality weight λ.** Minimum validation loss is achieved at $\lambda = 5 \times 10^{-5}$.

7. Choosing the Orthogonality Weight.

As defined in our loss function (Eq. 10), λ trades off the classification term L_{CLS} and the orthogonality regularizer L_{orth} that encourages the expert bases to remain (near-)orthonormal. Orthogonality/spectral penalties are known to reduce feature redundancy and stabilize optimization in deep networks, but they must be applied with *small* coefficients to avoid overwhelming the task objective [2, 48]. We therefore sweep $\lambda \in [0, 0.01]$ and train for 15 epochs on a 10% ImageNet dataset to locate a robust operating point.

As shown in Fig. S4, the combined validation loss follows a clear U-shape with a minimum at $\lambda = 5 \times 10^{-5}$. For very small values ($\leq 10^{-5}$), the penalty under-regularizes, yielding poorly conditioned bases and slower convergence; for large values ($\geq 10^{-3}$), the penalty dominates and degrades discriminative learning—consistent with prior observations on orthogonality regularization strength. Accordingly, we **fix** $\lambda = 5 \times 10^{-5}$ in all subsequent experiments.

8. Effectiveness of using top-k

We fix the routing threshold at T=0.5 and vary the number of selected experts $k \in \{1, 2, 3, 4, 6\}$ in each ERMoE layer. For each k, we evaluate three quantities on Tiny-ImageNet validation set: (i) task accuracy (Top-1), (ii) routing *tail mass*, and (iii) normalized step time. Tail mass measures how much probability leaks into marginal experts that merely clear the threshold and would be pruned by a smaller k. Formally, with scores $\{s_e\}$, set $A = \{e : s_e > T\}$ and let $S_k \subset A$ index the top-k experts by score; with mixture weights $w_e \propto s_e$ renormalized on A,

$$\text{TailMass} = \frac{\sum_{e \in A \setminus S_k} w_e}{\sum_{e \in A} w_e}. \tag{12}$$

Figure S5 shows a consistent pattern. **(1)** k=1 **underselects capacity.** Accuracy drops because many tokens that meet the threshold still benefit from a secondary expert; restricting to a single expert sacrifices complementary subspaces. **(2)** k=2 **is the sweet spot.** Accuracy peaks while tail mass remains low, indicating that most of the useful probability mass concentrates on two high-confidence experts; latency remains near the sparse MoE budget. **(3)** k>2 **dilutes predictions and slows inference.** As k grows, more above-threshold but low-score experts enter the mixture, tail mass increases monotonically, and Top-1 degrades due to averaging in weak directions; step time rises roughly linearly with the number of active experts per token. In short, given a fixed content threshold T, *top-k is necessary* to cap marginal contributors, preserve sparsity, and avoid accuracy loss from mixture dilution. This finding aligns with observations in the sparse MoE literature that controlling the number of active experts is critical for both computational efficiency and quality.

9. Training Efficiency.

Another key aspect for MoE architectures is scaling model capacity while controlling computational cost. We benchmark efficiency at 224^2 with ViT–B/16 and E=8 experts. As shown in table S2, compared to V-MoE, **ERMoE** reduces computation while keeping the same backbone and gating budget: FLOPs dropped **10.3%**, and inference time falls from 4.4,ms to **4.1,ms**. Total trainable parameters shrink **14.7%**, and active parameters per sample showed **12.9%** reduction. The "active parameters" notion is standard in sparse MoE: only the routed experts participate in a given forward/backward pass, so reductions here translate directly into lower per-step math and memory traffic.

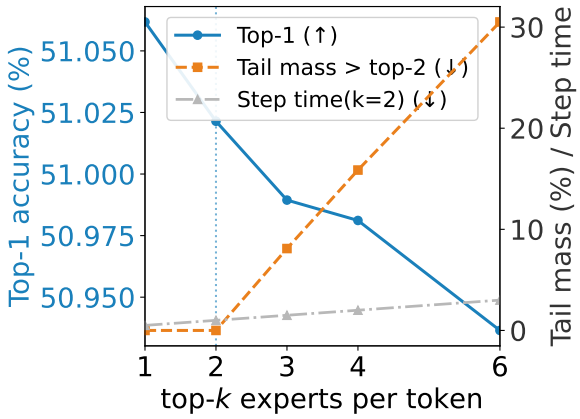

Figure S5. **Top-k ablation at fixed threshold T=0.5 on Tiny-ImageNet (val).** We vary the number of selected experts per token $k \in \{1, 2, 3, 4, 6\}$ and report Top-1 (left axis), routing tail mass from Eq. (12) (right axis), and normalized step time. k=1 underutilizes capacity; k=2 yields the best accuracy with a low tail mass and moderate cost; k>2 increases tail mass and latency while reducing accuracy due to mixture dilution by low-scoring, above-threshold experts.

Table S2. **Compute and efficiency comparison.** "Params (M)" shows total parameters; "(act)" shows per-sample *active* parameters. FLOPs are for 224×224. V-MoE numbers are for ViT-B/16 with 8 experts, top-2 gating every 2 blocks. Single-gated MoE and DeepMoE report on ResNet-18 (largest reported CNN settings).

Model	Params (M)	FLOPs (G)	Inference
Single-gated MoE	n/r	2.18	n/r
DeepMoE	n/r	1.81	n/r
V-MoE	284.9 (129.9 act)	26.3	4.4 ms
ERMoE	243.0 (113.2 act)	23.6	4.1 ms

These results confirms that ERMoE achieved superior accuracy without introducing any more trainable parameters or consuming more computational resources.

10. More ablations.

To address this directly, we added component ablations on CIFAR-10 5-shot classification, keeping experts(E), top-k, and T fixed to default settings. We report 3 variants (i)with standard expert weights, (ii)learned routing logits, and (iii)no orthogonality($\lambda = 0$). Tab.S3 shows that replacing eigenbasis experts with standard weights drops accuracy by 2%, indicating the parameterization is more than compression and strengthens the intended router–expert subspace coupling. Replacing cosine eigenbasis scoring yields the largest drop (5%) and a much higher CV^2(squared coefficient of variation), proving that token-choice learned routers often needing auxiliary balancing to avoid heavy-tail

expert usage. Setting λ=0 has little effect on accuracy but increases expert subspace overlap(inclined CV^2 and Fallback%).

Variant	Top-1 (%) ↑	Load CV^2 ↓	Fallback (%) ↓
normal weights	94.12	0.10	0.3
learned-logit router	91.07	0.32	2.4
λ=0	95.35	0.15	0.7
ERMoE	**96.05**	**0.08**	**0.2**

Table S3. **Component ablations.** Load CV^2 computed over per-expert token counts. Fallback compute same as Appendix Sec.6

Overall, the ablations show that cosine alignment is the main source of the gains; eigen-parameterized experts add a consistent improvement; orthogonality mostly improves stability rather than being the sole driver of accuracy.

11. Routing Overhead and Practical Runtime

ERMoE routing computes an Eigenbasis Score by projecting each token feature (and its attention-weighted context) into each expert basis and then taking cosine similarity. Compared to a token-choice router that computes logits using a single linear map, this adds routing arithmetic. Table S4 reports routing-step FLOPs and routing-step wall-clock.

For scaling, the routing compute is linear in the number of experts:

$$\text{routing FLOPs} = O(B\,E\,d\,r), \tag{13}$$

where B is the number of tokens, E is the number of experts, d is feature dimension, and r is the basis rank. The per-expert routing-state memory is also linear in E:

$$\text{routing-state memory} = O(E\,d\,r^2). \tag{14}$$

Orthogonality is enforced mainly via L_{ortho} during training, and we apply a lightweight re-orthonormalization to each expert basis once per epoch; the amortized cost is $O(E\,d\,r^2)$ per epoch.

12. Scalability With Expert Count

To provide scaling evidence under limited compute, we run a one-layer stress test at E=8 vs. E=32 and measure allocated GPU memory for one epoch under the same setting. ERMoE uses 20.5GB (E=8) and 84.7GB (E=32), while a standard MoE uses 18.0GB and 80.0GB, respectively. Thus, ERMoE adds 2.2GB while increasing E by $4\times$.

Since ERMoE does not perform any online $d \times d$ eigendecomposition, the dominant large-E bottleneck remains sparse MoE execution (token dispatch/gather and expert compute), rather than ERMoE scoring.

13. Additional Baseline: Expert-Choice

We add an Expert-Choice baseline on ImageNet under the same experimental settings. Expert-Choice achieves 78.95% top-1 accuracy, while ERMoE reaches 88.03% top-1 accuracy.

14. Multimodal Setting: Fine-tuning the Text Tower

We fine-tune the text tower in Fix-CLIP and evaluate text-to-image retrieval on COCO. The results are 50.3% R@1, 74.0% R@5, and 82.6% R@10, improving over the frozen-text Fix-CLIP setting (49.1%, 73.8%, and 82.4%). These results show that joint tuning of the text encoder can improve T2I retrieval, while our frozen-text experiments serve as a controlled setting to isolate the image encoder's contribution.

Algorithm 1: Training ERMoE (thresholded top-k with eigenbasis-aligned routing)

Input : Dataset $\mathcal{D} = \{(x, y)\}$; ViT backbone with M MoE blocks; experts per block E

Hyperparams: top-k; threshold T; orthogonality weight λ; optimizer $\mathcal{O}$

Output : Trained parameters Θ (backbone + expert bases/coeffs)

Initialize: For each expert $e \in \{1, \dots, E\}$, initialize an orthonormal basis B_e (columns) and expert parameters Φ_e.

for *each minibatch* $\mathcal{B} \subset \mathcal{D}$ **do**

  Tokenize every image $x \in \mathcal{B}$ into patches; embed to tokens $\{t_i \in \mathbb{R}^d\}$ with positional encodings.

  for $\ell \leftarrow 1$ **to** M **do**

    Apply MHSA and residual connections (ViT block) to obtain token features $\{z_i\}$ and attention-weighted contexts $\{\bar{c}_i\}$.

    `// ERMoE routing within block` ℓ

    for *each token* z_i *(and its context* $\bar{c}_i$*)* **do**

      for *each expert* $e \in \{1, \dots, E\}$ **do**

        $\tilde{x}_i \leftarrow x_i / \|x_i\|_2$; $\tilde{c}_i \leftarrow \bar{c}_i / \|\bar{c}_i\|_2$ `// Normalize`

        $u_i^{(e)} \leftarrow B_e^\top \tilde{x}_i$ `// Project token`

        $v_i^{(e)} \leftarrow B_e^\top \tilde{c}_i$ `// Project context`

        $s_{i,e} \leftarrow \cos\big(u_i^{(e)}, v_i^{(e)}\big)$ `// eigenbasis score`

      $\mathcal{E}_i \leftarrow \{\, e \mid s_{i,e} > T \,\}$

      if $|\mathcal{E}_i| \geq k$ **then**

        $\mathcal{S}_i \leftarrow$ top-k experts in $\mathcal{E}_i$ by $s_{i,e}$

      else

        $\mathcal{S}_i \leftarrow$ top-k experts over $\{1, \dots, E\}$ by $s_{i,e}$ `// fallback if no/too few exceed` T

      Set $w_{i,e} \propto s_{i,e}$ for $e \in \mathcal{S}_i$; set $w_{i,e} = 0$ otherwise; normalize $\sum_e w_{i,e} = 1$.

      Compute expert outputs $h_{i,e} \leftarrow \mathrm{Expert}_e(z_i; \Phi_e)$ for $e \in \mathcal{S}_i$.

      Combine $y_i \leftarrow \sum_{e \in \mathcal{S}_i} w_{i,e}\, h_{i,e}$.

    Apply residual and normalization to form the block output.

  Task loss: L_{task} (e.g., cross-entropy).

  Orthogonality penalty: $L_{\text{orth}} = \sum_e \big\| B_e^\top B_e - I \big\|_F^2$.

  Total loss $L \leftarrow L_{\text{task}} + \lambda L_{\text{orth}}$.

  Update Θ using optimizer $\mathcal{O}$.

Algorithm 2: Training ERMoE-*ba* for brain-age prediction (3D ViT with region & free experts)

Input : T1 volumes $\mathcal{D} = \{(V, \text{age})\}$; 3D ViT backbone with M MoE blocks

Hyperparams: experts per block E; top-k; threshold T; orthogonality weight λ; BA head $g(\cdot)$; optimizer $\mathcal{O}$

Output : Trained parameters Θ_{3D}

Expert sets: region experts $\mathcal{R} = \{\texttt{wm}, \texttt{gm}, \texttt{csf}\}$ (optionally warm-started on region-isolated inputs); free experts $\mathcal{F}$ (random init). Initialize (B_e, Φ_e) for all $e \in \mathcal{R} \cup \mathcal{F}$.

for *each minibatch* $\mathcal{B} \subset \mathcal{D}$ **do**

  Tokenize each V into non-overlapping 16×16×16 cubes; embed tokens $\{t_i \in \mathbb{R}^d\}$ with 3D positional encodings.

  for $\ell \leftarrow 1$ **to** M **do**

    Apply 3D MHSA and residual connections to obtain $\{z_i\}$.

    for *each token* z_i **do**

      for *each expert* $e \in \{1, \dots, E\}$ **do**

        $\hat{u}_{i,e} \leftarrow \mathrm{norm}(B_e^\top z_i)$; $\hat{v}_e \leftarrow \mathrm{norm}(\psi_e)$; $s_{i,e} \leftarrow \cos\big(\hat{u}_{i,e}, \hat{v}_e\big)$.

      $\mathcal{E}_i \leftarrow \{e \mid s_{i,e} > T\}$; $\mathcal{S}_i \leftarrow$ top-$k(\mathcal{E}_i)$ if $|\mathcal{E}_i| \geq k$, else top-k overall.

      $w_{i,e} \propto s_{i,e}$ for $e \in \mathcal{S}_i$; normalize $\sum_e w_{i,e} = 1$.

      $h_{i,e} \leftarrow \mathrm{Expert}_e(z_i; \Phi_e)$ for $e \in \mathcal{S}_i$; $y_i \leftarrow \sum_{e \in \mathcal{S}_i} w_{i,e}\, h_{i,e}$.

  Brain-age head: $\hat{\text{BA}} \leftarrow g\big(\texttt{[CLS]}(\{y_i\})\big)$.

  Task loss: $L_{\text{MAE}} = |\hat{\text{BA}} - \text{age}|$.

  Orthogonality penalty: $L_{\text{orth}} = \sum_e \|B_e^\top B_e - I\|_F^2$.

  Total loss $L \leftarrow L_{\text{MAE}} + \lambda L_{\text{orth}}$; update Θ_{3D} with $\mathcal{O}$.

Routing	FLOPs (G)	Wall-clock (ms)
V-MoE	0.631	0.03
ERMoE	1.169	0.05

Table S4. **Routing cost.** V-MoE denotes a standard vision MoE routing baseline.