MENDR: Manifold-Embedded Neural Data Representations for Channel-Agnostic EEG Foundation Modeling

Matthew Chen* **Micky C. Nnamdi*** **Benoit Louis Marteau** **Shaun Qien Yeau Tan**
J. Ben Tamo **May Dongmei Wang†**
Georgia Institute of Technology, Atlanta, GA, USA
maywang@gatech.edu

Abstract

Foundation models for EEG are pretrained on large unlabeled recordings so that one model can be adapted to many tasks, but most treat EEG as a generic time series. They do not model the covariance between electrodes explicitly, which measures how strongly each pair of signals rises and falls together. Covariance matrices are symmetric positive definite (SPD) and lie in a curved (Riemannian) space, where ordinary averages and distances are distorted. We introduce MENDR (Manifold-Embedded Neural Data Representations)‡, an EEG foundation model whose tokens are covariance matrices of short windows. A matrix logarithm flattens each matrix (Log-Euclidean tangent space), so a standard transformer can be pretrained to fill in masked tokens. A channel-agnostic Perceiver-style projection maps each dataset's electrodes to 32 learned virtual electrodes, so one pretrained model can be fine-tuned on 6 to 64 channels (one layout per fine-tuned model). MENDR's spatial encoder is 2.6 to 21 times smaller than the backbones of the compared models. Pretrained on the TUH EEG Corpus and evaluated on six benchmarks under a strict protocol, MENDR matches the best compared model on seizure and abnormality detection (no significant difference in AUROC) and is 0.03 to 0.06 weighted F1 behind it on the other four tasks. Controlled experiments show that the channel-agnostic projection beats a fixed linear one on two of six tasks (one confounded by electrode count), that pretraining beats training from scratch on one, and that larger encoders bring no significant gain.

1 Introduction

Electroencephalography (EEG) records the brain's electrical activity through scalp electrodes [7]. It is a routine clinical test in epilepsy, sleep medicine and intensive care, and also supports motor imagery decoding [2], emotion recognition [22] and brain-computer interfaces [13]. Foundation models are pretrained on large unlabeled EEG corpora so that one model can be fine-tuned for many such tasks [39, 18, 37, 36]. Most of them treat EEG as a generic multichannel time series and reuse architectures designed for text or images. This ignores the covariance matrix of the electrode signals over a short window, which records how strongly each pair of electrodes varies together and underlies classical decoders that remain competitive in brain-computer interfaces [6, 8]. Covariance matrices are symmetric positive definite (SPD) and form a curved space, not a vector space: averaging them

*Equal contribution.
†Corresponding author.
‡Code and pretrained weights: https://github.com/MatthewChen37/MENDR-Manifold-Embedded-Neural-Data-Representations

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

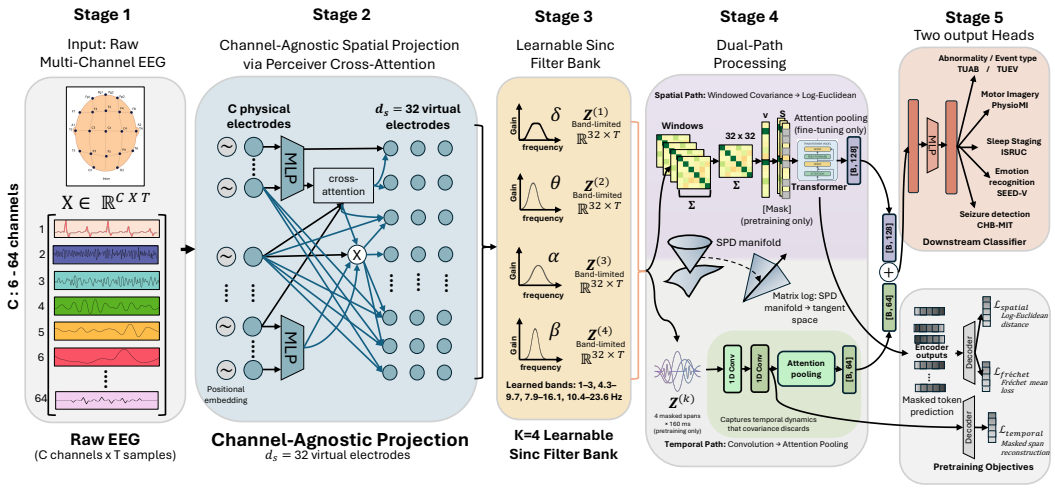


Figure 1: MENDR architecture. Raw EEG (6–64 channels) is mapped to 32 virtual electrodes by Perceiver cross-attention, split into $K=4$ learnable bands, and processed by a *spatial* path (Log-Euclidean embedding of windowed SPD covariances) and a *temporal* path (1D convolutions with attention pooling). Pretraining reconstructs masked SPD tokens and masked temporal spans, and fine-tuning concatenates the outputs of both paths. v: one tangent-space token $\mathbf{v}$ (Eq. (7)), S: the token sequence $\mathbf{S}$ over windows and bands.

element-wise, for example, can inflate their determinant [4, 29]. The matrix logarithm maps each SPD matrix to an ordinary symmetric matrix, where Euclidean distances and averages become the Log-Euclidean ones of the curved space [4]. Networks using this geometry exist for supervised EEG decoding [14, 27, 42] and for self-supervised learning on one clinical population [38], but the widely used general-purpose EEG foundation models do not use it.

A second problem is the electrode layout (the montage). Datasets use 6 to 256 electrodes in different positions and orders, and a pretrained model must know which input comes from which electrode. LaBraM [18] and EEGPT [36] learn an embedding per electrode name of the 10–20 system, BIOT [39] one per channel index, and CBraMod [37] applies a convolution along the channel order. How much of the pretrained model transfers therefore depends on how closely a new montage matches the pretraining one. Models that identify electrodes by their scalp coordinates are only beginning to appear [9].

Third, these models range from 3.2M (BIOT) to 369M (LaBraM-Huge) parameters, and part of this capacity may be spent learning structure that the architecture could supply directly. A small model is also easier to store and share on hospital workstations, where patient data cannot leave the site, although it is not necessarily faster (Appendix M).

We present **MENDR** (*Manifold-Embedded Neural Data Representations*), a small EEG foundation model that builds covariance geometry into the architecture instead of learning it from data (Figure 1). MENDR mixes the input channels into 32 learned virtual electrodes, splits them into four learned frequency bands and computes a covariance matrix in each short window. The matrix logarithm turns each matrix into a token, and a transformer is pretrained to reconstruct masked tokens on the Temple University Hospital (TUH) EEG Corpus [26] (14,559 subjects, about 13,300 hours). Under a strict evaluation protocol, MENDR is not significantly different from the best compared model on seizure and abnormality detection, and is behind it by 0.03 to 0.06 on the other four tasks (Section 4.2). Our contributions are:

- **A Riemannian masked autoencoder for EEG.** We combine SPD covariance tokens, the Log-Euclidean map and masked-autoencoder pretraining in one general-purpose EEG foundation model. It learns to reconstruct masked tokens under the squared Log-Euclidean distance and, as an auxiliary task, to predict their mean (the Fréchet mean). Each ingredient exists (Section 2), and our contribution is their combination in one pretrained model.

- **A channel-agnostic spatial projection.** As in LUNA [9], a Perceiver-style cross-attention module maps the electrodes to 32 learned virtual electrodes, so one pretrained encoder can be fine-tuned on 6 to 64 channels. It identifies channels by index, so each fine-tuned model is tied to its montage (Section 4.6).
- **A small model, and a test of where its accuracy comes from.** MENDR's spatial encoder has **1.2M parameters**, 2.6 to 21 times fewer than the backbones of the compared models. In controlled experiments, the channel-agnostic projection trained from scratch is ahead of a fixed linear one on two of six tasks, and pretraining is ahead of training from scratch on one. The spatial path alone is not significantly different from the full model, larger encoders (up to 23M parameters) are not significantly better, and reordering or removing test channels lowers accuracy.

2 Related Work

EEG foundation models. Recent self-supervised pretraining on large EEG corpora has produced models such as BENDR [21] (Wav2Vec-style contrastive), BIOT [39] (short-time Fourier transform (STFT) tokens, linear transformer), LaBraM [18] (vector-quantized tokenizer + masked modeling, scaling to 369M params on 2,500 hours), CBraMod [37] (criss-cross spatial and temporal attention, time- and frequency-domain patch embedding), and EEGPT [36] (masked reconstruction with representation alignment). Other brain-signal foundation models include Brant [41], BrainWave [40], NeuroLM [17], SleepFM [34], and Beatrix [43]. These models use signal- or spectrum-domain tokens, not SPD covariance tokens.

Riemannian geometry for EEG. Log-Euclidean [4, 5] and affine-invariant [29] classifiers on EEG covariance matrices remain competitive with deep learning in brain-computer interfaces [6, 8]. SPDNet [14] and follow-ups [27, 19] introduced learnable layers on the SPD manifold for *supervised* classification. Zhang & Etemad [42] instead learn on tangent vectors and evaluate on four datasets. Tibermacine et al. [35] review Riemannian and SPD methods for EEG. SPD Learn [3] provides such layers as a PyTorch library. In self-supervised learning, EEG-ReMinD [38] already combines masked reconstruction of covariance-derived SPD states with the Log-Euclidean distance and a weighted Log-Euclidean mean. It is pretrained separately on each of the two neurodegenerative-disease datasets. MAtt [27] uses the same distance and the closed-form Log-Euclidean mean, whose unweighted form our Fréchet objective uses. Both are trained per dataset (EEG-ReMinD self-supervised, MAtt supervised), whereas MENDR pretrains on a large clinical corpus and transfers to six benchmarks.

Channel-agnostic EEG models. Like LUNA [9], MENDR maps a variable set of electrodes to a fixed set of learned queries by cross-attention, but it mixes raw signals, not LUNA's learned features. MENDR's learned queries define 32 virtual electrodes, each a learned weighted combination of the physical electrodes. Every montage therefore yields covariance matrices of the same size, and one filter bank and SPD transformer serve all montages. LUNA identifies channels by electrode coordinates, making it independent of channel order. In contrast, MENDR's projection identifies channels by index, so a fine-tuned model expects the montage it was trained on (Section 4.6).

Frequency decomposition and masked modeling. EEG analysis has long relied on canonical bands ($\delta, \theta, \alpha, \beta, \gamma$) [1], and most foundation models use STFT [39, 37, 18] or wavelet [28] features. MENDR instead learns its bands with a sinc filter bank [31]. Among masked autoencoders [12, 25, 18, 36], MENDR differs in its tokens (Log-Euclidean tangent vectors of SPD covariances rather than signal patches) and in its token loss (an ℓ_2 loss in the tangent space rather than in signal space).

3 Methodology

Given a multi-channel EEG signal $\mathbf{X} \in \mathbb{R}^{C \times T}$ (C electrodes, T samples at 200 Hz), MENDR (i) projects the channels onto 32 virtual electrodes via cross-attention, (ii) decomposes the signal into learnable frequency bands, (iii) computes windowed covariance matrices on the SPD manifold, (iv) maps them to the Log-Euclidean tangent space via matrix logarithm, and (v) learns contextualized representations with a masked transformer (Figure 1).

3.1 Channel-Agnostic Spatial Projection

Instead of an electrode-name vocabulary (as in LaBraM and EEGPT, Section 1), a Perceiver-style [16] cross-attention module maps the C physical electrodes of any montage to a fixed set of d_s=32 *virtual electrodes*. A small multilayer perceptron (MLP) computes a feature vector for each electrode c from its temporal mean $\bar{x}_c$ and a learned embedding $\mathbf{p}_c \in \mathbb{R}^{d_k}$ (d_k=64) of its index c (row c of a 128-row learned table), so no electrode names are needed:

$$\mathbf{k}_c = \mathrm{MLP}\big([\bar{x}_c;\, \mathbf{p}_c]\big), \qquad \mathbf{K} = [\mathbf{k}_1, \ldots, \mathbf{k}_C]^\top \in \mathbb{R}^{C \times d_k}. \tag{1}$$

Learned queries $\mathbf{Q} \in \mathbb{R}^{d_s \times d_k}$, one per virtual electrode, attend to these features through standard multi-head attention (MHA, 4 heads, with learned query, key, value and output projections). Keys and values both come from $\mathbf{K}$. A residual connection and a LayerNorm follow the attention:

$$\mathbf{H} = \mathrm{LayerNorm}\big(\mathbf{Q} + \mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{K})\big) \in \mathbb{R}^{d_s \times d_k}. \tag{2}$$

Row i of $\mathbf{H}$, the updated query of virtual electrode i, is used only to score the channels. The scores give the mixing weights $\mathbf{A}$, which are applied to the signal $\mathbf{X} \in \mathbb{R}^{C \times T}$:

$$\mathbf{A} = \mathrm{softmax}\big(\mathbf{H}\mathbf{K}^\top / \sqrt{d_k}\big) \in \mathbb{R}^{d_s \times C}, \qquad \mathbf{Z} = \mathrm{LayerNorm}(\mathbf{A}\mathbf{X}) \in \mathbb{R}^{d_s \times T}, \tag{3}$$

where the softmax runs over channels and the LayerNorm over the d_s virtual electrodes at each time step. A_{ic} is the weight of channel c in virtual electrode i, and $\mathbf{Z}$, the d_s virtual-electrode signals, is the filter bank input (Section 3.2). No parameter depends on C (only the sizes of $\mathbf{K}$ and $\mathbf{A}$ do), so one model accepts any number of channels. Channels are identified by their index c, so the projection depends on channel order (Section 4.6). The LayerNorm subtracts the mean over virtual electrodes, so $\mathbf{Z}$ depends only on how the rows of $\mathbf{A}$ differ from their mean row $\bar{\mathbf{a}}$, i.e., on $\mathbf{D}\mathbf{X}$ with $\mathbf{D} = \mathbf{A} - \mathbf{1}\bar{\mathbf{a}}^\top$ ($\mathbf{1}$: the all-ones vector). Each row of $\mathbf{A}$ sums to one, so each row of $\mathbf{D}$ sums to zero: every virtual electrode is a spatial contrast between channels. A signal added equally to all channels therefore cancels. Re-referencing to the common average is such a change (it also shifts the channel means behind $\mathbf{A}$), and it changes the TUH Abnormal (TUAB) score by less than 0.001 (Appendix K.1).

3.2 Learnable Frequency Decomposition

Instead of fixed bands, a sinc filter bank [31] learns K=4 bandpass filters. Each filter k has two learnable scalars, $\tilde{f}_k$ and Δf_k, which set its cutoffs and impulse response:

$$\begin{aligned} h_k[n] &= \big[2f_k^{\mathrm{hi}}\,\mathrm{sinc}(f_k^{\mathrm{hi}} n) - 2f_k^{\mathrm{lo}}\,\mathrm{sinc}(f_k^{\mathrm{lo}} n)\big] \cdot \omega[n], \\ f_k^{\mathrm{lo}} &= f_{\min} + |\tilde{f}_k|, \qquad f_k^{\mathrm{hi}} = f_k^{\mathrm{lo}} + \Delta f_{\min} + |\Delta f_k|, \end{aligned} \tag{4}$$

where $\omega[n]$ is a Hamming window of length 65, $\mathrm{sinc}(fn) \equiv \sin(2\pi fn/f_s)/(2\pi fn/f_s)$ with sampling rate f_s=200 Hz, $f_{\min}$=1 Hz is a lower bound on the cutoff, and $\Delta f_{\min}$=2 Hz enforces a minimum bandwidth. Each filter is ℓ_1-normalized and applied to every channel of $\mathbf{Z}$ by 1D convolution. The filters start near the δ, θ, α and β bands, with initial passbands of 2–7, 5–11, 9–16, and 14–33 Hz. To prevent filter collapse, a diversity regularizer $\mathcal{L}_{\mathrm{div}} = \frac{1}{K(K-1)} \sum_{i \neq j} \max(0,\, f_{\mathrm{sep}} - |\nu_i - \nu_j|)$, used during pretraining only, penalizes filters whose center frequencies ν_k are closer than f_{sep}=5 Hz (filter trajectories: Appendix F). The output is K band-limited signals $\{\mathbf{Z}^{(k)}\}_{k=1}^{K}$, each in $\mathbb{R}^{d_s \times T}$.

3.3 SPD Manifold Embedding

Windowed covariance. For each frequency band k, we partition $\mathbf{Z}^{(k)}$ into overlapping windows of w=64 samples with stride s=32 by default (per-task values: Section 4.1). Within each window j, we compute the sample covariance matrix, trace-normalize, and regularize to ensure strict positive definiteness:

$$\mathbf{\Sigma}_j^{(k)} = \frac{\tilde{\mathbf{Z}}_j^{(k)} \big(\tilde{\mathbf{Z}}_j^{(k)}\big)^\top}{w - 1}, \qquad \hat{\mathbf{\Sigma}}_j^{(k)} = \frac{\mathbf{\Sigma}_j^{(k)}}{\mathrm{tr}(\mathbf{\Sigma}_j^{(k)})} + \epsilon\, \mathbf{I}_{d_s}, \tag{5}$$

where $\tilde{\mathbf{Z}}_j^{(k)}$ is the mean-centered window and ϵ=10^{-3}. The trace normalization makes the token invariant to the overall signal scale (the temporal path and projection are not; Table 16). The eigendecomposition below is computed in double precision. This produces $W \approx T/s$ SPD matrices $\hat{\mathbf{\Sigma}}_j^{(k)} \in \mathcal{S}_{++}^{d_s}$ per band.

Log-Euclidean embedding. We map each $\hat{\mathbf{\Sigma}} \in \mathcal{S}_{++}^{d_s}$ to the tangent space at the identity by the matrix logarithm, computed through the eigendecomposition $\hat{\mathbf{\Sigma}} = \mathbf{U}\ \mathrm{diag}(\lambda_1, \ldots, \lambda_{d_s})\ \mathbf{U}^\top$:

$$\mathrm{Log}(\hat{\mathbf{\Sigma}}) = \mathbf{U}\ \mathrm{diag}\big(\log \lambda_1, \ldots, \log \lambda_{d_s}\big)\ \mathbf{U}^\top. \tag{6}$$

We implement a numerically stable backward pass using the Daleckii–Kreĭn formula [15]. In the eigenbasis, we multiply the incoming gradient element-wise by $\Gamma_{ij} = (\log \lambda_i - \log \lambda_j)/(\lambda_i - \lambda_j)$, with $\Gamma_{ii} = 1/\lambda_i$. For near-equal eigenvalues, Γ_{ij} is replaced by its limit $1/\lambda_i$ (Appendix A).

The resulting symmetric matrix $\mathbf{L} = \mathrm{Log}(\hat{\mathbf{\Sigma}})$ is isometrically vectorized [4]. The half-vectorization vech stacks its upper triangle (its $d_s(d_s+1)/2$ entries on and above the diagonal) into a vector, and $\mathrm{vech}_{\sqrt{2}}$ also multiplies each off-diagonal entry by $\sqrt{2}$:

$$\begin{aligned} \mathbf{v} &= \mathrm{vech}_{\sqrt{2}}(\mathbf{L}) = \big(L_{11},\ \sqrt{2}L_{12},\ \ldots,\ \sqrt{2}L_{1d_s},\ L_{22},\ \ldots,\ L_{d_sd_s}\big) \in \mathbb{R}^{d_v}, \\ d_v &= \tfrac{d_s(d_s+1)}{2} = 528. \end{aligned} \tag{7}$$

Each off-diagonal entry stands for two equal entries of $\mathbf{L}$, so the $\sqrt{2}$ factor makes $\|\mathbf{v}\|_2 = \|\mathbf{L}\|_F$. The ℓ_2 distance between two tangent vectors therefore equals the Log-Euclidean geodesic distance on $\mathcal{S}_{++}^{d_s}$ (Appendix B). Means and ℓ_2 losses computed on these vectors are thus Log-Euclidean means and distances, without requiring custom manifold-valued layers. The token sequence is formed by concatenating the vectors across all bands, $\mathbf{S} = [\mathbf{v}_1^{(1)}, \ldots, \mathbf{v}_W^{(1)}, \ldots, \mathbf{v}_W^{(K)}] \in \mathbb{R}^{P \times d_v}$, where $P = KW$.

3.4 Riemannian Masked Autoencoder

Following the masked-autoencoder paradigm [12], we randomly mask a fraction ρ=0.5 of the token sequence $\mathbf{S}$ and train the model to reconstruct the masked tokens in the Log-Euclidean tangent space.

Encoder. Tokens are projected from d_v=528 to d_m=128 via a linear layer and a LayerNorm, summed with learnable positional embeddings, and processed by a transformer encoder with L=4 post-norm layers, H=4 attention heads, and feedforward dimension $4d_m$=512.

Decoder and spatial loss. A learnable mask token $\mathbf{m} \in \mathbb{R}^{d_v}$ replaces each masked token before the input projection, preserving sequence order. The transformer encoder processes the full sequence, and a two-layer MLP decoder ($d_m \to d_m \to d_v$, with Gaussian error linear unit (GELU) activation and layer normalization) predicts the original tangent vectors at masked positions. There is no separate decoder transformer.

The spatial loss measures reconstruction fidelity in the Log-Euclidean metric:

$$\mathcal{L}_{\text{spatial}} = \frac{1}{|\mathcal{M}|\, d_v} \sum_{j \in \mathcal{M}} \|\hat{\mathbf{v}}_j - \mathbf{v}_j\|_2^2, \tag{8}$$

where $\mathcal{M}$ is the set of masked indices.

Fréchet mean loss. As an auxiliary objective, we predict the Fréchet mean of the masked tokens from the encoder outputs at the unmasked positions:

$$\mathcal{L}_{\text{frechet}} = \frac{1}{d_v} \big\|\hat{\boldsymbol{\mu}} - \bar{\mathbf{v}}_{\mathcal{M}}\big\|_2^2, \quad \bar{\mathbf{v}}_{\mathcal{M}} = \frac{1}{|\mathcal{M}|} \sum_{j \in \mathcal{M}} \mathbf{v}_j, \tag{9}$$

where $\hat{\boldsymbol{\mu}}$ is produced by a head applied to the average of those encoder outputs. Under the Log-Euclidean metric, the Fréchet mean is the arithmetic mean of tangent vectors (Appendix B).

Temporal path. In parallel, a temporal convolutional network (two 1D convolution layers with kernel 15, d_t=64, with attentive pooling) processes each band signal $\mathbf{Z}^{(k)}$ directly and is designed to capture fine-grained temporal dynamics that the covariance bottleneck discards. The same network serves all bands. To pretrain it, we zero out four non-overlapping 160 ms spans at random positions of each band signal $\mathbf{Z}^{(k)}$ as it enters the temporal path (the spatial path sees it unmasked). A two-layer convolutional decoder used only in pretraining predicts every sample from the per-time-step convolution features, so the attentive pooling is not pretrained. The loss $\mathcal{L}_{\text{temporal}}$ is the mean squared error on the masked samples, with the target $\mathbf{Z}^{(k)}$ standardized per channel over time. Gradients are stopped at the band signals, so this loss does not train the projection, filter bank, or spatial path. Appendix J.1 reports its reconstruction error.

3.5 Training Objective and Pretraining

The full pretraining loss combines four terms:

$$\mathcal{L} = \mathcal{L}_{\text{spatial}} + \alpha\,\mathcal{L}_{\text{frechet}} + \beta\,\mathcal{L}_{\text{temporal}} + \gamma\,\mathcal{L}_{\text{div}}, \tag{10}$$

with α=0.5, β=0.1, γ=1.0. The model is pretrained for 30 epochs on the TUH EEG Corpus (Section 4.1) using AdamW [24]. Full optimizer and schedule details are in Appendix E.

For downstream tasks, the decoders and the Fréchet head are discarded. Single-head attention with a learned query (unused in pretraining) pools the spatial transformer output to $\mathbb{R}^{d_m}$. The pooled vector is concatenated with the band-averaged temporal path output ($\mathbb{R}^{d_t}$) and passed to a two-layer MLP classifier. We first train the classifier with the encoder frozen, then train end-to-end with a lower learning rate for the encoder (Appendix E). The projection, filter bank, and spatial transformer have 1,218,696 parameters (1.2M), which Table 1 compares with the baselines' backbones. The temporal path adds 34,177 (on SEED-V, 146,112 trained from scratch) and the head adds about 38K.

4 Experiments

Table 1: Downstream performance on 6 EEG benchmarks (mean ± std over 5 seeds unless marked). **Bold**: highest mean per column (before rounding). † below the score reported in CBraMod's paper on TUAB or TUEV (Appendix D). Params: backbone parameters that fine-tuning trains, without the task head, and for MENDR also without its temporal path (34K parameters, 146K on SEED-V, Section 3.5).

Model	Params	*Abnormality / Event*		*Motor / Emotion*		*Seizure / Sleep*	
		TUAB (AUROC↑)	TUEV (wF1↑)	PhysioMI (wF1↑)	SEED-V (wF1↑)	CHB-MIT (AUROC↑)	ISRUC (wF1↑)
MENDR	1.2M	**0.903**$_{\pm.002}$	0.769$_{\pm.016}$	0.579$_{\pm.006}$	0.389$_{\pm.007}$	**0.987**$_{\pm.010}$	0.753$_{\pm.016}$
LaBraM-Base	5.8M	0.903$_{\pm.001}$	0.787$_{\pm.016}$	0.617$_{\pm.004}$	**0.431**$_{\pm.011}$	0.978$_{\pm.014}$	0.807$_{\pm.004}$
BIOT	3.2M	0.880$_{\pm.003}$	0.737$_{\pm.023}$	0.303$_{\pm.008}$	0.356$_{\pm.009}^{(n=10)}$	0.954$_{\pm.025}^{(n=10)}$	0.771$_{\pm.007}^{(n=10)}$
CBraMod	4.9M	0.850$_{\pm.012}^{\dagger}$	**0.794**$_{\pm.011}^{\dagger}$	**0.621**$_{\pm.003}$	0.399$_{\pm.010}^{(n=10)}$	0.944$_{\pm.070}^{(n=10)}$	0.786$_{\pm.008}^{(n=10)}$
LUNA-Base	6.6M	0.887$_{\pm.005}$	0.761$_{\pm.018}$	0.574$_{\pm.003}$	0.357$_{\pm.013}^{(n=3)}$	0.964$_{\pm.035}$	**0.816**$_{\pm.007}$

Table 2: Baseline − MENDR per cell of Table 1 (Welch p in parentheses, positive: baseline higher).

Model	TUAB	TUEV	PhysioMI	SEED-V	CHB-MIT	ISRUC
LaBraM-Base	0.000 (0.94)	+0.018 (0.11)	+0.038 (<0.001)	+0.042 (<0.001)	−0.009 (0.31)	+0.054 (0.001)
BIOT	−0.023 (<0.001)	−0.032 (0.04)	−0.276 (<0.001)	−0.032 (<0.001)	−0.033 (0.003)	+0.018 (0.07)
CBraMod	−0.052 (<0.001)	+0.025 (0.02)	+0.042 (<0.001)	+0.011 (0.03)	−0.042 (0.09)	+0.033 (0.008)
LUNA-Base	−0.016 (0.002)	−0.007 (0.50)	−0.005 (0.18)	−0.031 (0.04)	−0.022 (0.24)	+0.063 (<0.001)

4.1 Setup

Pretraining data. We pretrain on the TUH EEG Corpus v2.0.1 [26] (69,672 recordings from 14,987 subjects). After quality filtering (Appendix C), 50,210 recordings from 14,559 subjects (about 13,300 hours) enter pretraining. We select 19 electrodes from the international 10–20 system, apply a 60 Hz notch filter and a nominal 0.1–75 Hz bandpass, resample to 200 Hz, and extract non-overlapping 6-second segments. Each pretraining epoch draws up to 10 random segments from every recording (about 765 hours). TUAB and TUEV (TUH EEG Events) are subsets of this corpus with their own subject-level partitions, and their test recordings remain in the unlabeled pretraining set (252 of the 253 TUAB test subjects).

Downstream benchmarks. We score the six datasets in Table 3 by area under the receiver operating characteristic curve (AUROC) on binary tasks and by weighted F1 (wF1) otherwise.

Baselines and protocol. We compare against BIOT [39], LaBraM-Base [18], CBraMod [37] and LUNA-Base [9] (Table 1). We report EEGPT [36] (25.3M backbone parameters) on TUAB and TUEV here, not in the table. On TUAB it is behind MENDR (AUROC 0.892 ± 0.005 over 5 seeds, EEGPT − MENDR = −0.011, Welch $p = 0.005$). On TUEV it is ahead and has the highest mean of all models (wF1 0.802 ± 0.007, EEGPT − MENDR = +0.033, $p = 0.006$, Appendix D). Every

Table 3: Downstream evaluation datasets (PhysioMI: PhysioNet motor imagery, CV: cross-validation).

Dataset	Task	Classes	Channels	Rate	Segment	Train/val/test split
TUAB [23, 26]	Abnormality detection	2	19	200 Hz	6 s	Official + 10% val
TUEV [26, 11]	Event classification	6	19	200 Hz	5 s	Official + 10% val
PhysioMI [32, 10]	Motor imagery	4	64	200 Hz	4 s	Subj. 5-fold CV + 15% val
SEED-V [22]	Emotion recognition	5	62	200 Hz	1 s	Trials 1–5/6–10/11–15
CHB-MIT [33]	Seizure detection	2	16	200 Hz	10 s	chb01–19/20–21/22–23
ISRUC [20]	Sleep staging	5	6	200 Hz	30 s	Subj. 1–80/81–90/91–100

model, including ours, is fine-tuned from its official pretrained weights on the same splits under one *strict protocol*: we select the checkpoint on validation and score the test set once per seed. Each baseline uses its own published optimizer, schedule, and input normalization (settings and deviations: Appendix D). Where we ran more than one configuration for a baseline, Table 1 reports the strongest by test mean. We use two-sided t-tests on per-seed scores: Welch's test between models and a paired test between MENDR variants that share seeds, unless stated otherwise. We write "ahead of" or "behind" only where $p < 0.05$, and otherwise "not significantly different". This means only that $p \geq 0.05$, not that the two are equal. We apply no multiplicity correction, and differences are computed before rounding.

MENDR configuration. All MENDR results in Table 1 fine-tune one pretrained channel-agnostic (CA) encoder, with the configuration chosen per task on validation scores only and scored on five seeds (789, 1024, 31, 64, 99). The *default configuration* uses a 320 ms covariance window (64 samples, stride 32) and re-initializes the projection for the target montage. CHB-MIT and ISRUC use it unchanged. TUAB, TUEV and PhysioMI add light augmentation (time shift up to 5%, channel dropout with p=0.1). TUEV and PhysioMI use a 640 ms window, and TUEV keeps the pretrained projection. SEED-V uses a 1 s window (one covariance per band), filters reset to 4–8, 8–13, 13–30 and 30–45 Hz without weight decay, and a larger temporal path (multi-scale convolutions, two-layer transformer) trained from scratch, so only its spatial transformer is pretrained. All runs use label smoothing, class weighting and per-band BatchNorm statistics at evaluation (Appendix E).

4.2 Comparison with EEG Foundation Models

MENDR has the highest mean on TUAB and CHB-MIT, and no model is significantly ahead of it on these two tasks (Table 2). On the other four tasks, at least one model is ahead: CBraMod and EEGPT on TUEV (Section 4.1); LaBraM-Base and CBraMod on PhysioMI and SEED-V; and LaBraM-Base, CBraMod and LUNA-Base on ISRUC. MENDR is ahead of BIOT on every task except ISRUC, and ahead of LUNA-Base on TUAB and SEED-V. Larger models do not consistently score higher (Figure 2): EEGPT, the largest, is behind MENDR on TUAB but highest on TUEV. LUNA-Base has the highest mean on ISRUC, and its SEED-V entry is the best of seven configurations (Appendix D).

Table 4: Per-class F1 (in label order) and macro F1 of the Table 1 MENDR models, mean over 5 seeds (PhysioMI: over folds and seeds). Class names: Appendix C.

Task	Per-class F1						Macro F1
TUEV	0.21	0.73	0.41	0.60	0.44	0.87	0.543
classes: SPSW, GPED, PLED, EYEM, ARTF, BCKG							
PhysioMI	0.63	0.63	0.52	0.54	—	—	0.579
classes: left fist, right fist, both fists, both feet							
ISRUC	0.85	0.47	0.75	0.82	0.80	—	0.737
classes: W, N1, N2, N3, REM							
SEED-V	0.43	0.35	0.46	0.31	0.36	—	0.384
classes: Happy, Sad, Neutral, Disgust, Fear							

Per-class behavior (Table 4, Appendix G). On TUEV, the background class dominates the weighted score, and rare spike and sharp wave (SPSW) segments are more often classified as one of the two periodic-discharge classes than as SPSW. On ISRUC, the N1 stage is the weak class. PhysioMI saliency maps are in Appendix H.

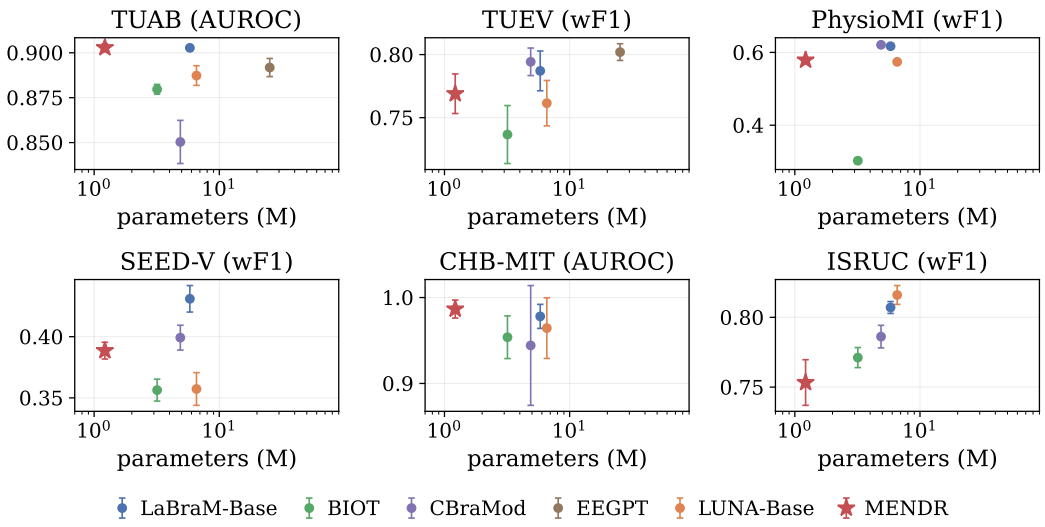


Figure 2: Score versus parameter count (mean ± std): the models of Table 1, and EEGPT on TUAB and TUEV (Section 4.1). Parameters: backbone counts as in Table 1.

4.3 Architecture and Pretraining

Table 5: Architecture versus pretraining (5 seeds, paired t-test by seed). Both projections are trained from scratch in each task's Table 1 configuration. CA: channel-agnostic. Fixed: a learned linear layer on the task's channels (PhysioMI: 19 of its 64). Δ_{arch} = CA − Fixed. Δ_{pre} = MENDR row of Table 1 − CA from scratch. It also includes a schedule difference (Appendix E).

Task	Fixed, from scratch	CA, from scratch	Δ_{arch}	Δ_{pre}
TUAB (AUROC)	$0.885_{\pm.008}$	$0.899_{\pm.005}$	$+0.014$ (p=0.040)	$+0.004$ (p=0.222)
TUEV (wF1)	$0.752_{\pm.016}$	$0.769_{\pm.017}$	$+0.017$ (p=0.149)	0.000 (p=0.989)
PhysioMI (wF1)	$0.506_{\pm.001}$	$0.543_{\pm.010}$	$+0.037$ (p=0.001)	$+0.036$ (p<0.001)
SEED-V (wF1)	$0.384_{\pm.012}$	$0.390_{\pm.010}$	$+0.006$ (p=0.453)	-0.002 (p=0.829)
CHB-MIT (AUROC)	$0.973_{\pm.018}$	$0.986_{\pm.011}$	$+0.013$ (p=0.065)	$+0.001$ (p=0.923)
ISRUC (wF1)	$0.785_{\pm.011}$	$0.769_{\pm.019}$	-0.016 (p=0.189)	-0.016 (p=0.343)

Architecture versus pretraining (Table 5). Trained from scratch, the CA projection is ahead of the fixed one on PhysioMI and TUAB and not significantly different from it on the other four tasks, with the fixed projection's mean higher on ISRUC. On PhysioMI, the fixed projection sees the 19 electrodes of the pretraining montage rather than all 64, so the PhysioMI Δ_{arch} reflects both projection type and electrode count. Pretraining is ahead of training from scratch on PhysioMI, with no significant difference on the other five tasks. Δ_{pre} also includes the pretrained runs' five head-only epochs and halved learning rates (Appendix E). On SEED-V, using only the 19 pretraining electrodes improves no fine-tuned model, and a classifier trained on the frozen pretrained encoder is close to chance (Appendix I).

Reconstruction (Figure 3). On masked tokens, the pretrained model explains 0.67 of the variance of each tangent coordinate at 25% masking and 0.47 at 75% (Appendix J.1).

4.4 Component Ablation

The spatial path alone is not significantly different from the full model on either task, while the temporal path alone is behind it on TUEV and not significantly different from it on TUAB (Table 6, Appendix J.2). On TUAB, the temporal path with the projection (70K parameters) is within 0.006 of the full model, so TUAB does not show the spatial path's contribution. A tangent-space

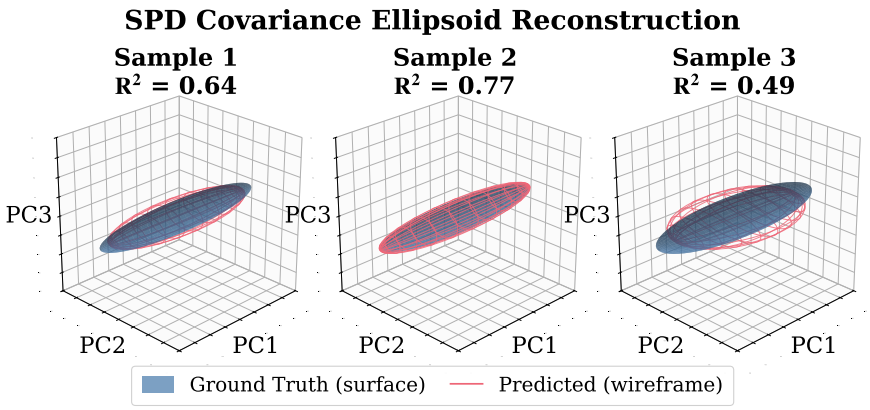


Figure 3: The first masked SPD token of three segments at 50% masking: ground truth (blue surface) and prediction (rose wireframe) in three eigen-directions (ranks 1, 11 and 22 of 32, labeled PC1–PC3) of the true token, radii on a log-eigenvalue scale. The printed R^2 uses each coordinate's mean as the baseline prediction (Appendix J.1).

logistic regression with one covariance per segment is behind the spatial path alone on both tasks (Appendix L).

Table 6: Component ablation, from scratch, Table 1 configuration (mean ± std, 5 seeds, paired t-test by seed). Full = CA, from scratch, in Table 5 (1.2M spatial encoder plus 34K temporal path). Params: backbone parameters the variant trains.

Variant	Params	TUAB (AUROC)	TUEV (wF1)
Spatial path only	1.2M	$0.894_{\pm.003}$	$0.768_{\pm.014}$
Projection + temporal path	70K	$0.893_{\pm.003}$	$0.702_{\pm.009}$
Full − spatial only		$+0.005$ (p=0.131)	$+0.001$ (p=0.894)
Full − temporal only		$+0.006$ (p=0.068)	$+0.067$ (p=0.002)

4.5 Model Scale

Table 7: Model size (mean ± std, 5 seeds), fine-tuned with the Table 1 configuration and seeds. The from-scratch Default cells are the CA cells of Table 5. Pretrained: one encoder per size, pretrained as the Table 1 encoder but with weight decay 0.01 (Table 1: 0.05). Pretrained runs add five head-only epochs and lower learning rates, as in Table 5. Params: as in Table 1.

		TUAB (AUROC)		TUEV (wF1)	
Variant	Params	from scratch	pretrained	from scratch	pretrained
Tiny (d=64, L=4)	0.4M	$0.899_{\pm.004}$	$0.899_{\pm.002}$	$0.753_{\pm.031}$	$0.784_{\pm.012}$
Default (d=128, L=4)	1.2M	$0.899_{\pm.005}$	$0.901_{\pm.003}$	$0.769_{\pm.017}$	$0.791_{\pm.011}$
Small (d=192, L=6)	3.3M	$0.899_{\pm.004}$	$0.902_{\pm.006}$	$0.787_{\pm.019}$	$0.757_{\pm.007}$
Base (d=256, L=8)	7.3M	$0.900_{\pm.004}$	$0.903_{\pm.001}$	$0.775_{\pm.033}$	$0.710_{\pm.024}$
Large (d=384, L=12)	22.9M	$0.895_{\pm.004}$	$0.899_{\pm.002}$	$0.784_{\pm.014}$	$0.724_{\pm.007}$

Larger encoders do not help (Table 7). From scratch, TUAB stays between 0.895 and 0.900 AUROC over a more than 50-fold increase in parameters, and on TUEV no size differs significantly from Tiny (Welch $p \geq 0.076$). Pretrained TUAB is not significantly different from scratch at any size (Welch $p \geq 0.092$). Pretrained TUEV is not significantly different from scratch at Tiny and Default ($p \geq 0.051$) and behind at Small, Base and Large (by 0.029 to 0.065, $p \leq 0.020$). Pretraining weight decay alone (0.01 here, 0.05 in Table 1) moves the Default TUEV score by 0.022. Before fine-tuning,

the Base and Large spatial transformers, unlike smaller ones, output a constant (Appendix J.3). On CHB-MIT and ISRUC, the 0.4M Tiny model from scratch ($0.983_{\pm.005}$ AUROC, $0.760_{\pm.023}$ wF1) does not differ significantly from the 1.2M CA model (Table 5, p=0.753, 0.548).

4.6 Limits of Channel-Agnostic Transfer

The projection identifies channels by their position in the input, not by name. Permuting, reversing or removing test channels therefore lowers AUROC sharply, with or without pretraining. On TUAB, the pretrained model falls from 0.902 to 0.721 when channels are permuted and to 0.740 when 8 are removed (from scratch: 0.543) (Appendix K.1).

Table 8: Spatial projection after fine-tuning (Table 1 configuration, seed 789). $\mathbf{A}$: mixing weights. $\mathbf{D}$: $\mathbf{A}$ minus its mean row, the part that differs between virtual electrodes (Section 3.1). Max dev.: largest relative deviation of a weight from $1/C$ in the median segment. Eff. rank: effective number of independent rows (maximum for $\mathbf{D}$ after the slash). Size of $\mathbf{D}$: $\|\mathbf{D}\|_F$ relative to the shared part of $\mathbf{A}$. Last three columns: $\mathbf{D}$ before (pretrained for TUEV, random otherwise) vs. after fine-tuning (Appendix K.2).

						Before vs. after fine-tuning		
Task	C	Max dev.	Eff. rank $\mathbf{A}$	Eff. rank $\mathbf{D}$	Size of $\mathbf{D}$	Eff. rank before	Size ratio	Similarity
TUAB	19	0.9%	1.05	16.1 / 18	0.14%	14.0	0.9×	0.68
TUEV	19	1.7%	1.11	12.7 / 18	0.37%	10.8	2.1×	0.82
ISRUC	6	2.3%	1.13	4.5 / 5	0.93%	4.7	5.4×	0.73
PhysioMI	64	1.3%	1.09	23.4 / 31	0.23%	22.4	1.4×	0.79

What the projection computes (Table 8). After training, the mixing weights are almost uniform: in the median segment, every weight is within 3% of $1/C$, so each virtual electrode starts close to the channel average. The LayerNorm removes this common average, leaving the small differences between virtual electrodes ($\mathbf{D}$), which it multiplies by an almost fixed factor because they are tiny (Appendix K.2). The projection therefore computes many weighted differences between channels rather than selecting a few. $\mathbf{D}$ has a high effective rank (at least 70% of its maximum), and fine-tuning keeps much of its mostly random starting pattern (cosine similarity 0.68–0.82).

5 Conclusion

MENDR builds the Riemannian structure of EEG covariance into a small foundation model. Its spatial encoder is 2.6 to 21 times smaller than the backbones of the compared models. MENDR is not significantly different from the best of them on seizure and abnormality detection, and behind by 0.03 on event classification and by 0.04 to 0.06 on sleep staging, motor imagery, and emotion recognition. Trained from scratch, the same architecture is not significantly different from the pretrained model on five of six tasks (Table 5), so these scores owe little to pretraining. **Limitations.** (i) TUAB and TUEV overlap the pretraining corpus, so neither is an out-of-domain test. (ii) The fine-tuned model is not robust to test-time channel permutation or removal (Section 4.6), and trained from scratch, its projection is ahead of a fixed linear one on only two of six tasks (Table 5). (iii) No experiment isolates the matrix logarithm or the Fréchet term, so the paper does not show that the Log-Euclidean embedding itself accounts for the results. (iv) Self-supervised pretraining gives no consistent benefit (ahead on one of six tasks), and larger encoders do not help: pretraining the two largest collapsed them to a constant output (Sections 4.3 and 4.5). (v) Tests use five seeds on fixed splits, which matters most for CHB-MIT with its two test cases. (vi) The temporal objective trains only the temporal convolutions, and in the lowest band it reconstructs masked spans worse than interpolation (Appendix J.1). (vii) Fewer parameters do not make MENDR cheaper to run. On CPU it is slower than every baseline except EEGPT, and on GPU it is the slowest at batch sizes 32, 64 and 256, where its double-precision eigendecompositions take up to two thirds of the time (Appendix M). **Future work.** Directions include a multi-scale SPD path with task-adaptive covariance resolution, an electrode-coordinate projection, and pretraining on corpora beyond clinical 19-channel EEG.

Acknowledgments and Disclosure of Funding

The authors acknowledge the AI Makerspace of the College of Engineering (RRID:SCR_028058), provided by the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology, Atlanta, Georgia, USA (RRID:SCR_027619), for computational resources used in this work. We also gratefully acknowledge funding and fellowships that contributed to this work, including a Wallace H. Coulter Distinguished Faculty Fellowship, a Petit Institute Faculty Fellowship, and research funding from Amazon and Microsoft Research awarded to Professor May D. Wang. The authors declare no competing interests.

References

[1] P. A. Abhang, B. W. Gawali, and S. C. Mehrotra. *Introduction to EEG-and speech-based emotion recognition*. Elsevier, 2016.

[2] H. Altaheri, G. Muhammad, M. Alsulaiman, S. U. Amin, G. A. Altuwaijri, W. Abdul, M. A. Bencherif, and M. Faisal. Deep learning techniques for classification of electroencephalogram (EEG) motor imagery (MI) signals: A review. *Neural Computing and Applications*, 35(20):14681–14722, 2023.

[3] B. Aristimunha, C. Ju, A. Collas, F. Bouchard, A. Mian, B. Thirion, S. Chevallier, and R. Kobler. SPD Learn: A geometric deep learning python library for neural decoding through trivialization. *arXiv preprint arXiv:2602.22895*, 2026.

[4] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, 56(2):411–421, 2006.

[5] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 29(1):328–347, 2007.

[6] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten. Multiclass brain–computer interface classification by Riemannian geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928, 2012.

[7] A. Biasiucci, B. Franceschiello, and M. M. Murray. Electroencephalography. *Current Biology*, 29(3):R80–R85, 2019.

[8] M. Congedo, A. Barachant, and R. Bhatia. Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review. *Brain-Computer Interfaces*, 4(3):155–174, 2017.

[9] B. Döner, T. M. Ingolfsson, L. Benini, and Y. Li. LUNA: Efficient and Topology-Agnostic Foundation Model for EEG Signal Analysis. In *Advances in Neural Information Processing Systems*, 2025.

[10] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[11] A. Harati, M. Golmohammadi, S. Lopez, I. Obeid, and J. Picone. Improved EEG event classification using differential energy. In *2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–4. IEEE, 2015.

[12] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, 2022.

[13] Y. He, T. P. Luu, K. Nathan, S. Nakagome, and J. L. Contreras-Vidal. A mobile brain-body imaging dataset recorded during treadmill walking with a brain-computer interface. *Scientific Data*, 5(1):180074, 2018.

[14] Z. Huang and L. Van Gool. A Riemannian network for SPD matrix learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 2036–2042. AAAI Press, 2017.

[15] C. Ionescu, O. Vantzos, and C. Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2965–2973, 2015.

[16] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning*, pages 4651–4664. PMLR, 2021.

[17] W.-B. Jiang, Y. Wang, B.-L. Lu, and D. Li. NeuroLM: A Universal Multi-task Foundation Model for Bridging the Gap between Language and EEG Signals. In *International Conference on Learning Representations*, pages 55436–55457, 2025.

[18] W.-B. Jiang, L.-M. Zhao, and B.-L. Lu. Large Brain Model for Learning Generic Representations with Tremendous EEG Data in BCI. In *International Conference on Learning Representations*, pages 16405–16426, 2024.

[19] I. Katsman, E. Chen, S. Holalkere, A. Asch, A. Lou, S. N. Lim, and C. M. De Sa. Riemannian residual neural networks. *Advances in Neural Information Processing Systems*, 36:63502–63514, 2023.

[20] S. Khalighi, T. Sousa, J. M. Santos, and U. Nunes. ISRUC-Sleep: A comprehensive public dataset for sleep researchers. *Computer Methods and Programs in Biomedicine*, 124:180–192, 2016.

[21] D. Kostas, S. Aroca-Ouellette, and F. 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.

[22] W. Liu, J.-L. Qiu, W.-L. Zheng, and B.-L. Lu. Comparing recognition performance and robustness of multimodal deep learning models for multimodal emotion recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 14(2):715–729, 2022.

[23] S. López, G. Suarez, D. Jungreis, I. Obeid, and J. Picone. Automated identification of abnormal adult EEGs. In *2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–5. IEEE, 2015.

[24] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[25] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[26] I. Obeid and J. Picone. The Temple University Hospital EEG data corpus. *Frontiers in Neuroscience*, 10:196, 2016.

[27] Y.-T. Pan, J.-L. Chou, and C.-S. Wei. MAtt: A manifold attention network for EEG decoding. *Advances in Neural Information Processing Systems*, 35:31116–31129, 2022.

[28] R. Peng, C. Zhao, Y. Xu, J. Jiang, G. Kuang, J. Shao, and D. Wu. WAVELET2VEC: A filter bank masked autoencoder for EEG-based seizure subtype classification. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[29] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.

[30] G. Pfurtscheller and F. H. Lopes da Silva. Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical Neurophysiology*, 110(11):1842–1857, 1999.

[31] M. Ravanelli and Y. Bengio. Speaker recognition from raw waveform with SincNet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028. IEEE, 2018.

[32] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw. BCI2000: a general-purpose brain-computer interface (BCI) system. *IEEE Transactions on Biomedical Engineering*, 51(6):1034–1043, 2004.

[33] A. H. Shoeb. *Application of machine learning to epileptic seizure onset detection and treatment*. PhD thesis, Massachusetts Institute of Technology, 2009.

[34] R. Thapa, B. He, M. R. Kjær, H. Moore, G. Ganjoo, E. Mignot, and J. Zou. SleepFM: multi-modal representation learning for sleep across brain activity, ECG and respiratory signals. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024.

[35] I. E. Tibermacine, S. Russo, A. Tibermacine, A. Rabehi, B. Nail, K. Kadri, and C. Napoli. Riemannian geometry-based EEG approaches: A literature review. *arXiv preprint arXiv:2407.20250*, 2024.

[36] G. Wang, W. Liu, Y. He, C. Xu, L. Ma, and H. Li. EEGPT: Pretrained transformer for universal and reliable representation of EEG signals. *Advances in Neural Information Processing Systems*, 37:39249–39280, 2024.

[37] J. Wang, S. Zhao, Z. Luo, Y. Zhou, H. Jiang, S. Li, T. Li, and G. Pan. CBraMod: A criss-cross brain foundation model for EEG decoding. In *International Conference on Learning Representations*, 2025.

[38] Z. Wang, Z. Song, Y. Guo, Y. Liu, G. Xu, M. Zhang, and Z. Zhang. EEG-ReMinD: Enhancing neurodegenerative EEG decoding through self-supervised state reconstruction-primed Riemannian dynamics. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[39] C. Yang, M. Westover, and J. Sun. BIOT: Biosignal transformer for cross-data learning in the wild. *Advances in Neural Information Processing Systems*, 36:78240–78260, 2023.

[40] Z. Yuan, F. Shen, M. Li, Y. Yu, F. Wu, C. Tan, and Y. Yang. BrainWave: A brain signal foundation model for clinical applications. *arXiv preprint arXiv:2402.10251*, 2024.

[41] D. Zhang, Z. Yuan, Y. Yang, J. Chen, J. Wang, and Y. Li. Brant: Foundation model for intracranial neural signal. *Advances in Neural Information Processing Systems*, 36:26304–26321, 2023.

[42] G. Zhang and A. Etemad. Spatio-temporal EEG representation learning on Riemannian manifold and Euclidean space. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(2):1469–1483, 2024.

[43] R. Zheng, D. Zhang, Z. Yuan, J. Chen, Y. Yang, et al. Beatrix: Out-of-distribution generalization of large EEG model via invariant contrastive fine-tuning. OpenReview, `https://openreview.net/forum?id=IjBndR92Zy`, 2025.

A Numerically Stable Matrix Logarithm Backward Pass

The matrix logarithm $\mathrm{Log}(\mathbf{A}) = \mathbf{U}\,\mathrm{diag}(\log\lambda_1, \ldots, \log\lambda_n)\mathbf{U}^\top$ requires differentiating through the eigendecomposition. The standard backward pass involves the Daleckii–Kreĭn derivative kernel [15]:

$$\Gamma_{ij} = \begin{cases} \dfrac{\log\lambda_i - \log\lambda_j}{\lambda_i - \lambda_j} & \text{if } \lambda_i \neq \lambda_j, \\ \dfrac{1}{\lambda_i} & \text{if } \lambda_i = \lambda_j, \end{cases} \tag{11}$$

where the degenerate case follows from L'Hôpital's rule as $\lambda_j \to \lambda_i$. The two cases depend on whether $\lambda_i = \lambda_j$, not on whether $i = j$. This is because $1/\lambda_i$ is the correct limit for *any* repeated eigenvalue, including off-diagonal entries ($i \neq j$) with $\lambda_i = \lambda_j$. EEG covariance matrices often have pairs of nearly equal eigenvalues, because several channels measure similar cortical activity. When $|\lambda_i - \lambda_j|$ is small, this causes catastrophic cancellation in the finite-difference branch.

We implement a numerically stable version in three steps. (i) We compute the finite-difference kernel only when $|\lambda_i - \lambda_j| > \tau$ with $\tau = 10^{-6} \cdot \overline{|\lambda|}$, where $\overline{|\lambda|}$ is the mean absolute eigenvalue of the matrix, and replace it with the L'Hôpital limit $1/\lambda_i$ otherwise, or where the kernel is not finite. (ii) We clamp all eigenvalues to $\lambda \geq 10^{-12}$ before the logarithm, as a safeguard against numerical error in the eigensolver (the $\epsilon\mathbf{I}$ term in Eq. (5) already guarantees $\lambda_{\min} \geq \epsilon = 10^{-3}$). (iii) Any remaining NaN or Inf gradient entries are set to zero. The eigendecomposition and this backward pass run in double precision. Before the eigendecomposition, we symmetrize the matrix as $(\mathbf{A} + \mathbf{A}^\top)/2$. The batched GPU eigensolver can fail to converge when two eigenvalues are exactly equal, so we also add a tiny diagonal offset that grows linearly along the diagonal, $10^{-7}\,\overline{|A_{ii}|}\;\mathrm{diag}(0, \frac{1}{n-1}, \ldots, 1)$, where $\overline{|A_{ii}|}$ is the mean absolute diagonal entry. This separates exactly repeated eigenvalues by a negligible amount. The offset is at most $\tau/10$, so such pairs still take the $1/\lambda_i$ branch. When $\lambda_i = \lambda_j$, the limits $1/\lambda_i$ and $1/\lambda_j$ are equal, so using $1/\lambda_i$ for both (i, j) and (j, i) is consistent. The explicit gradient symmetrization below removes any remaining asymmetry.

Given the gradient $\frac{\partial\mathcal{L}}{\partial\,\mathrm{Log}(\mathbf{A})}$, the gradient with respect to $\mathbf{A}$ is:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{A}} = \mathbf{U}\left(\mathbf{\Gamma} \odot \left(\mathbf{U}^\top \frac{\partial\mathcal{L}}{\partial\,\mathrm{Log}(\mathbf{A})}\mathbf{U}\right)\right)\mathbf{U}^\top, \tag{12}$$

where $\odot$ denotes the Hadamard (element-wise) product and $\mathbf{\Gamma} = [\Gamma_{ij}]$ is the kernel matrix from Eq. (11). Because $\mathbf{A}$ must be symmetric, the result is symmetrized as $\frac{1}{2}\big(\partial\mathcal{L}/\partial\mathbf{A} + (\partial\mathcal{L}/\partial\mathbf{A})^\top\big)$. The incoming gradient is symmetrized in the same way.

B SPD Guarantee and Log-Euclidean Metric Properties

SPD guarantee. We verify that the windowed covariance matrices $\hat{\mathbf{\Sigma}}_j^{(k)}$ in Eq. (5) are strictly SPD. Let $\tilde{\mathbf{Z}} \in \mathbb{R}^{d_s \times w}$ be a mean-centered window. The sample covariance $\mathbf{\Sigma} = \tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^\top/(w-1)$ is positive semi-definite (PSD) by construction: for any $\mathbf{x} \in \mathbb{R}^{d_s}$, $\mathbf{x}^\top\mathbf{\Sigma}\mathbf{x} = \|\tilde{\mathbf{Z}}^\top\mathbf{x}\|^2/(w-1) \geq 0$. Trace normalization preserves PSD since $\mathrm{tr}(\mathbf{\Sigma}) > 0$ for any window that is not constant over time. A constant window gives $\mathbf{\Sigma} = \mathbf{0}$, and the implementation then returns $\hat{\mathbf{\Sigma}} = \epsilon\mathbf{I}$. Adding $\epsilon\mathbf{I}$ with $\epsilon = 10^{-3}$ shifts all eigenvalues by ϵ, making the result strictly positive definite: $\lambda_{\min}(\hat{\mathbf{\Sigma}}) \geq \epsilon > 0$, so the matrix logarithm is well defined. We summarize the properties of the Log-Euclidean framework [4, 5] that MENDR uses.

Geodesic distance. For $\mathbf{A}, \mathbf{B} \in \mathcal{S}_{++}^n$, the Log-Euclidean distance is:

$$d_{\mathrm{LE}}(\mathbf{A}, \mathbf{B}) = \|\,\mathrm{Log}(\mathbf{A}) - \mathrm{Log}(\mathbf{B})\|_F. \tag{13}$$

After the distance-preserving vectorization of Eq. (7), this becomes $d_{\mathrm{LE}}(\mathbf{A}, \mathbf{B}) = \|\mathbf{v}_A - \mathbf{v}_B\|_2$, so the squared ℓ_2 loss of Eq. (8) is proportional to the mean squared geodesic distance over the masked tokens.

Fréchet mean. The Fréchet mean $\bar{\mathbf{S}}$ minimizes $\sum_i d_{\mathrm{LE}}^2(\mathbf{S}_i, \bar{\mathbf{S}})$ and has a closed form:

$$\bar{\mathbf{S}} = \mathrm{Exp}\left(\frac{1}{N}\sum_{i=1}^{N}\mathrm{Log}(\mathbf{S}_i)\right), \tag{14}$$

which, after vectorization, reduces to the arithmetic mean: $\bar{\mathbf{v}} = \frac{1}{N}\sum_i \mathbf{v}_i$. This justifies our Fréchet mean loss (Eq. (9)).

Orthogonal invariance. Unlike the affine-invariant Riemannian metric (AIRM) [29], which is invariant under any invertible congruence $\mathbf{PAP}^\top$, the Log-Euclidean distance is invariant under *orthogonal* congruence: for any orthogonal $\mathbf{O}$ ($\mathbf{O}^\top\mathbf{O} = \mathbf{I}$), $d_{\mathrm{LE}}(\mathbf{OAO}^\top, \mathbf{OBO}^\top) = d_{\mathrm{LE}}(\mathbf{A}, \mathbf{B})$ (and under matrix inversion and a common scaling). The tangent matrix itself is equivariant, not invariant: $\mathrm{Log}(\mathbf{OAO}^\top) = \mathbf{O}\,\mathrm{Log}(\mathbf{A})\mathbf{O}^\top$, so layers that read $\mathbf{v}$ directly are not rotation-invariant. In MENDR, the covariance is taken after a data-dependent projection and a per-time-step LayerNorm, so a rotation of the physical channels does not act on $\hat{\mathbf{\Sigma}}$ as a fixed orthogonal congruence. Invariance under every invertible congruence would need a metric such as AIRM, which lacks Log-Euclidean's computational convenience and tangent-space linearity.

C Dataset Details

Datasets. TUAB is the TUH Abnormal EEG Corpus, and TUEV is the TUH EEG Events Corpus. TUEV has six classes: spike and sharp wave (SPSW), generalized periodic epileptiform discharge (GPED), periodic lateralized epileptiform discharge (PLED), eye movement (EYEM), artifact (ARTF) and background (BCKG). PhysioMI is the PhysioNet EEG Motor Movement/Imagery Dataset. SEED-V is the five-emotion SJTU Emotion EEG Dataset of Shanghai Jiao Tong University. CHB-MIT is the CHB-MIT Scalp EEG Database of Children's Hospital Boston and the Massachusetts Institute of Technology. ISRUC is subgroup I of the ISRUC-Sleep dataset, and its five classes are wake (W), the non-rapid-eye-movement stages N1, N2 and N3, and rapid eye movement (REM) sleep.

Preprocessing. All datasets are resampled to 200 Hz (ISRUC is recorded at that rate) and scaled to units of 100 μV, without z-scoring. TUEV, PhysioMI and SEED-V are bandpass filtered at 0.1–75 Hz (zero-phase, 5th-order Butterworth) and notch filtered at the local mains frequency (60 Hz for TUEV and PhysioMI, 50 Hz for SEED-V). The band edges are clamped between 0.001 and 0.999 times the raw recording's Nyquist frequency, raising the lower edge to about 0.13 Hz for TUEV and 0.5 Hz for SEED-V and lowering the upper edge to 63.9 Hz for the three PhysioMI subjects recorded at 128 Hz. TUAB uses the same band and a 60 Hz notch with MNE's default zero-phase finite impulse response (FIR) filters and drops segments whose amplitude exceeds 500 μV. ISRUC is bandpass filtered at 0.3–35 Hz with a 50 Hz notch, and CHB-MIT is used without filtering. PhysioMI is re-referenced to the common average before bandpass filtering. The TUH pretraining data use the 0.1–75 Hz bandpass with the same lower-edge rule and a 60 Hz notch. TUH recordings of 130 s or less are excluded, 60 s are trimmed from each end of the others, and 6-s segments whose amplitude exceeds 100 μV are rejected. For TUAB and TUEV, the official evaluation set is the test set, and 10% of the official training subjects (208 for TUAB, 29 for TUEV) form the validation set. For PhysioMI, we use runs 4/8/12 (left/right fist imagery) and 6/10/14 (both fists/feet imagery) with subjects split into 5 folds. For TUEV we follow the protocol of BIOT, LaBraM and CBraMod: one 5-s window on all channels is cut around every row of the event annotation files, so an event annotated on several channels yields identical windows. Only about 11% of the test windows are distinct (3,107 of 28,098), and the weighted F1 therefore counts background and events annotated on many channels several times. We drop windows that cross a recording edge, whereas the baselines' code wraps them around. Every model in Table 1 uses the same windows. For SEED-V, we follow the cross-trial protocol used by CBraMod [37]: within each of the three sessions, trials 1–5 form the training set, trials 6–10 the validation set, and trials 11–15 the test set, with all 16 subjects present in every split. One SEED-V recording is unreadable and is excluded, so 47 of the 48 sessions enter the splits. ISRUC uses subjects 1–80 for training, 81–90 for validation, and 91–100 for testing. Subjects 8 and 40 are also excluded because their recordings lack the six referenced channels, leaving 78 training subjects. CHB-MIT uses cases chb01–chb19 for training, chb20–chb21 for validation, and chb22–chb23 for testing. Cases chb12, chb13 and chb17 are not in the preprocessed data (BIOT's CHB-MIT preprocessing [39]), so 16 cases enter training. Case chb21 is a later recording of the chb01 subject.

D Baseline Fine-tuning Details

Table 1 contains no numbers taken from the baselines' original papers. Every baseline is fine-tuned from the pretrained weights released by its authors: BIOT's 16-channel model pretrained on PREST, LaBraM-Base, CBraMod, the large EEGPT variant (58 channels, 4 s), and LUNA-Base. Table 9 lists

the per-baseline fine-tuning hyperparameters. All models share the splits and select their checkpoint on validation, each with its own selection metric. The baselines use seeds 42, 123, 456, 789 and 1024 (the ten-seed entries, marked $(n{=}10)$, add 7, 31, 64, 99 and 2026, and LUNA-Base's SEED-V entry uses only 42, 123 and 456), and MENDR uses 789, 1024, 31, 64 and 99.

Table 9: Baseline fine-tuning hyperparameters (T, E, P, S, C, I: TUAB, TUEV, PhysioMI, SEED-V, CHB-MIT, ISRUC). WD: weight decay. Without early stopping, every epoch is run and the best validation epoch is kept.

Model	Optimizer, learning rate	WD	Batch	Epochs	Early stop	Warm-up, schedule
BIOT	Adam, 10^{-3}	10^{-5}	64 T, 512 E, 256 P/S/C/I	100 T/E/P/I, 50 S/C	patience 5 T, 10 E/P/S, 15 C/I	none, cosine (T) or constant
LaBraM-Base	AdamW, 5×10^{-4}, layer decay 0.65 (C: none)	0.05	512 T/E, 64 P/S/C/I	50	none T/E, patience 20 P/S, 10 C/I	5 epochs, cosine
CBraMod	AdamW, 10^{-4} (head 5×10^{-4})	0.05	64	50	patience 15	none, cosine
EEGPT	AdamW, 5×10^{-4}	0.05	100	50	none	5 epochs, cosine
LUNA-Base	AdamW, 5×10^{-4}, layer decay 0.75	0.05	256 T/E/P, 64 S/C/I	30 (S: 50)	none	10 epochs, cosine
MENDR	AdamW, 10^{-4} (Appendix E)	0.05	64	50	patience 15	Appendix E

BIOT [39] uses 16 bipolar channels at 200 Hz with STFT-based tokenization ($n_{\text{fft}}{=}200$, $\text{hop}_{\text{length}}{=}100$), and on ISRUC its six channels are zero-padded to 16. On SEED-V, we zero-pad its 1-s windows to 10 s. Padded and unpadded scores do not differ significantly (0.356 against 0.354 over the same ten seeds, paired $p = 0.53$). On PhysioMI, its 4-s trials are not padded.

LaBraM-Base [18] uses each task's own channels (on SEED-V, the 60 without CB1 and CB2) at 200 Hz in patches of 200 samples, without padding to 10 s, and on ISRUC its temporal position embedding is interpolated from 16 to 30 patches.

CBraMod [37] uses 16 bipolar channels on TUAB, TUEV and CHB-MIT and each dataset's own channels otherwise (on SEED-V the 60 without CB1 and CB2), at 200 Hz, reshaped into $(C, S, 200)$ where S is the number of 1-second patches, in units of 100 μV.

EEGPT [36] is evaluated on TUAB and TUEV with the settings of its official TUAB fine-tuning script (5 warm-up epochs, 50 epochs, head dropout 0.5, no early stopping), on the same windows as the other baselines. The checkpoint is selected on validation accuracy on TUAB, as in that script, and on validation weighted F1 on TUEV, as for LaBraM. On both tasks, the main differences from that script are the following. A newly initialized LayerNorm is applied to the encoder output (the official script applies none). The effective batch is 100 instead of 200. Training uses fp32 (32-bit floating point) with gradient clipping at 1.0, instead of fp16 (16-bit floating point) without clipping. The cosine schedule is stepped per epoch instead of per iteration, after a warm-up that starts at 10^{-4}. On TUAB, the inputs are 6 s windows (1,200 samples) of 19 channels after MENDR's preprocessing (filtering and 500 μV rejection), instead of 10 s windows of 23 channels. On TUEV, a second setting follows its official TUEV script (30 epochs, effective batch 800, head dropout 0.8, learning rate warmed up from 0 and decayed per iteration to 10^{-6}, no gradient clipping), in 32-bit instead of 16-bit precision. The stronger of the two settings by test mean, the TUAB-script setting, is reported. EEGPT's channel-mapping block (a learned $1{\times}1$ convolution followed by a depthwise temporal convolution) maps the 19 input channels onto its 20 named channels. Inputs are in $\mu\text{V}/100$, and the newly initialized classifier head is sized to the encoder's output. The layer-wise learning-rate decay of 0.65 in the official script applies to none of the model's parameters, so all parameters train at the same learning rate.

LUNA-Base [9] is fine-tuned from the publicly released checkpoint on our six benchmarks under the strict protocol. We use the optimizer and schedule of the fine-tuning configuration released with LUNA's code (AdamW, base learning rate 5×10^{-4}, layer-wise learning-rate decay 0.75, weight decay 0.05, 10-epoch linear warm-up followed by cosine decay) for 30 epochs (50 on SEED-V). LUNA's paper lists other values. The released configuration uses an effective batch of 1,024 (256 on each of four GPUs), samples batches uniformly, does not clip gradients, trains in bf16 (16-bit brain

floating point) mixed precision, and steps the schedule per iteration. It stops early on validation loss (patience 8) and keeps the lowest-loss epoch. We instead draw class-balanced batches, clip gradients at norm 1.0, train in fp32, step the schedule per epoch, run every epoch, and keep the one that is best on validation AUROC or weighted F1. On TUAB, TUEV, and PhysioMI, we use a batch size of 256, resample our windows to 256 Hz, and z-score each channel within each window, as in LUNA's preprocessing (on our referential channels, not its bipolar pairs). On SEED-V, CHB-MIT, and ISRUC, we use a batch size of 64 with our 200 Hz input, not z-scored. The loss is unweighted cross-entropy. We randomly initialize the classifier head (0.53M parameters) and load the backbone (6.6M, Table 1) from the released safetensors. LUNA's paper reports results on only two of our tasks: 0.887 AUROC (TUAB) and 0.339 wF1 (SEED-V).

LUNA-Base on SEED-V. LUNA-Base's SEED-V result is the best of seven configurations: six around its released optimizer settings (three seeds each) and one with the preprocessing described in LUNA's paper (256 Hz, z-scored) (five seeds, Table 10). The strongest by test mean (learning rate 5×10^{-4}, 50 epochs) is the one in Table 1. Selecting it by test mean favors LUNA. MENDR is ahead of it (Table 2), but with three LUNA seeds this comparison has limited power.

Table 10: The other LUNA-Base configurations evaluated on SEED-V (test wF1, mean $\pm$ std, 3 seeds unless marked). The strongest result (lr 5e-4 for 50 epochs) is in Table 1. The configurations listed here train for 30 epochs, and the paper-preprocessing row uses batch size 256. lr: learning rate, wd: weight decay.

Configuration	Test wF1
lr 1e-3	$0.349_{\pm.015}$
lr 1e-4	$0.340_{\pm.003}$
lr 3e-4	$0.338_{\pm.018}$
lr 3e-4, wd 0.1	$0.354_{\pm.018}$
lr 5e-4 (released optimizer settings)	$0.344_{\pm.009}$
LUNA's paper preprocessing (256 Hz, z-scored), official coordinates (5 seeds)	$0.330_{\pm.018}$

Notes on Table 1 entries. CBraMod's paper reports 0.923 / 0.834 on TUAB / TUEV, and LaBraM-Base's paper reports 0.902 / 0.831, so the LaBraM-Base TUEV entry is also below its reported value. Table 1 also reports the strongest of several configurations for LaBraM-Base on PhysioMI (other configuration: 0.601) and CHB-MIT (with layer decay: 0.944), for CBraMod on PhysioMI (three others: 0.605 to 0.619), and for LUNA-Base on CHB-MIT (0.945) and ISRUC (0.801). BIOT's CHB-MIT entry uses a fixed input scaling, which scores above BIOT's published per-window normalization (0.867 over the same ten seeds). With zero-padding to 10 s, BIOT on PhysioMI is not significantly above chance (0.268 over the same five seeds, one-sided $p = 0.11$ against 0.25).

Checkpoints are selected on validation AUROC on TUAB and CHB-MIT and on weighted F1 otherwise, with these exceptions: LaBraM-Base and EEGPT use accuracy on TUAB, and CBraMod uses Cohen's κ on TUEV and macro F1 on SEED-V and ISRUC. On CHB-MIT and ISRUC, every baseline draws its training batches with a class-balanced sampler (weights $1/n_c$, with replacement). In addition, BIOT on both tasks and CBraMod on ISRUC weight the cross-entropy of a class with n_c training samples by $1/\sqrt{n_c}$. LaBraM-Base computes the same weights after balanced sampling, so they are close to 1. CBraMod on CHB-MIT uses an unweighted loss. LaBraM-Base uses label smoothing of 0.1 on every task except TUAB; CBraMod and EEGPT use it on their multi-class tasks, and BIOT uses it on CHB-MIT and ISRUC. Gradients are clipped at norm 1.0 except for BIOT outside TUAB and LaBraM-Base on TUEV. LaBraM-Base (except on TUEV) steps its schedule per epoch, whereas its official code steps per iteration. BIOT on TUAB decays its learning rate by a per-epoch cosine, whereas its official code keeps it constant. BIOT trains in fp32 and CBraMod under bf16 autocast. LaBraM-Base trains in fp32 on TUAB and CHB-MIT, under bf16 autocast on TUEV and in fp16 mixed precision otherwise. All fp32 matrix products use TensorFloat-32 (TF32).

E MENDR Fine-tuning Protocol and Hyperparameters

MENDR uses a two-stage fine-tuning protocol on every downstream benchmark, at most 50 epochs in total. **Stage 1 (head only, epochs 1–5**). Only the classification head is trained. It is a two-layer MLP on the attention-pooled spatial features and the band-averaged temporal-path features. Everything

else is frozen, including parts that are newly initialized or not pretrained: a re-initialized projection, the spatial and temporal attention pooling and, on SEED-V, the new temporal path and the reset filters. The head is trained at 10 times the base learning rate, warmed up linearly over three epochs and then decayed by a cosine schedule. This keeps the randomly initialized head from corrupting pretrained features. **Stage 2 (end-to-end, up to 45 epochs).** A new AdamW optimizer trains the encoder at 0.5 times and the head at 5 times the base learning rate of 10^{-4}. Weight decay is 0.05 on every parameter (including the filter cutoffs, except on SEED-V). Training uses a cosine schedule stepped per epoch, gradient clipping at norm 1.0, and batch size 64. Training uses early stopping on the validation score (AUROC on TUAB and CHB-MIT, weighted F1 otherwise) with patience 15, counted from the first epoch. The checkpoint with the best validation score is evaluated once on the test set.

Loss and sampling. On every task, the loss is cross-entropy with label smoothing 0.1 and class weights proportional to $1/\sqrt{n_c}$ for a class with n_c training samples, normalized to mean 1. On TUAB, TUEV, SEED-V, CHB-MIT and ISRUC, training batches are also drawn by a class-balanced sampler (sample weights $1/n_c$, with replacement). PhysioMI training data are shuffled uniformly.

Augmentation (TUAB, TUEV and PhysioMI, training batches only). Each segment is shifted in time by a random number of samples, up to 5% of its length in either direction, with the edge sample repeated as padding, and each channel is zeroed independently with probability 0.1 (never all channels).

BatchNorm statistics at evaluation. One temporal network serves all K bands, so its BatchNorm running statistics mix the bands. Each band is therefore normalized with its own statistics at evaluation. Before every validation and test pass, the mean and variance of each feature channel are computed per band over 50 shuffled training batches without augmentation. The running statistics are not used at evaluation.

Precision. Training runs in 32-bit floating point, with TF32 matrix products and convolutions on the GPU, and the eigendecomposition and its backward pass run in 64-bit floating point.

From-scratch models (Tables 5–7, 12, 13, 15 and 16) skip stage 1 and train the encoder at 10^{-4} and the head at 10^{-3} from the start, with the same warm-up and cosine schedule, for up to 50 epochs.

We set the covariance window, augmentation, projection initialization, and, for SEED-V, filter initialization and temporal path per task (Section 4.1), and share everything else. Hyperparameters are in Table 11.

Table 11: MENDR architecture and training hyperparameters (defaults). LR: learning rate. Per-task windows, augmentation, and SEED-V's temporal path are in Section 4.1. SEED-V's temporal path (146,112 parameters) has parallel convolutions with kernels 7, 31, and 65 followed by a depthwise-separable convolution (kernel 15), a two-layer transformer over 8 segments, and attention pooling.

Architecture		*Pretraining*	
Spatial dim (d_s)	32	Optimizer	AdamW
Cross-attn key dim (d_k)	64	Learning rate	5×10^{-4}
Cross-attn heads (H_{ca})	4	Weight decay	0.05
Num filters (K)	4	Batch size	128
Filter kernel length	65	Epochs	30
Window samples (w)	64	Warmup epochs	3
Stride samples (s)	32	LR schedule	Cosine
SPD regularization (ϵ)	10^{-3}	Gradient clip norm	1.0
Tangent dim (d_v)	528	Mask ratio (ρ)	0.5
Transformer dim (d_m)	128	α (Fréchet)	0.5
Transformer layers (L)	4	β (temporal)	0.1
Attention heads (H)	4	γ (diversity)	1.0
Temporal dim (d_t)	64	Temporal masked spans	4×32 samples
Temporal kernel	15	Temporal decoder kernel	15
Temporal layers	2	*Fine-tuning*	
Dropout	0.1	Encoder LR mult	$0.5\times$
Spatial encoder params	**1,218,696**	Classifier LR mult	$5\times$
Temporal path params	**34,177**	Head-only epochs (stage 1)	5

Covariance window length. No window length between 32 and 200 samples differs significantly from the default 64 on either task (Table 12, paired t-tests by seed, all $p \geq 0.10$).

Table 12: Covariance window length, MENDR from scratch, default configuration (mean ± std, 5 seeds). Stride is half the window.

Window (samples)	CHB-MIT (AUROC)	ISRUC (wF1)
32 (0.16 s)	$0.977_{\pm.005}$	$0.751_{\pm.013}$
64 (0.32 s, default)	CA, from scratch, Table 5	
128 (0.64 s)	$0.986_{\pm.013}$	$0.764_{\pm.023}$
200 (1.00 s)	$0.984_{\pm.010}$	$0.754_{\pm.018}$

F Filter Evolution During Pretraining

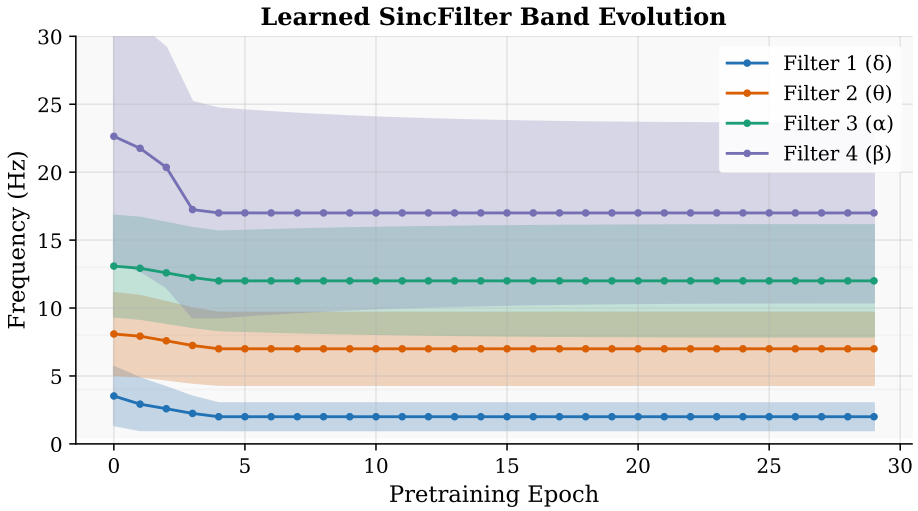


Figure 4: Per-band lower and upper sinc-filter cutoff frequencies tracked across 30 pretraining epochs. (line: center frequency, colored band: passband between the cutoffs, legend letters: the canonical clinical band each filter starts near).

After five epochs, the center frequencies are at 2, 7, 12, and 17 Hz and stay there. They are 5 Hz apart (equal to f_{sep}), and the lowest filter is at the smallest cutoff and bandwidth that Eq. (4) allows. After that, only the cutoffs of the two upper filters move (about 0.5 and 1 Hz), ending at 1.0–3.0, 4.3–9.7, 7.9–16.1 and 10.4–23.6 Hz (Figure 4). Filters 2–3 and 3–4 overlap. Pretraining applies weight decay to the cutoffs, as to every other parameter.

G Confusion Matrices

Figure 5 shows the confusion matrices of the Table 1 MENDR models on the four multi-class tasks. The test confusion matrices of the five seeds (on PhysioMI, of all five folds of each seed) are summed, and each row is normalized. On TUEV, spike and sharp wave (SPSW) segments are more often classified as one of the two periodic-discharge classes (GPED, PLED) than as SPSW, and 29% of artifact (ARTF) segments are labeled background. On PhysioMI, both-fists and both-feet trials are most often confused with each other. On ISRUC, N1 is the weakest stage and is most often labeled REM or N2. On SEED-V the errors are spread over all classes, and disgust is the most frequent wrong label.

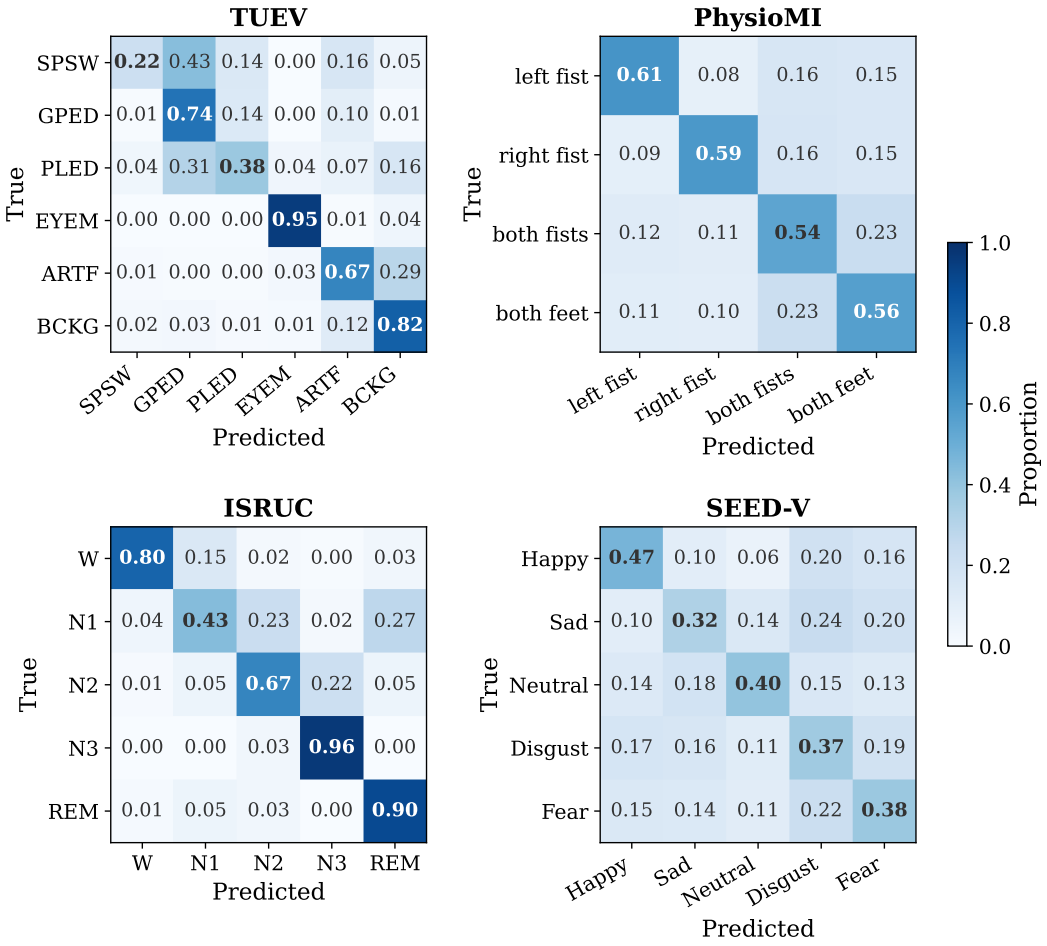


Figure 5: Row-normalized test confusion matrices of the Table 1 MENDR models, summed over five seeds (PhysioMI: over folds and seeds).

H Spatial Saliency on Motor Imagery

Gradient saliency measures how much the model's score for a class would change if the signal at one electrode changed slightly. It shows which electrodes the trained model is sensitive to, not which electrodes carry the most signal. For each correctly classified PhysioMI test trial $\mathbf{x}$ of class y, we take the absolute gradient of the class-y logit with respect to the input, $|\partial \mathrm{logit}_y / \partial \mathbf{x}|$, and sum it over time, which gives one value per electrode. Averaging these values over the trials of each class, pooled over the held-out subjects of all 5 folds, gives one saliency map per class. The maps use the seed-789 models trained with the configuration in Table 1. Saliency is computed on CPU, where fold scores agree with the GPU evaluation to within 0.006.

In the class maps pooled over folds, C3, over the left sensorimotor hand area [30], is the most salient electrode for all four classes. The next ranks mix sensorimotor (C4, CP3, CP4, Cz) and peripheral sites (F7, F8, Iz), with CP3 fourth for both the two-fist and both-feet classes. The class maps are similar (pairwise correlation 0.87–0.96). The both-feet map is higher than the two single-fist maps at all 64 electrodes and higher than the both-fists map at 62 of them. The largest left–right difference is less than half of the largest feet–fists difference (Figure 6). Single-fold maps vary: C3 is the top electrode in 13 of the 20 fold–class maps.

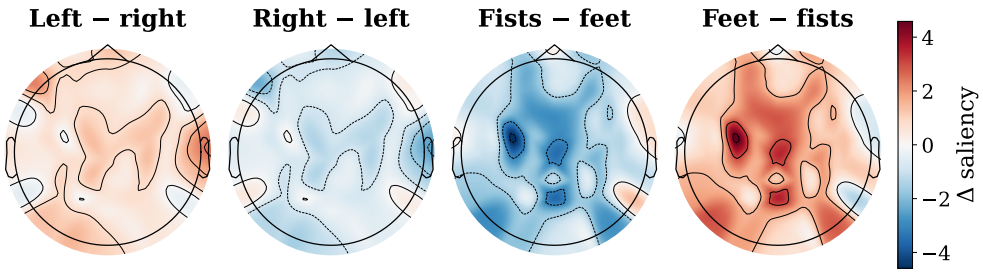


Figure 6: PhysioMI gradient-saliency differences between paired classes. Each panel subtracts one class map from another (left: left fist, right: right fist, fists: both fists, feet: both feet). Red marks electrodes to which the first class's score is more sensitive. Each pair is shown in both orders, so the second panel of a pair is the first with its sign reversed. Feet − fists is positive at 62 of 64 electrodes, largest at C3 and POz.

I SEED-V Transfer Controls

In Table 13, fine-tuned models use the Table 1 SEED-V configuration, and the frozen probes use the default configuration. To test whether the mismatch between SEED-V's montage and the pretraining montage limits transfer, we also present SEED-V through the 19 electrodes of the pretraining montage, in the pretraining channel order. This improves no fine-tuned model, and the Table 1 pretrained model (projection re-initialized) and the from-scratch model do not differ significantly on either montage (on 62 channels Δ_{pre} in Table 5, on 19 electrodes -0.003 with p=0.595). Loading the pretrained projection as well ("full encoder") scores below re-initializing it on 62 channels (-0.022, p=0.021). On 19 electrodes, they do not differ significantly (p=0.102). A frozen pretrained encoder with a trained head is close to five-class chance with either montage (balanced accuracy 0.21–0.24 against 0.20). The TUH representation therefore carries little linearly decodable information about SEED-V emotion labels, and the fine-tuned models' SEED-V scores come from fine-tuning.

Table 13: SEED-V transfer controls (test wF1, mean ± std, 5 seeds): the 62-channel montage versus the 19 pretraining electrodes. Δ = 19 − 62, paired t-test by seed.

Setting	62 channels	19 electrodes	Δ
from scratch	CA, Table 5	$0.381_{\pm.009}$	-0.010 (p=0.119)
pretrained, projection re-initialized	Table 1	$0.377_{\pm.006}$	-0.011 (p=0.017)
pretrained, full encoder	$0.367_{\pm.007}$	$0.363_{\pm.012}$	-0.004 (p=0.634)
frozen pretrained encoder (probe)	$0.151_{\pm.016}$	$0.170_{\pm.009}$	$+0.018$ (p=0.064)

J Representation Analyses

J.1 Reconstruction Quality

Temporal path. On 178 pretraining segments, the temporal decoder's error on masked samples is 0.37 of the target variance (mean over bands), against 0.77 for the better of linear interpolation and zero prediction in each band. In the lowest band, interpolation is better (0.07 against 0.18).

Spatial path. We measure masked-token reconstruction on 64 TUH segments from the pretraining corpus at three mask ratios (Table 14). Tokens and segments are not selected by reconstruction quality. R^2 is computed over the masked tokens in the tangent space, relative to three references: the mean of all elements (pooled), the mean of each tangent coordinate, and each sample's own mean token. The pooled value is high partly because diagonal and off-diagonal coordinates differ in level (Figure 7). The model explains 0.67 of each coordinate's variance at 25% masking and 0.47 at 75%. An untrained model is below zero for all three references.

Table 14: Masked-token reconstruction R^2 of the pretrained model, by reference, and R^2 of the Fréchet-mean prediction (pooled).

Mask ratio	Pooled	Per coordinate	Within sample	Fréchet mean
25%	0.983	0.667	0.570	0.999
50%	0.978	0.588	0.468	0.999
75%	0.972	0.470	0.314	0.998

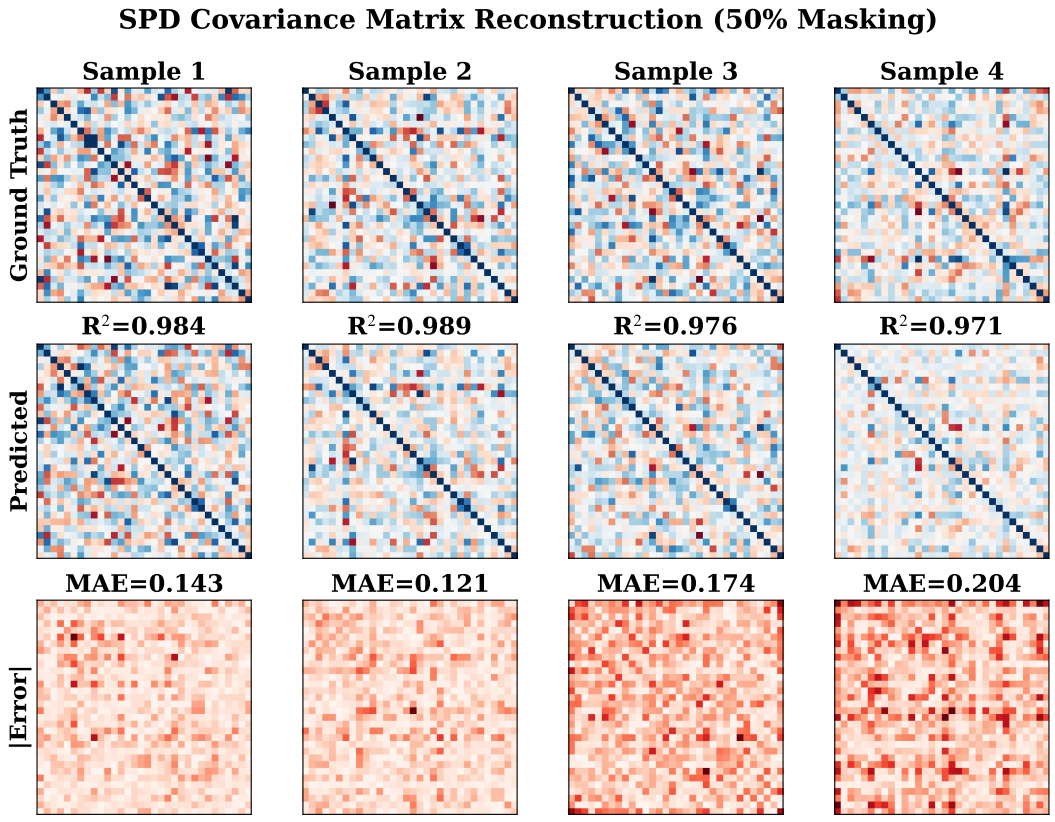


Figure 7: One masked token from each of four segments at 50% masking, in the tangent space: ground truth, prediction, and absolute error. MAE in the panels: mean absolute error over the token's tangent coordinates. The printed R^2 is relative to the mean over that token's coordinates (pooled type, Table 14).

J.2 Dual-Path Representation Geometry Across Datasets

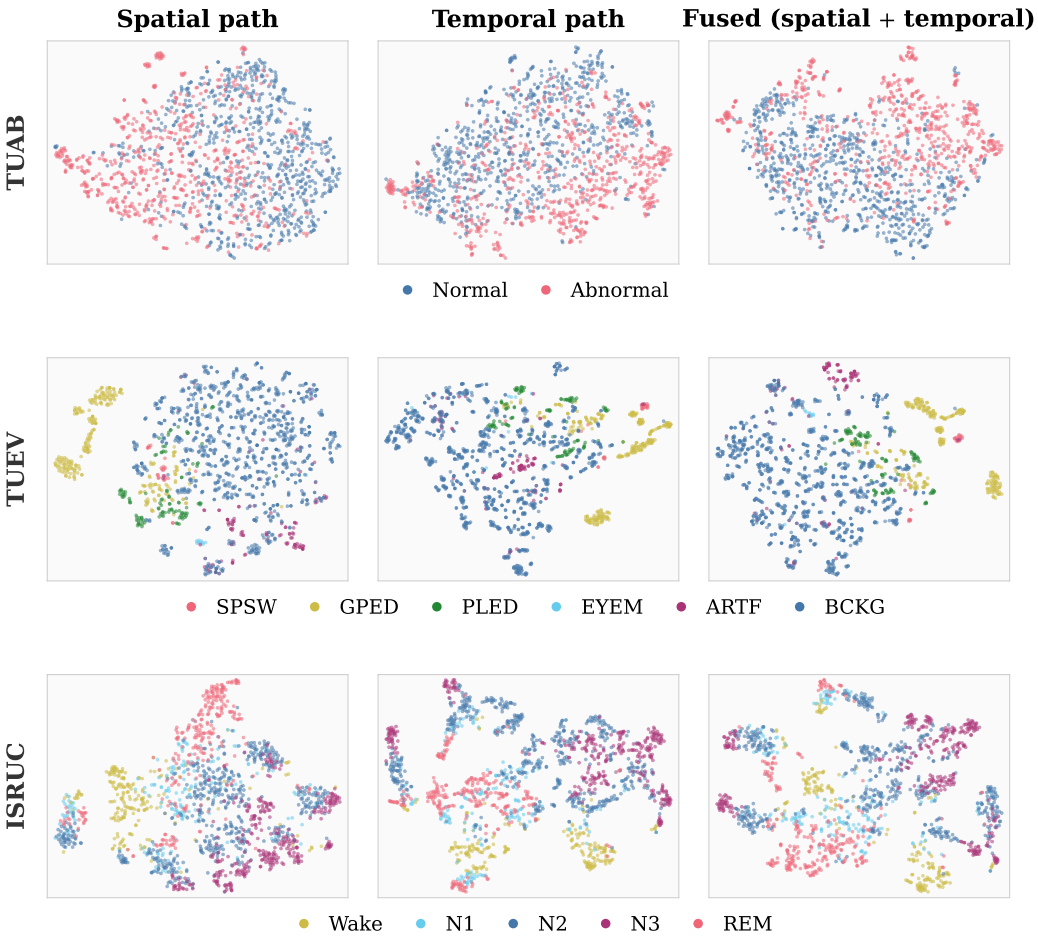


Figure 8: t-distributed stochastic neighbor embedding (t-SNE) of the spatial-path, temporal-path, and fused representations on TUAB, TUEV, and ISRUC test sets (n=1,500 each).

Figure 8 shows the path features of models fine-tuned with the configuration in Table 1 (seed 789, features computed on CPU, test scores within 0.003 of the GPU evaluation). On TUAB, the 2-D embedding of the spatial-path features separates the classes more than that of the temporal-path features (silhouette 0.15 against 0.09, fused 0.13), although in the feature space itself all three silhouettes are about 0.08. On ISRUC, all three are close to zero. On TUEV, the silhouette is negative in every panel. Background is 998 of the 1,500 points. A 2-D silhouette of a jointly trained model's features does not show how well each path classifies on its own. Table 6 measures that directly on TUAB and TUEV by training each path alone, from scratch.

J.3 Collapse of the Larger Encoders

Before fine-tuning, the pretrained Base and Large spatial transformers of Table 7 give the same output for every input. Across positions and inputs, their output varies by about 10^{-7} relative to its size, also when every token is visible as in fine-tuning. Their masked-token error equals that of predicting one constant token. The Tiny, Default, and Small encoders are not collapsed.

K Spatial Projection Analyses

K.1 Channel Perturbation

Table 15: Test-time channel perturbation on TUAB (AUROC, mean ± std over 5 seeds). These models are trained separately from those of Tables 1 and 5, so their unperturbed scores can differ slightly. Other conditions are in Table 16.

Model	Unperturbed	Permuted	Reversed	Drop 2	Drop 8
MENDR (Table 1 configuration)	$0.902_{\pm.001}$	$0.721_{\pm.018}$	$0.776_{\pm.010}$	$0.804_{\pm.008}$	$0.740_{\pm.043}$
MENDR, from scratch	$0.899_{\pm.005}$	$0.703_{\pm.028}$	$0.789_{\pm.018}$	$0.815_{\pm.020}$	$0.543_{\pm.085}$

Table 15 gives the perturbations discussed in Section 4.6, and Table 16 gives the other conditions for the same two models. Permutation, subset, and channel-removal conditions use 10 random draws per seed, the per-channel gain and noise conditions use 5, and the other conditions are deterministic. A common average reference changes neither score to three decimals (not tabulated). Per-channel gain and 10 dB noise cost little, while a tenfold amplitude change, 0 dB noise, and channel subsets cost more.

Table 16: Test-time perturbation on TUAB (AUROC, mean ± std over 5 seeds, Δ from the unperturbed score). SNR: signal-to-noise ratio.

	Table 1 configuration		From scratch	
Condition	AUROC	Δ	AUROC	Δ
75% of channels kept	$0.782_{\pm.017}$	-0.120	$0.753_{\pm.023}$	-0.145
50% of channels kept	$0.700_{\pm.054}$	-0.202	$0.569_{\pm.071}$	-0.330
amplitude $\times 10$	$0.802_{\pm.023}$	-0.101	$0.808_{\pm.008}$	-0.091
per-channel gain	$0.882_{\pm.004}$	-0.020	$0.877_{\pm.004}$	-0.021
noise, 10 dB SNR	$0.878_{\pm.007}$	-0.024	$0.854_{\pm.023}$	-0.044
noise, 0 dB SNR	$0.817_{\pm.009}$	-0.085	$0.770_{\pm.043}$	-0.128

K.2 Projection Measurements

Table 8 uses models fine-tuned with the Table 1 configuration (seed 789, PhysioMI fold 0), applied to 512 training segments. With $\mathbf{A}$ and $\mathbf{D} = \mathbf{A} - \mathbf{1}\bar{\mathbf{a}}^\top$ as in Section 3.1, the columns are defined as follows. Max dev. is $\max_{ij} |A_{ij} - 1/C|$ divided by $1/C$, in the median segment. In the worst segment of any task, it is at most 8.2%. All other columns use the segment-averaged matrices. The effective rank is the exponential of the entropy of the normalized singular values. It is at most $\min(d_s, C)$ for $\mathbf{A}$ and $\min(d_s, C) - 1$ for $\mathbf{D}$. Size of $\mathbf{D}$ is $\|\mathbf{D}\|_F / \|\mathbf{1}\bar{\mathbf{a}}^\top\|_F$. Before fine-tuning, the projection is the pretrained projection for TUEV and, for the other tasks, the randomly initialized projection from the same run (the rest of the encoder is pretrained). $\mathbf{D}_{\text{start}}$ is computed on the same 512 segments. Size ratio is $\|\mathbf{D}\|_F / \|\mathbf{D}_{\text{start}}\|_F$ (0.9 to 5.4×), and similarity is the cosine between $\mathbf{D}$ and $\mathbf{D}_{\text{start}}$, flattened to vectors (0.68 to 0.82).

The LayerNorm after the projection divides by $\sqrt{\text{variance} + \epsilon_{\text{LN}}}$, with the variance taken over the d_s virtual electrodes at each time step. This variance is far below $\epsilon_{\text{LN}} = 10^{-5}$ (median at most 0.3% of ϵ_{LN}, and below ϵ_{LN} at 99.3% or more of time steps). The LayerNorm is therefore almost a fixed affine map. It outputs $\mathbf{DX}$ times its gain divided by $\sqrt{\epsilon_{\text{LN}}}$, plus its bias. These quantities are computed in float32 on CPU (training and evaluation use TF32 matrix multiplication on GPU).

L Tangent-Space Classification Baseline

To see how much the covariance features achieve without the learned model, we fit a Log-Euclidean tangent-space classifier on MENDR's own TUAB and TUEV segments. Each segment gives one covariance over the 19 channels, normalized as in Eq. (5). We then map it via the matrix logarithm, vectorize it, standardize it, and classify it with a logistic regression trained on all training segments, with class weighting and regularization selected on validation.

This single deterministic fit reaches AUROC 0.801 on TUAB and weighted F1 0.696 on TUEV, both below MENDR's spatial path trained alone (Table 6, one-sample t-test over the five seeds: $p < 0.001$ on TUAB, $p < 0.001$ on TUEV). The spatial path differs from this baseline in windowing, bands, projection, and transformer, and the comparison does not separate them.

M Compute Cost

Table 17 measures the cost of each model in its TUAB configuration (Table 1, and Section 4.1 for EEGPT) on 6-s windows, each in its own input format (medians of 50 timed forward passes after warm-up, one run per setting). The CPU is a single thread on an Intel Xeon 6972P, and the GPU is an NVIDIA H100. On the GPU, all models run with 32-bit weights, TF32 matrix products, and, for MENDR, 64-bit eigendecompositions. This matches the precision of their TUAB fine-tuning runs, except for CBraMod, which fine-tunes under bf16 autocast. MENDR has the fewest parameters and the second-lowest number of floating-point operations (FLOPs) but is not cheaper to run. On the CPU, it is slower per window than BIOT, LUNA-Base, CBraMod, and LaBraM-Base and faster only than EEGPT. On the GPU, it has the lowest inference throughput of the six models at batch sizes 32, 64, and 256. At batch size 256, about two-thirds of its time goes to its 144 double-precision eigendecompositions per window (56% at batch size 64). At batch size 1, every model is limited by kernel launches, so we do not rank them there.

Table 17: Compute cost per 6-s window. Params as in Table 1 (EEGPT: Section 4.1). GFLOPs include MENDR's eigendecompositions.

Model	Params	GFLOPs	CPU ms (batch 1)	GPU windows/s (batch 32 / 256)
MENDR	1.2M	0.73	24.3	3,142 / 5,441
BIOT	3.2M	1.14	13.9	8,274 / 33,577
LaBraM-Base	5.8M	1.45	21.6	7,299 / 11,916
CBraMod	4.9M	1.00	19.3	6,293 / 27,629
LUNA-Base	6.6M	0.61	15.3	4,159 / 26,669
EEGPT	25.3M	21.94	160.0	5,487 / 6,240

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the contributions (an integration of Riemannian SPD tokens with masked-autoencoder pretraining, a Perceiver projection that is flexible in channel count at fine-tuning time, and parameter efficiency under a strict protocol with controlled attribution), each supported by experiments: benchmark comparison in Section 4.2, the contribution of pretraining in Section 4.3, the component ablation in Section 4.4, scale in Section 4.5, and channel transfer and its limits in Section 4.6. Claims are scoped to the six benchmarks evaluated and the TUH pretraining corpus.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The *Limitations* paragraph in Section 5 states seven limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper is primarily empirical. The appendix gives the Daleckii-Kreĭn backward formula for the matrix logarithm, a known result we cite (Appendix A, Eqs. (11)–(12)), and proves the positive-definiteness guarantee under trace-normalization plus $\epsilon\mathbf{I}$ regularization (Appendix B). All assumptions (eigenvalue clamping, near-degeneracy via L'Hôpital) are explicitly stated.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3 fully specifies the architecture (10 numbered equations) and pretraining objective. Appendix E (Table 11) lists every architecture and training hyperparameter. Appendix C documents data splits and preprocessing for all six benchmarks. Appendix D lists baseline fine-tuning hyperparameters. Every reported cell is a mean over fixed seeds, five unless marked in the table.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The pretraining corpus (TUH EEG Corpus) and all six benchmarks (TUAB, TUEV, PhysioMI, SEED-V, CHB-MIT, ISRUC) are publicly available. The TUH EEG Corpus, TUAB and TUEV are available from the Neural Engineering Data Consortium (`https://isip.piconepress.com/projects/nedc/html/tuh_eeg/`), PhysioMI (`https://physionet.org/content/eegmmidb/1.0.0/`) and CHB-MIT (`https://physionet.org/content/chbmit/1.0.0/`) from PhysioNet, SEED-V from the BCMI laboratory (`https://bcmi.sjtu.edu.cn/home/seed/seed-v.html`) and ISRUC-Sleep from `https://sleeptight.isr.uc.pt/`. Preprocessing details are in Appendix C. Code and pretrained checkpoints are available at the URL in the footnote on page 1.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The pretraining objective and protocol are described in Section 3.5, and the two-stage fine-tuning protocol (head-only epochs on the frozen encoder followed by end-to-end training with a lower encoder learning rate) is described in Appendix E, with complete pretraining and fine-tuning hyperparameters in Table 11. Train/eval splits are documented per dataset in Appendix C, and baseline hyperparameters are in Appendix D. The fine-tuning configuration was chosen on validation per task (Section 4.1).

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All downstream results (Tables 1, 7) are reported as mean $\pm$ standard deviation over 5 random seeds (training seeds on fixed data splits), with exceptions marked in the tables. Unless stated otherwise, tests are two-sided. Cross-model comparisons use Welch t-tests, and within-MENDR comparisons that share seeds use paired t-tests, including the component ablation (Section 4.4). The comparisons in Table 7 use Welch t-tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Every training run in the paper used one NVIDIA H100 (80 GB) or H200 (141 GB) GPU, and the CPU timings in Appendix M ran on an Intel Xeon 6972P node. Pretraining: 30 epochs on TUH (about 13,300 hours of EEG, of which each epoch draws about 765) took about 15 h on one H100. A Table 1 fine-tuning run (one task and seed, PhysioMI with all five folds) takes between 2 min and 2.8 h on one GPU (median 24 min). Total project compute, including failed, preliminary and exploratory runs that are not in the paper, was approximately 8,400 GPU-hours on H100 and H200 GPUs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. We use only publicly released, de-identified EEG datasets governed by their original data-use agreements, and no new human data is collected. We do not deploy the model in any clinical decision-making setting, and we discuss the limits of cross-domain transfer (Section 5).

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: *Positive impacts:* a small EEG foundation model with released code and weights lowers the barrier to research on automated screening for abnormality, seizure, and sleep events. *Negative impacts and mitigations:* MENDR is a research model and has not been validated for clinical use, and it could be misused as a sole diagnostic tool. The paper reports where it is behind other models (sleep staging, motor imagery, emotion recognition, and event classification) and its sensitivity to channel changes, so users do not over-extrapolate. We do not release any patient-identifying data.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: MENDR is a small SPD-manifold model (1.2M-parameter spatial encoder) for EEG representation learning. It does not generate text, images, or audio, has no plausible misuse path (e.g., disinformation, surveillance), and is trained only on de-identified EEG already released under research-use licenses. Therefore, no additional release safeguards are required.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets are properly cited (TUH [26], TUAB [23], TUEV [11], PhysioMI [32, 10], SEED-V [22], CHB-MIT [33], ISRUC [20]) and used under their respective research-use agreements (the TUH data-use agreement, PhysioNet's Open Data Commons Attribution (ODC-By) 1.0 license, and the providers' terms for SEED-V and ISRUC). We evaluate baselines (BIOT [39], LaBraM [18], CBraMod [37], EEGPT [36], LUNA-Base [9]) using the officially released checkpoints under their open-source licenses (MIT/Apache-2.0). Architectural inspirations (Perceiver [16], SincConv [31], masked autoencoders [12], AdamW [24]) are cited in Sections 2 and 3.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Two new assets are released: (i) the MENDR pretrained encoder (a 1.2M-parameter spatial encoder and a 34K-parameter temporal path, trained on TUH for 30 epochs), fine-tuned weights for every Table 1 MENDR task and seed (trained with the Table 1 configurations, with task means within 0.003 of Table 1), and (ii) the open-source training/fine-tuning codebase. Both include a structured `README` documenting the architecture, training command lines, hyperparameter values, expected output formats, and evaluation scripts for each downstream task. The release URL is in the footnote on page 1.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing or new research with human subjects. We exclusively use pre-existing publicly released EEG datasets that were collected by the original dataset authors under their own ethics approvals.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: We did not collect or interact with new human-subject data. All EEG datasets used (TUH, TUAB, TUEV, PhysioMI, SEED-V, CHB-MIT, ISRUC) were collected and de-identified by their original authors under their own institutional review processes, and we use only the publicly released versions.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: No large language model is used as part of the MENDR architecture, training procedure, or evaluation pipeline. The core method is a Riemannian SPD-manifold masked autoencoder for EEG.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.