

Supplementary Information

Algorithm S1: Training Procedure for Deep-SVM Hybrid Model (inner-loop view)

Input: Outer-train epochs $\mathcal{D}_{\text{train}}$ (subject-wise), labels $\mathbf{y}$

Output: Trained SVM $\mathcal{M}_{\text{SVM}}$, scaler $\mathcal{S}$, PCA $\mathcal{P}$

1. Perform stratified inner CV on $\mathcal{D}_{\text{train}}$ (subject-wise where applicable).
 2. For each inner split:
 - (a) Fit z-score scaler $\mathcal{S}$ on **inner-train** only; transform inner-train.
 - (b) Apply **SMOTE** on **inner-train only** (no SMOTE on inner-val).
 - (c) Fit PCA $\mathcal{P}$ on the oversampled inner-train; transform inner-train and inner-val (no refit).
 - (d) Grid-search SVM (C, γ) using inner-val performance (macro-F1).
 3. Refit $\mathcal{S}$ and $\mathcal{P}$ on the full outer-train; apply SMOTE on outer-train only; train SVM with best (C, γ) .
-

Algorithm S2: Inference Procedure for Deep-SVM Hybrid Model

Input: Unseen EEG signals $\mathbf{X}_{new}$, CNN parameters Θ , PCA matrix $\mathbf{P}$, trained SVM model $\mathcal{M}_{SVM}$

Output: Predicted label $\hat{y}$

1. Preprocess $\mathbf{X}_{new}$ (filtering, ASR, interpolation).

2. Extract features via 1D-CNN:

$$F_{CNN}^{new} = f_{CNN}(\mathbf{X}_{clean}; \Theta)$$

3. Normalize features:

$$F^{new} = \frac{\mathbf{F}^{new} - \mu(F_{train})}{\sigma(F_{train})}$$

4. Apply PCA projection:

$$F_{PCA}^{new} = \mathbf{P} \cdot F^{new}$$

5. Predict label:

$$\hat{y} = \arg \max_c \mathcal{M}_{SVM}(F_{PCA}^{new})_c$$

return $\hat{y}$

Table S1. Inner-CV sensitivity to PCA dimensionality and SMOTE neighbors (mean $\pm$ SD macro-F1 over inner folds). Best setting in bold.

SMOTE k	PCA dimensionality k					
	64	96	128	160	192	256
3	0.92 $\pm$ 0.02	0.93 $\pm$ 0.02	0.94 $\pm$ 0.02	0.93 $\pm$ 0.03	0.92 $\pm$ 0.03	0.91 $\pm$ 0.03
5	0.93 $\pm$ 0.02	0.94 $\pm$ 0.02	0.95 $\pm$ 0.01	0.94 $\pm$ 0.02	0.93 $\pm$ 0.02	0.92 $\pm$ 0.03
7	0.92 $\pm$ 0.03	0.93 $\pm$ 0.02	0.94 $\pm$ 0.02	0.93 $\pm$ 0.02	0.92 $\pm$ 0.03	0.91 $\pm$ 0.03
9	0.91 $\pm$ 0.03	0.92 $\pm$ 0.03	0.93 $\pm$ 0.02	0.92 $\pm$ 0.03	0.91 $\pm$ 0.03	0.90 $\pm$ 0.04

Given outer-fold macro-F1 scores for the proposed model $\{m_i\}_{i=1}^K$ and a baseline $\{b_i\}_{i=1}^K$ (with $K = 5$ outer folds), draw 10,000 bootstrap resamples of fold indices with replacement. For each resample r , compute $\bar{\Delta}^{(r)} = \frac{1}{K} \sum_{i \in \mathcal{J}(r)} (m_i - b_i)$. The two-sided percentile p -value is the fraction of $\bar{\Delta}^{(r)}$ whose sign is opposite to the observed $\bar{\Delta}$ (times two). We also report the 95% bootstrap CI for $\bar{\Delta}$ (percentile method).

Table S2. Class counts before/after SMOTE within inner-train splits for each outer fold. Validation/test sets are never oversampled.

Outer fold	Before SMOTE (inner-train)			After SMOTE (inner-train)		
	AD	FTD	CN	AD	FTD	CN
Fold 1	242	210	226	242	242	242
Fold 2	240	208	223	240	240	240
Fold 3	241	209	224	241	241	241
Fold 4	239	207	222	239	239	239
Fold 5	240	209	223	240	240	240

Table S3. Paired bootstrap comparison (10,000 resamples) of macro-F1 versus the strongest reproduced baseline across the outer folds; percentile 95% CI and two-sided p -value reported.

Model comparison	$\bar{\Delta}$ (macro-F1)	95% CI	p -value
Proposed (Deep-SVM) vs. Best Baseline (reprod.)	0.043	[0.018, 0.068]	0.001

Table S4. Inner-CV search spaces and selected settings (mode across folds).

Component	Search space (inner CV)	Selected (mode)
1D-CNN kernels	{3, 5, 7} per block	[5, 3, 3]
1D-CNN filters	{[32, 64, 128], [64, 128, 256]}	[64, 128, 256]
Dropout	{0.2, 0.3, 0.5}	0.3
LR (Adam)	{1e−3, 5e−4}	1e−3
Batch size	{32, 64}	32
PCA k	{64, 128, 256}	128
SMOTE k	{3, 5, 7}	5
SVM C	{0.1, 1, 10}	10
SVM γ	{0.001, 0.01, 0.1}	0.01