PROMISES & PERILS OF EEG FOUNDATION MODELS: ON THE ROLE OF PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Foundation models for electroencephalography (EEG) typically rely on masked prediction, with the promise that pretraining on large unlabeled collections yields general-purpose representations. Yet published evaluations rarely isolate what pretraining itself contributes, leaving open whether reported gains come from representations learned from unlabeled EEG or from the encoder and adaptation protocol. We ask what pretraining contributes, whether its benefits are task-dependent, and whether the pretext task is aligned with all downstream tasks or only with some. Across 23 published encoder configurations and 34 datasets, evaluated with frozen weights, we find that pretraining gains are uneven: substantial for sleep staging and clinical EEG, but small or absent for motor imagery and event-related potentials, and almost entirely diminished once classifiers also receive handcrafted spectral and covariance features. To understand why, we pretrain encoders on transformed EEG that preserves only selected statistics of the original signals. Transformations retaining spectra and cross-spectra alone reproduce the benefits of original-EEG pretraining, yet the same transformation substantially impairs supervised motor-imagery and ERP decoding. Finally, adding a supervised term to reconstruction shows these representations to be learnable from the same data at negligible cost to the pretext objective; but removing supervision erodes most of the acquired decoding gains with little change in reconstruction loss. In the configurations studied, good reconstruction therefore does not assure the acquisition or retention of representations useful for motor-imagery and ERP decoding. These findings highlight objective–task alignment as a central challenge for EEG pretraining, motivating objectives better aligned with downstream tasks.

1 INTRODUCTION

Foundation models for electroencephalography (EEG) promise to turn a single investment in large-scale pretraining into reusable representations for many downstream tasks. By learning from thousands of hours of unlabeled recordings, one encoder could support diverse EEG applications, including sleep staging, clinical classification, and decoding of motor imagery (MI) and event-related potentials (ERPs), with each new task requiring only a lightweight readout trained on its own labels. A growing body of work pursues this goal through different model architectures, scales, and pretraining data (Jiang et al., 2024; Wang et al., 2025a; El Ouahidi et al., 2025; Wang et al., 2024; Döner et al., 2025; Zhou et al., 2025).

Whether this promise holds is usually assessed through downstream performance. Such performance, however, reflects an entire pipeline: the architecture, preprocessing, pretrained weights, classifier, and downstream training procedure (Xiong et al., 2026; Kastrati et al., 2025). Comparing these pipelines can guide model selection, but does not by itself establish what pretraining contributes. A model may perform well because its architecture already provides useful features, or because supervised adaptation learns what the downstream task requires. In the context of pretraining, many current approaches rely on predicting masked portions of the signal or their representations, on the premise that learning to recover missing information will capture structure useful beyond the pretraining task. To assess the value of pretraining itself, we must isolate the benefit of the pretrained weights and determine how that benefit varies across tasks. For EEG, this also requires an informative reference: spectral and spatial descriptors have long supported decoding (Ang et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2012; Barachant et al., 2012). The question is therefore not only whether pretraining helps, but whether it adds predictive value beyond these established representations.

This question matters because, by design, self-supervised learning optimizes a pretext task rather than a particular application task downstream. As such, its usefulness depends on alignment between the information needed to solve that pretext task as well as the transferable information needed for later downstream prediction (Lee et al., 2021). Recovering masked EEG signal may require some properties of a recording without requiring all the structure relevant to its labels. Consequently, success at masked prediction need not imply equally useful representations for every EEG task. Therefore, understanding this relationship requires more than evaluating published checkpoints: it requires interventions on pretraining to examine what the objective needs to learn and whether that matches what downstream tasks need.

In this paper, we investigate what EEG pretraining contributes and what determines the scope of its benefits. We perform a systematic and extensive empirical evaluation of published approaches in a series of controlled experiments, organized around three questions:

1. **What does pretraining add across EEG tasks?** We compare 23 published encoder configurations with randomly initialized copies of the same architectures across 34 datasets and tasks (Section 2). Frozen encoders and matched readouts (classification heads) isolate the contribution of pretrained weights under a common evaluation protocol. Repeating the comparison with classical spectral and spatial descriptors supplied to both readouts tests whether pretraining provides additional predictive value beyond these descriptors. We find that this is not usually the case: pretraining gains, substantial for sleep staging and clinical classification but small or absent for motor imagery and ERPs, nearly vanish once both readouts receive the descriptors.
2. **Does pretraining rely on the same signal properties as downstream decoding?** We intervene on pretraining itself and ask what it needs to solve its pretext task: we ran several pretrainings, on original EEG but also on synthetic transformations of the same recordings (Section 3). And pretraining on one transformation that preserves only power spectra and cross-spectra, destroying waveforms and temporal organization, ended up reconstructing masked original EEG nearly as well as pretraining on original recordings, and transferring equally well on every task family, across objectives and model sizes. Yet the same transformation removes information that a supervised EEGNet relies on: ERP decoding falls to chance and motor-imagery decoding loses most of its performance, while sleep, clinical and SSVEP decoding are largely unaffected.
3. **What role do the encoder, the data, and the objective play?** To distinguish these possible sources of limited transfer, we train the same encoder architecture on each task's own recordings, comparing self-supervised reconstruction alone with reconstruction augmented by label supervision (Section 4). We then remove the supervised objective and continue reconstruction on the same data. This design tests both whether the encoder can learn task-relevant information from the available recordings and whether reconstruction alone encourages its acquisition and retention. We find that label supervision enables the encoder to learn information relevant to motor imagery and ERPs at little cost in reconstruction error, whereas reconstruction alone yields weak decoding performance and fails to retain most of the gains acquired through supervision.

2 Isolating the contribution of pretraining across EEG tasks

Our first set of analyses focuses on assessing the contribution of pretraining to downstream decoding. To this end, we compare frozen pretrained encoders with randomly initialized variants of the same architectures under matched conditions. Specifically, we evaluate many published encoder configurations across an extensive set of datasets and tasks. Here we aim to assess the gain from pretrained weights across these diverse tasks and datasets. We then extend this analysis, and compare these performances to classical spectral and spatial descriptors supplied to both readouts. We aim with this experiment to measure the remaining performance gain, when such standard, statistical, descriptors are available.

Experimental setup. We evaluate 23 published encoder configurations spanning masked reconstruction, contrastive learning, self-distillation, autoregressive modeling, and latent prediction (Appendix A.2). The benchmark contains 34 datasets, covering motor imagery, ERP, SSVEP, sleep staging, clinical classification, emotion recognition, mental workload, and imagined speech (Ap-

pendix A.1). Each dataset contributes at most 6,000 windows, and subjects are disjoint across training, validation, and test splits. Because EEG foundation models are meant to support downstream learning with few labels, we also repeat the main analyses with 32 and 128 labeled windows per class (Appendix G).

For each encoder, we compare the pretrained representation with a randomly initialized copy of the same architecture. Specifically, let Z_P denote the tokens of a frozen pretrained encoder, and Z_0 those of its frozen, randomly initialized counterpart. We additionally compute classical descriptors, denoted by H_d, from the same normalized window. These include channel power spectra, covariance matrices in six frequency bands computed over the whole window, its halves, and its quarters, and coherency between channels. Because these descriptors have substantially higher dimensionality than the pooled encoder features, we use PCA to limit the dimensional imbalance between the two feature blocks, retaining at most d components, where d is the pooled encoder feature width. The projection is fitted without labels on the training and validation windows (Appendix A.3). Repeating the analysis without PCA preserves the main conclusions, indicating that they are not driven by dimensionality reduction (Appendix B.5, Table 13).

For each representation, we freeze the encoder and train a small head on top of it. The head applies attention pooling to collapse the encoder's tokens into a single vector Z, then feeds Z to a linear classifier. Pooling keeps the number of head parameters independent of the token count, which varies widely across architectures. When H_d is used, it is concatenated with Z before the classifier. The only trained parameters are the pooling query and the linear layer; alternative pooling and readout choices are examined in Appendices B.4 and B.6. Encoders remain frozen by design: once target labels update the encoder, its representation mixes what pretraining produced with what supervision induced, and a gain can no longer be attributed to the pretrained weights (Appendix J.1). Full fine-tuning instead measures the value of pretraining as an initialization; a targeted comparison on REVE-base shows that, at the full label budget, it raises the scores of pretrained and random encoders by similar amounts ($+0.05$ and $+0.06$ κ), leaving a pretrained–random gap comparable to the frozen one (Appendix J).

Within each pretrained–random pair, both heads see the same labeled windows and test subjects and follow the same hyperparameter search, with hyperparameters chosen separately for each head (Appendix A.4).

With $s(X)$ denoting the test score of a head trained on features X, we measure

$$\Delta_{\text{pre}} = s(Z_P) - s(Z_0), \tag{1}$$

$$\Delta_H = s([H_d, Z_P]) - s([H_d, Z_0]). \tag{2}$$

Δ_{pre} measures the gain obtained from the pretrained weights over the same randomly initialized architecture, while Δ_H measures how much of this gain remains when both heads are also provided with the classical descriptors. Comparing absolute scores across models would not isolate this contribution, since differences between architectures can already be present with random weights (Appendix B.1). The performance is measured as Cohen's κ (0 at chance), averaged over three head initializations. A family score averages all its model–dataset pairs, and the overall score weights the five main families equally (Appendix A.5).

2.1 Pretraining gains concentrate in sleep and clinical tasks

Averaged equally across the five main families, pretrained weights improve Cohen's κ by 0.070, although the gains vary substantially across tasks (Figure 1a). The largest gains are observed for sleep staging ($+0.162$) and clinical classification ($+0.105$), which are also the only two families whose 95% intervals exclude zero. Gains are smaller for SSVEP ($+0.055$) and motor imagery ($+0.029$), while almost no improvement is observed for ERP (-0.002).

The experiments also show that this pattern extends across architectures. For 14 of the 19 encoders evaluated in all five families, pretraining helps sleep and clinical tasks more than motor imagery and ERP (Appendix B.3). The difference between absolute performance and the contribution of pretraining is also evident across individual models. For example, on MI (Stieger et al., 2021), pretrained REVE outscores pretrained EEG-DINO (medium) by 0.18 κ, although both gain about 0.05 from pretraining and the difference between them is already present with random weights (Appendix A.5).

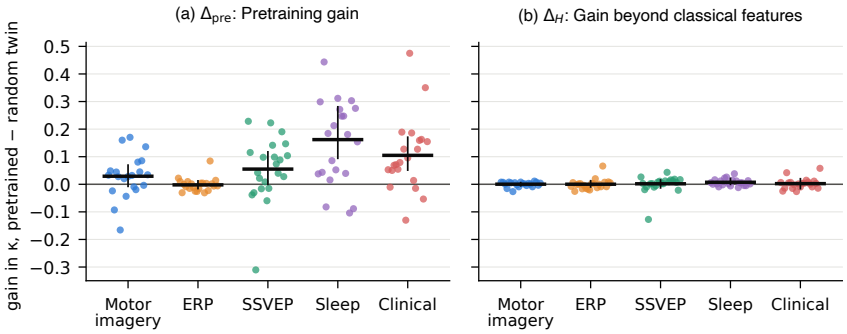


Figure 1: **Pretraining gains are uneven and nearly vanish once classical descriptors are available.** Gain in Cohen's κ of each pretrained encoder over its randomly initialized copy, with frozen encoders and matched attention-pooling heads. **(a)** Heads receive encoder features only (Δ_{pre}). **(b)** Both heads also receive the same classical descriptors (Δ_H). Each point averages one encoder configuration over the datasets of a family. Black bars show the mean over all model–dataset pairs of the family, with 95% intervals from a bootstrap over models and datasets (Appendix A.5). Results for the remaining seven tasks are given in Appendix B.1.

2.2 Pretraining adds little beyond classical descriptors

When the same classical descriptors are provided to both heads, the mean pretrained–random gap decreases from 0.070 to 0.002 κ (Figure 1b). The reduction is particularly pronounced for sleep staging and clinical classification, where the gains from pretraining are the largest when using encoder features alone, but fall below 0.01 κ once the descriptors are included. Power spectra alone account for most of this reduction in these two families (Appendix B.2). At the same time, adding the classical descriptors to pretrained features improves absolute performance by 0.12 to 0.21 κ in every main family except ERP (Appendix B.1).

These results indicate substantial overlap in predictive value between the pretrained representations and classical EEG descriptors under the evaluated readouts. They do not, however, establish that pretrained representations contain no additional information. Some model-specific exceptions remain. Specifically, REVE retains a mean ERP gain of 0.066, while LEAD retains a mean clinical gain of 0.058 (Appendix B.3). The same general reduction is observed with mean pooling, electrode-wise pooling, and without PCA. Although the residual gain depends on these choices, it remains below 0.04 κ in every main family (Appendices B.4–B.6). The same pattern is observed in the three additional families (Appendix B.1).

The near-zero gain from pretraining does not imply that classical descriptors exhaust the predictive information available for these tasks. EEGNet trained with supervision on the same tasks achieves substantial further improvements (Appendix H). This shows that additional predictive information can be learned, even when the evaluated pretrained representations provide little additional benefit beyond classical descriptors.

3 Signal properties that support pretraining and downstream decoding

We now ask whether the signal properties that suffice for effective pretraining are the same ones that downstream decoding relies on. To answer this, we transform EEG windows so that selected statistics are preserved while other signal structure is disrupted, and use each transformation in two complementary experiments. First, pretraining on transformed EEG and evaluating on original EEG tests whether the preserved statistics are sufficient to recover reconstruction and transfer performance. Second, training a supervised decoder on transformed EEG tests whether the task labels remain predictable from those statistics alone. If a transformation leaves pretraining performance intact but impairs supervised decoding, then the structure it removes is needed by the downstream task but not by pretraining, revealing a mismatch between the two.

Table 1: **Pretraining and supervised decoding respond differently to the same signal transformations.** Pretrained REVE-small with MAE pretrained on different transformed signal versions and evaluated on original EEG data; EEGNetv4 is trained from scratch and tested on the corresponding version. Scores are averaged equally across six datasets (mean ± SD over three seeds; one pretraining run for Mean + variance). The random baseline uses three encoder initialization.

Signal version	**Recon.** R_0^2	**Transfer** κ	**EEGNetv4** κ
Original EEG	0.657 ± 0.003	0.347 ± 0.005	0.523 ± 0.008
PSD + cross-spectra	0.648 ± 0.003	0.349 ± 0.005	0.374 ± 0.003
Covariance	0.570 ± 0.008	0.262 ± 0.006	0.118 ± 0.003
PSD	0.053 ± 0.001	0.283 ± 0.006	0.284 ± 0.005
Mean + variance	0.000	0.154	0.000 ± 0.002
Random initialization	—	0.146 ± 0.008	—

Experimental setup. We compare original EEG with four transformations of the same four-second windows. *PSD + cross-spectra* applies the same random Fourier phase rotation to all channels at each frequency, preserving whole-window power spectra and complex cross-spectra, including relative phases between channels, while altering waveforms and event timing. *PSD* uses independent rotations across channels, preserving their power spectra while disrupting cross-channel phase relationships. *Covariance* generates temporally white Gaussian signals matching the window's channel means and regularized spatial covariance in expectation. *Mean + variance* generates independent Gaussian noise matching each channel's mean and variance in expectation. These transformations test different combinations of signal properties; they do not form a single ordered sequence of degradations. Spectral preservation concerns the full window, not shorter intervals. Transformed training windows are generated afresh at every presentation (Appendix D).

To test what supports pretraining, we train REVE-small (El Ouahidi et al., 2025) from scratch on original TUEG recordings (Obeid & Picone, 2016) or their transformed versions, keeping the architecture, optimization settings, and training budget fixed. The main experiment uses masked reconstruction, hiding 75% of the one-second electrode tokens. Both the visible input and the reconstruction target come from the same original or transformed window. We then evaluate every model on *original EEG*: masked reconstruction is measured by R_0^2, one minus squared error divided by the energy of the masked samples, and transfer by a trained attention-pooling and linear classification head on the frozen encoder. Both fitting and evaluating this head use original EEG.

To test what supports downstream decoding, we apply the same transformations to each task's own windows and train a supervised decoder from scratch for each signal condition, retaining the original labels. Training, validation, and testing all use the corresponding signal version and the same subject-disjoint splits across conditions. We use EEGNetv4 (Lawhern et al., 2018) as a compact decoder to assess how much label-predictive information remains accessible after each transformation. The relevant comparison is the change from original to transformed EEG within each experiment: whether a transformation affects pretraining and supervised decoding in the same way.

Both experiments use the same downstream windows and subject partitions across six dataset–task combinations: BCI IV-2a (Tangermann et al., 2012) and PhysioNet MI (Schalk, 2009) for motor imagery, ERP Lee et al. (2019) and SSVEP (Lee et al., 2019), ISRUC (Khalighi et al., 2016) for sleep staging, and TUAB (Obeid & Picone, 2016) for abnormality detection. Classification is measured by Cohen's κ; aggregate scores weight the six combinations equally (Appendix C).

Additional controls test whether the pretraining comparison extends to JEPA-style latent prediction (Assran et al., 2023), alone or combined with reconstruction, alternative pooling strategies, a larger encoder, a fivefold training budget, and LaBraM with its native discrete prediction objective (Appendices E.2, E.3, and I).

3.1 Preserving spectra and cross-spectra recovers pretraining performance

The results show that REVE-small pretrained on PSD + cross-spectra signals achieves reconstruction and downstream scores close to those obtained after pretraining on original EEG data: 0.648 vs. 0.657 R_0^2 and 0.349 vs. 0.347 κ (Table 1).Both pretrained encoders substantially outperform

the randomly initialized encoder on average (0.146 κ). The similarity between the two pretraining conditions is also evident across task families (Figure 2) and individual dataset–task combinations (Appendix E.1). In both conditions, gains over random initialization are much larger on sleep staging, abnormality detection and SSVEP than on motor imagery and ERPs.

In contrast, the other transformations yield lower mean downstream performance. Pretraining on features that capture only channel means and variances gives a reconstruction score close to zero on original EEG data and downstream performance close to that of the randomly initialized encoder. PSD and Covariance both improve downstream performance relative to random initialization, but the gains are smaller than those obtained with real-EEG data pretraining. The comparison also shows that reconstruction and downstream performance do not follow the same ordering. Covariance reconstructs much better than PSD (0.570 vs. 0.053 R_0^2), but leads to lower downstream performance (0.262 vs. 0.283 κ). These findings indicate that better reconstruction does not necessarily correspond to better downstream performance.

3.2 The result extends across objectives and training configurations

Our results demonstrate that models pretrained on original EEG or PSD + cross-spectra signals achieve similar mean downstream performance under all three objectives (Table 2). The similarity observed with masked reconstruction therefore also extends to the evaluated latent-prediction JEPA objectives.The downstream scores remain close across these comparisons, regardless of pooling strategies, encoder sizes and training budgets. At the larger budget, attention-pooling transfer reaches 0.371 for original EEG data and 0.368 for PSD + cross-spectra. The baselines trained with PSD and Covariance yield lower mean transfer throughout the objective and encoder-size comparisons (Appendices E.2–E.3).

To validate our pretraining setup, we compared our own pretraining of REVE-base against the official pretrained weights under the same downstream evaluation. Despite differences in pretraining data and compute, our model matches the published one ($\kappa = 0.368$ for ours vs. 0.347 for the official weights; Appendix E.2, Table 24), confirming that our setup is a reasonable basis for studying the effects of pretraining. These findings show that the original EEG waveforms are not required in the pretraining data to obtain similar performance under the evaluated configurations. However, they do not establish that the encoders rely exclusively on the retained statistics.

3.3 The same transformation impairs motor-imagery and ERP decoding

The supervised experiment shows that the effect of the same transformation depends strongly on the downstream task. PSD + cross-spectra leaves EEGNetv4 performance largely unchanged on sleep staging, abnormality detection and SSVEP decoding (Figure 2). In contrast, ERP Lee et al. (2019) decoding falls to chance, from 0.341 to 0 κ, and motor-imagery decoding drops from 0.440 to 0.168 on BCI IV-2a and from 0.395 to 0.075 on PhysioNet MI (Appendix E.1). For ERPs, similar real–synthetic transfer thus reflects a shared limitation: neither pretraining condition improves over random initialization, although the real signal supports supervised decoding. The agreement between these conditions therefore does not indicate successful transfer to ERP decoding. Together, these results support the alignment hypothesis in Section 1: the transformation preserves the ob-

Table 2: **Real and PSD + cross-spectra pretraining yield similar transfer across objectives and encoder sizes.** Frozen transfer to six real-EEG corpora, weighted equally, with attention pooling (κ). REVE-small entries are mean $\pm$ SD over three pretraining seeds; REVE-base uses one run per signal. Δ is synthetic minus real, computed from unrounded scores within corresponding pretraining-run indices before aggregation.

Encoder	Objective	Original EEG	PSD + cross-spectra	Δ
REVE-small	MAE	0.347 ± 0.005	0.349 ± 0.005	$+0.002 \pm 0.008$
	JEPA	0.354 ± 0.002	0.349 ± 0.003	-0.005 ± 0.005
	JEPA + recon.	0.352 ± 0.004	0.349 ± 0.003	-0.003 ± 0.004
REVE-base	MAE	0.368	0.364	-0.003

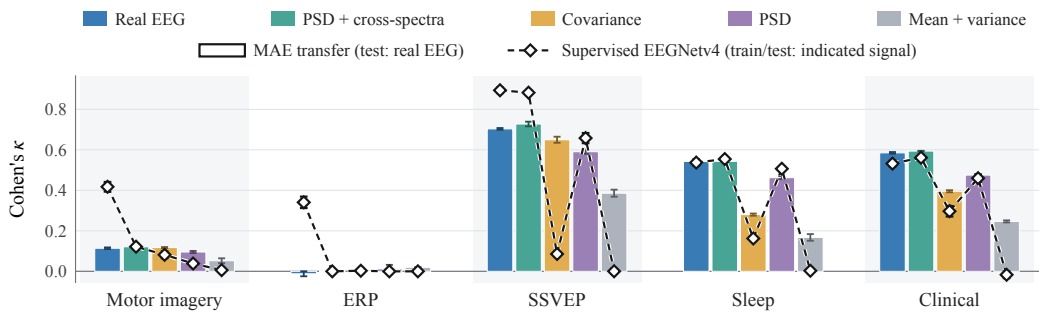


Figure 2: **Preserving pretraining transfer does not preserve downstream decoding.** REVE-small MAE encoders pretrained on each signal version are frozen and evaluated on original EEG (bars); EEGNetv4 is trained from scratch and tested on the corresponding signal version (diamonds). Signal versions are compared within each model. Values are mean ± SD across three seeds (except for Mean + variance pretraining, on one seed)

served pretraining benefit while disrupting structure useful for decoding the tasks that pretraining helps least. Note that these findings do not establish complete information loss.

4 Reconstruction does not retain most of the supervised gain

The previous analyses leave open whether the limited transfer to motor-imagery and ERP tasks is due to the clinical pretraining data, the encoder of choice, or the training objective. To separate these factors, we pretrain the same encoder architecture directly on the recordings of each downstream task, adding a supervised loss on the task labels to its usual reconstruction objective. This first stage shows what the labels contribute when data and architecture are held fixed. We then drop the label loss and continue training on the same windows with reconstruction alone, as in label-free pretraining, to test whether the encoder retains the decoding gains that supervision had produced. This lets us examine both the acquisition and the retention of representations useful for decoding.

Experimental setup. We train REVE-small masked autoencoders on four tasks, treating each corpus separately: motor imagery (MI) (Stieger et al., 2021), event-related potentials (ERP) (Lee et al., 2019), sleep staging (Khalighi et al., 2016), and abnormality detection (Obeid & Picone, 2016), treating each corpus separately. For each data set and seed, we initialize two models with the same random weights and train them on the same windows. One minimizes the masked-patch L1 loss L_{recon}; the other minimizes $L_{\text{recon}} + \lambda L_{\text{label}}$, with $\lambda = 0.02$. The label loss is the cross-entropy of a linear classifier applied to the mean-pooled encoder features of the complete, unmasked window.

After 4.096 million window samples (the *split*), we copy the jointly trained model, including its optimizer state, into two branches. One branch keeps both losses; the other drops the label loss and continues with reconstruction alone. Both branches then train on another 8.192 million samples, seen in the same order and with the same masks. The reconstruction-only reference also keeps training, up to the same total of 12.288 million samples. Each dataset has about 24,000 training windows, so windows are seen many times. All runs use the same number of updates and samples; supervision only adds computation, because the label loss needs an extra pass on the complete, unmasked window (Appendices F.1–F.2).

Encoder training, classification-head development, and testing use disjoint subjects. At every checkpoint, we freeze the encoder and fit a fresh attention-pooling readout, independent of the classifier used during encoder training. The head takes either the encoder features Z alone or their concatenation with H_d. The latter forms $[H_{512}, Z]$, which this contains the classical descriptors of Section 2 reduced to $d = 512$ principal components. For absolute performance, we pool test windows and average the resulting scores first over three head initializations and then over three encoder seeds. The gains from supervision when descriptors are available and passed to both heads are calculated by averaging within-subject differences against the reconstruction-only reference at the same training budget, with identical descriptors passed to both heads (Appendices F.3–F.4).

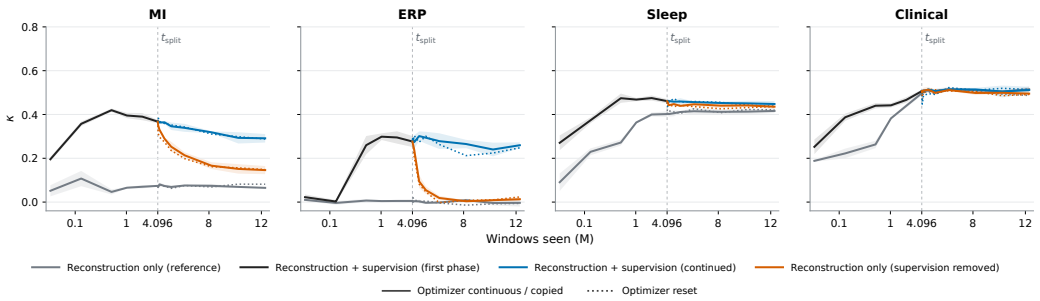


Figure 3: **Continuing with reconstruction alone loses most of the decoding gain on motor imagery and ERPs.** Test κ of an attention-pooling readout refitted at every checkpoint on frozen encoder features Z, on subjects unseen during encoder training. Until the split (4.096M samples), encoders are trained with reconstruction alone (grey) or with reconstruction plus label supervision (black). The jointly trained encoder is then copied and continues with (blue) or without (orange) the label loss, on the same windows, in the same order, with the same masks. Dotted lines rerun both continuations and the reference with a fresh optimizer. Curves show mean $\pm$ SD over three seeds, each averaged over three head initializations. x-axis is logarithmic before the split and linear after.

4.1 Labels reveal predictive gains that reconstruction does not obtain

The results on task-specific recordings show the same differences between tasks observed in the preceding experiments. Reconstruction alone produces representations useful for sleep staging and abnormality detection, but leads to much weaker decoding performance on motor imagery and ERPs (Figure 3, grey). At the split, leveraging label supervision increases pooled test κ from 0.074 to 0.367 on MI (Stieger et al., 2021) and from 0.005 to 0.277 on ERP. These gains remain when both heads receive H_{512}: paired subject-level gains are $+0.203$ and $+0.258$, respectively ($+0.159$ and $+0.223$ with the full descriptors; Appendix F.6). The same architecture and recordings thus support predictive gains that reconstruction-only training does not obtain. A mismatch with the clinical pretraining corpus cannot, by itself, explain this result.

We find that the benefit of label supervision is much smaller on ISRUC and TUAB, particularly when the heads also receive the classical descriptors. As these tasks provide little additional gain to assess, we focus further analyses on MI and ERP (Appendix F.5).

4.2 The decoding gain has little cost in reconstruction error

We next examine whether the decoding gains obtained with supervision are accompanied by an increase in reconstruction loss. We compare masked-patch L1 losses at the split, using identical masks on a fixed evaluation set of subjects excluded from encoder training. On MI and ERP, the jointly trained models' reconstruction losses exceed those of the reconstruction-only reference of up to 0.74% across the three encoder seeds. The upper bounds of all paired 95% batch-bootstrap intervals are below 1% (Table 34; Appendix F.4). Substantial decoding gains can therefore coexist with almost unchanged mean reconstruction loss. Although this does not completely rule out competition between the two objectives during training, it shows that the observed decoding gains require little increase in the measured reconstruction loss.

4.3 Reconstruction-only continuation loses most of the acquired gain

The decoding gains decline when training resumes with reconstruction alone. Without labels, pooled test κ falls to 0.146 on MI and 0.013 on ERP, compared with 0.291 and 0.260 under continued supervised training (Figure 3). Across encoder seeds, the reconstruction-only continuation retains 22–32% of the initial advantage over the reference on MI and 2–14% on ERP. On MI, we find performance also declines under continued supervision, although a larger proportion of the gain is retained.

Across both tasks and seeds, the performance of the variant trained without supervision degrades compared to its counterpart, with paired 95% subject-bootstrap intervals (Table 31). Moreover, the

supervision-induced advantage with descriptors supplied to both heads also shrinks: final performance gains drop to $+0.028$ on MI and $+0.002$ on ERP without labels, compared to $+0.150$ and $+0.205$ with labels (Table 32). Since every checkpoint receives a newly fitted head, this decline is not simply a mismatch with an outdated classifier. It measures a loss of predictive value accessible to the classification head, not necessarily the disappearance of all task information.

At the endpoint, both continuations still satisfy the reconstruction criterion on MI and ERP: their loss excesses over the reference have upper interval bounds below 1%. Losing most of the decoding gain is thus not necessary to remain within this tolerance. The decline also occurs with a fresh optimizer and a fivefold larger label weight (one encoder seed per control; Appendix F.6). Restricting the labels available to the fresh readouts preserves the retention gap without changing encoder training (Appendix G.5). This mirrors the dissociation reported by Warner et al. (2026), whose downstream accuracy fell during training, on some datasets to near chance, while its reconstruction kept improving.

5 Related Work

Several benchmarks compare EEG foundation models under shared protocols (Xiong et al., 2026; Kastrati et al., 2025; Banville et al., 2026; Kontras et al., 2026). While they are extensive and useful for model selection, they focus on complete pipelines, with a very limited analysis on the contribution of pretrained weights. The closest work to ours is the one of Yang et al. (2026), who find that linear probing performs poorly and that, when data are scarce, fine-tuned models often fail to significantly outperform compact networks or classical non-neural decoders. We instead isolate the contribution of the pretrained weights. We compare each checkpoint with a randomly initialized copy of the same architecture (Saxe et al., 2011; Wieting & Kiela, 2019), and we give the same classical descriptors to the readout in both cases. This extends the from-scratch or random-initialization baselines used in some benchmarks (Xiong et al., 2026; Kontras et al., 2026). These results implicate objective–task alignment as a limiting factor for MI and ERP decoding under the tested training setups.

The usefulness of pretraining depends in part on the alignment between the pretext and downstream tasks (Lee et al., 2021). Reconstruction objectives may not meet this condition. In vision, they favor high-variance components that can be irrelevant for perception (Balestriero & LeCun, 2024). In EEG, masked-reconstruction gradients are only weakly aligned with classification gradients (Xiong et al., 2026), and both reconstruction losses and embeddings are dominated by low-frequency, aperiodic activity (Yu et al., 2026; Kommineni et al., 2026). These studies diagnose the problem with the objective; we go further and intervene on pretraining, either by changing the signal the model sees or by adding label supervision. Prior work has used synthetic signals as EEG pretraining data (Grieger et al., 2025; Gnassounou et al., 2025); we instead use transformed recordings as a probe, in the spirit of surrogate-data testing (Theiler et al., 1992; Prichard & Theiler, 1994). Lastly, Warner et al. (2026) observe that reconstruction quality and downstream performance can diverge during training. Our continuation experiments test this with matched data exposure, reconstruction masks, and optimization steps, which lets us separate whether downstream gains are acquired or retained.

6 Conclusion

In this work we perform an extensive empirical analysis of the benefits of EEG foundation models and the contribution of pretraining across datasets and tasks. Our findings show that pretraining on transformed EEG that preserves spectra and cross-spectra can almost reproduce the performance obtained by pretraining on raw EEG data, while the same transformation impairs supervised motor imagery and ERP decoding. We find that, supervised objective functions enable the same encoders to learn these tasks, but continued reconstruction alone erodes most of the gains. These results identify the pretraining objective as a key bottleneck: It does not ensure that an encoder learns or retains representations useful across EEG tasks. As such, our work focuses on diagnosing these limitations – identifying and validating self-supervised objectives that recover and retain these predictive gains is left to future work.

AI USE STATEMENT

The scientific ideas, methods, experimental design, analyses, and conclusions originate from the authors. Generative AI tools were used solely for assistance with translation and wording, preparation of some plots from the authors' results, and organization of existing code for anonymous release. They were not used to develop or implement the experimental methods or to conduct experiments.

ETHICS STATEMENT

This study uses existing EEG datasets made available to the research community and cited in the paper. No new participants were recruited and no new recordings were collected for this work. Our experiments investigate representation learning using previously collected data. The reported results do not establish the clinical safety or effectiveness of the evaluated models and should not be interpreted as validating them for clinical deployment.

REPRODUCIBILITY STATEMENT

An anonymized version of our code accompanies this submission to support reproduction of the reported experiments. The experimental protocols are described in the paper and its appendices, including data preprocessing, subject-disjoint splits, encoder configurations, descriptor construction, training and evaluation procedures, surrogate generation, and continuation experiments.

REFERENCES

Diego Alvarez-Estevez and Roselyne M. Rijsman. Inter-database validation of a deep learning approach for automatic sleep scoring. *PLOS ONE*, 16(8):e0256111, 2021. doi: 10.1371/journal.pone.0256111. URL https://doi.org/10.1371/journal.pone.0256111.

Kai Keng Ang, Zheng Yang Chin, Chuanchu Wang, Cuntai Guan, and Haihong Zhang. Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. *Frontiers in Neuroscience*, 6:39, 2012. doi: 10.3389/fnins.2012.00039. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2012.00039/full.

Md Fahim Anjum, Arturo I. Espinoza, Rachel C. Cole, Arun Singh, Patrick May, Ergun Y. Uc, Soura Dasgupta, and Nandakumar S. Narayanan. Resting-state EEG measures cognitive impairment in Parkinson's disease. *npj Parkinson's Disease*, 10(1):6, 2024. doi: 10.1038/s41531-023-00602-0. URL https://doi.org/10.1038/s41531-023-00602-0.

P Aricò, F Aloise, F Schettini, S Salinari, D Mattia, and F Cincotti. Influence of P300 latency jitter on event related potential-based brain–computer interface performance. *Journal of Neural Engineering*, 11(3):035008, 2014. doi: 10.1088/1741-2560/11/3/035008. URL https://doi.org/10.1088/1741-2560/11/3/035008.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-Supervised Learning From Images With a Joint-Embedding Predictive Architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Assran_Self-Supervised_Learning_From_Images_With_a_Joint-Embedding_Predictive_Architecture_CVPR_2023_paper.html.

Randall Balestriero and Yann LeCun. How Learning by Reconstruction Produces Uninformative Features For Perception. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 2566–2585. PMLR, 2024. URL https://proceedings.mlr.press/v235/balestriero24b.html.

Hubert Banville, Stéphane d'Ascoli, Simon Dahan, Jérémy Rapin, Marlène Careil, Yohann Benchetrit, Jarod Lévy, Saarang Panchavati, Antoine Ratouchniak, Mingfang (Lucy) Zhang, Elisa Cascardi, Katelyn Begany, Teon Brooks, and Jean-Rémi King. NeuralBench: A Unifying Framework to Benchmark NeuroAI Models. arXiv preprint arXiv:2605.08495, 2026. URL https://arxiv.org/abs/2605.08495.

Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Multiclass brain-computer interface classification by Riemannian geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928, 2012. doi: 10.1109/TBME.2011.2172210. URL https://hal.science/hal-00681328.

Konstantinos Barmpas, Na Lee, Alexandros Koliousis, Yannis Panagakis, Dimitrios A. Adamos, Nikolaos Laskaris, and Stefanos Zafeiriou. NeuroRVQ: Multi-scale EEG tokenization for generative large brainwave models. arXiv preprint arXiv:2510.13068v1, 2025. URL https://arxiv.org/abs/2510.13068v1. Version 1.

Jingjing Chen, Xiaobin Wang, Chen Huang, Xin Hu, Xinke Shen, and Dan Zhang. A large finer-grained affective computing EEG dataset. *Scientific Data*, 10(1):740, 2023. doi: 10.1038/s41597-023-02650-w. URL https://doi.org/10.1038/s41597-023-02650-w.

Hohyun Cho, Minkyu Ahn, Sangtae Ahn, Moonyoung Kwon, and Sung Chan Jun. EEG datasets for motor imagery brain–computer interface. *GigaScience*, 6(7):gix034, 2017. doi: 10.1093/gigascience/gix034. URL https://doi.org/10.1093/gigascience/gix034.

Wenhui Cui, Woojae Jeong, Philipp Thölke, Takfarinas Medani, Karim Jerbi, Anand A. Joshi, and Richard M. Leahy. Neuro-GPT: Towards A Foundation Model For EEG. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, 2024. doi: 10.1109/isbi56570.2024.10635453. URL https://doi.org/10.1109/isbi56570.2024.10635453.

Berkay Döner, Thorir Mar Ingolfsson, Luca Benini, and Yawei Li. LUNA: Efficient and topology-agnostic foundation model for EEG signal analysis. arXiv preprint arXiv:2510.22257, 2025. URL https://arxiv.org/abs/2510.22257.

Pauline Dreyer, Aline Roc, Léa Pillette, Sébastien Rimbert, and Fabien Lotte. A large EEG database with users' profile information for motor imagery brain-computer interface research. *Scientific Data*, 10(1):580, 2023. doi: 10.1038/s41597-023-02445-z. URL https://doi.org/10.1038/s41597-023-02445-z.

Yassine El Ouahidi, Jonathan Lys, Philipp Thölke, Nicolas Farrugia, Bastien Pasdeloup, Vincent Gripon, Karim Jerbi, and Giulia Lioi. REVE: A foundation model for EEG – adapting to any setup with large-scale pretraining on 25,000 subjects. arXiv preprint arXiv:2510.21585, 2025. URL https://arxiv.org/abs/2510.21585.

Théo Gnassounou, Yessin Moakher, Shifeng Xie, Vasilii Feofanov, and Ievgen Redko. Leveraging Generic Time Series Foundation Models for EEG Classification. arXiv preprint arXiv:2510.27522, 2025. URL https://arxiv.org/abs/2510.27522.

Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000. doi: 10.1161/01.CIR.101.23.e215. URL https://doi.org/10.1161/01.CIR.101.23.e215.

Niklas Grieger, Siamak Mehrkanoon, and Stephan Bialonski. Data-Efficient Sleep Staging with Synthetic Time Series Pretraining. *Algorithms*, 18(9):580, 2025. doi: 10.3390/a18090580. URL https://doi.org/10.3390/a18090580.

Pierre Guetschel, Thomas Moreau, and Michael Tangermann. S-JEPA: Towards seamless cross-dataset transfer through dynamic spatial attention. arXiv preprint arXiv:2403.11772, 2024. URL https://arxiv.org/abs/2403.11772.

Christoph Guger, Shahab Daban, Eric Sellers, Clemens Holzner, Gunther Krausz, Roberta Carabalona, Furio Gramatica, and Guenter Edlinger. How many people are able to control a P300-based brain–computer interface (BCI)? *Neuroscience Letters*, 462(1):94–98, 2009. doi: 10.1016/j.neulet.2009.06.045. URL https://doi.org/10.1016/j.neulet.2009.06.045.

John Guttag. CHB-MIT Scalp EEG Database. PhysioNet dataset, 2010. URL https://physionet.org/content/chbmit/1.0.0/. Version 1.0.0.

Ji-Hoon Jeong, Jeong-Hyun Cho, Young-Eun Lee, Seo-Hyun Lee, Gi-Hwan Shin, Young-Seok Kweon, José del R. Millán, Klaus-Robert Müller, and Seong-Whan Lee. 2020 International brain–computer interface competition: A review. *Frontiers in Human Neuroscience*, 16:898300, 2022. doi: 10.3389/fnhum.2022.898300. URL https://doi.org/10.3389/fnhum.2022.898300.

Wei-Bang Jiang, Li-Ming Zhao, and Bao-Liang Lu. Large brain model for learning generic representations with tremendous EEG data in BCI. arXiv preprint arXiv:2405.18765, 2024. URL https://arxiv.org/abs/2405.18765.

Wei-Bang Jiang, Yansen Wang, Bao-Liang Lu, and Dongsheng Li. NeuroLM: A Universal Multi-task Foundation Model for Bridging the Gap between Language and EEG Signals. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/8b4add8b0aa8749d80a34ca5d941c355-Abstract-Conference.html.

Emmanuel K. Kalunga, Sylvain Chevallier, Quentin Barthélemy, Karim Djouani, Eric Monacelli, and Yskandar Hamam. Online SSVEP-based BCI using riemannian geometry. *Neurocomputing*, 191:55–68, 2016. doi: 10.1016/j.neucom.2016.01.007. URL https://doi.org/10.1016/j.neucom.2016.01.007.

Emily S. Kappenman, Jaclyn L. Farrens, Wendy Zhang, Andrew X. Stewart, and Steven J. Luck. ERP CORE: An open resource for human event-related potential research. *NeuroImage*, 225:117465, 2021. doi: 10.1016/j.neuroimage.2020.117465. URL https://doi.org/10.1016/j.neuroimage.2020.117465.

Ard Kastrati, Josua Bürki, Jonas Lauer, Cheng Xuan, Raffaele Iaquinto, and Roger Wattenhofer. EEG-Bench: A benchmark for EEG foundation models in clinical applications. arXiv preprint arXiv:2512.08959, 2025. URL https://arxiv.org/abs/2512.08959.

Stamos Katsigiannis and Naeem Ramzan. DREAMER: A database for emotion recognition through EEG and ECG signals from wireless low-cost off-the-shelf devices. *IEEE Journal of Biomedical and Health Informatics*, 22(1):98–107, 2018. doi: 10.1109/JBHI.2017.2688239. URL https://doi.org/10.1109/JBHI.2017.2688239.

B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Oberye. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. *IEEE Transactions on Biomedical Engineering*, 47(9):1185–1194, 2000. doi: 10.1109/10.867928. URL https://doi.org/10.1109/10.867928.

Sirvan Khalighi, Teresa Sousa, José Moutinho Santos, and Urbano Nunes. ISRUC-Sleep: A comprehensive public dataset for sleep researchers. *Computer Methods and Programs in Biomedicine*, 124:180–192, 2016. doi: 10.1016/j.cmpb.2015.10.013. URL https://doi.org/10.1016/j.cmpb.2015.10.013.

Aditya Kommineni, Emily Zhou, Kleanthis Avramidis, Simon Bock Segaard, Jeppe Roden Münster, Andreas Peter Juhl Hansen, Takfarinas Medani, Tiantian Feng, Richard Leahy, and Shrikanth Narayanan. Aperiodic and Low-Frequency Spectral Bias in Reconstruction based EEG Foundation Models. arXiv preprint arXiv:2605.26434, 2026. URL https://arxiv.org/abs/2605.26434.

Konstantinos Kontras, Trui Osselaer, Stylianos G. Mouslech, Angeliki-Ilektra Karaiskou, Guido Gagliardi, Thomas Strypsteen, Mohammad Hossein Badiei, Anku Rani, Maarten Vanmarcke, Miguel Bhagubai, Chanakya Ekbote, Jaedong Hwang, Christos Chatzichristos, Paul Pu Liang, and Maarten De Vos. NeuroAtlas: Benchmarking Foundation Models for Clinical EEG and Brain-Computer Interfaces. arXiv preprint arXiv:2605.14698, 2026. URL https://arxiv.org/abs/2605.14698.

Demetres Kostas, Stephane Aroca-Ouellette, and Frank Rudzicz. BENDR: Using transformers and a contrastive self-supervised learning task to learn from massive amounts of EEG data. arXiv preprint arXiv:2101.12037, 2021. URL https://arxiv.org/abs/2101.12037.

Vernon J. Lawhern, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou P. Hung, and Brent J. Lance. EEGNet: A compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, 15(5):056013, 2018. doi: 10.1088/1741-2552/aace8c. URL https://doi.org/10.1088/1741-2552/aace8c.

Jason D. Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting what you already know helps: Provable self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 309–323, 2021. URL https://proceedings.nips.cc/paper_files/paper/2021/hash/02e656adee09f8394b402d9958389b7d-Abstract.html.

Min-Ho Lee, O-Yeon Kwon, Yong-Jeong Kim, Hong-Kyung Kim, Young-Eun Lee, John Williamson, Siamac Fazli, and Seong-Whan Lee. EEG dataset and OpenBMI toolbox for three BCI paradigms: an investigation into BCI illiteracy. *GigaScience*, 8(5):giz002, 2019. doi: 10.1093/gigascience/giz002. URL https://doi.org/10.1093/gigascience/giz002.

Bingchuan Liu, Xiaoshan Huang, Yijun Wang, Xiaogang Chen, and Xiaorong Gao. BETA: A large benchmark database toward SSVEP-BCI application. *Frontiers in Neuroscience*, 14:627, 2020. doi: 10.3389/fnins.2020.00627. URL https://doi.org/10.3389/fnins.2020.00627.

Wei Liu, Jie-Lin Qiu, Wei-Long Zheng, and Bao-Liang Lu. Comparing Recognition Performance and Robustness of Multimodal Deep Learning Models for Multimodal Emotion Recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 14(2):715–729, 2022. doi: 10.1109/tcds.2021.3071170. URL https://doi.org/10.1109/tcds.2021.3071170.

Jingying Ma, Feng Wu, Qika Lin, Yucheng Xing, Chenyu Liu, Ziyu Jia, and Mengling Feng. CodeBrain: Bridging decoupled tokenizer and multi-scale architecture for EEG foundation model. arXiv preprint arXiv:2506.09110, 2025. URL https://arxiv.org/abs/2506.09110.

Andreas Miltiadous, Katerina D. Tzimourta, Theodora Afrantou, Panagiotis Ioannidis, Nikolaos Grigoriadis, Dimitrios G. Tsalikakis, Pantelis Angelidis, Markos G. Tsipouras, Euripidis Glavas, Nikolaos Giannakeas, and Alexandros T. Tzallas. A dataset of scalp EEG recordings of alzheimer's disease, frontotemporal dementia and healthy subjects from routine EEG. *Data*, 8(6):95, 2023. doi: 10.3390/data8060095. URL https://doi.org/10.3390/data8060095.

Wajid Mumtaz. MDD Patients and Healthy Controls EEG Data (New). Figshare dataset, 2016. URL https://doi.org/10.6084/m9.figshare.4244171.v2. Version 2.

Masaki Nakanishi, Yijun Wang, Yu-Te Wang, and Tzyy-Ping Jung. A comparison study of canonical correlation analysis based methods for detecting steady-state visual evoked potentials. *PLOS ONE*, 10(10):e0140703, 2015. doi: 10.1371/journal.pone.0140703. URL https://doi.org/10.1371/journal.pone.0140703.

Spiros Nikolopoulos. MAMEM EEG SSVEP Dataset III (14 channels, 11 subjects, 5 frequencies presented simultaneously). Figshare dataset, 2021. URL https://doi.org/10.6084/m9.figshare.3413851.v3. Version 3.

Iyad Obeid and Joseph Picone. The Temple University Hospital EEG Data Corpus. *Frontiers in Neuroscience*, 10:196, 2016. doi: 10.3389/fnins.2016.00196. URL https://doi.org/10.3389/fnins.2016.00196.

Dean Prichard and James Theiler. Generating surrogate data for time series with several simultaneously measured variables. *Physical Review Letters*, 73(7):951–954, 1994. doi: 10.1103/PhysRevLett.73.951. URL https://doi.org/10.1103/PhysRevLett.73.951.

Andrew M. Saxe, Pang Wei Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Y. Ng. On Random Weights and Unsupervised Feature Learning. In *Proceedings of the 28th International Conference on Machine Learning*, 2011. URL https://ai.stanford.edu/~ang/papers/icml11-RandomWeights.pdf.

Gerwin Schalk. EEG motor movement/imagery dataset. *PhysioNet*, September 2009. doi: 10.13026/C28G6P. URL https://doi.org/10.13026/C28G6P. Version 1.0.0.

Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, 38(11):5391–5420, 2017. doi: 10.1002/hbm.23730. URL https://doi.org/10.1002/hbm.23730.

Jaeyoung Shin, Alexander von Lühmann, Benjamin Blankertz, Do-Won Kim, Jichai Jeong, Han-Jeong Hwang, and Klaus-Robert Müller. Open access dataset for EEG+NIRS single-trial classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(10):1735–1745, 2017. doi: 10.1109/TNSRE.2016.2628057. URL https://doi.org/10.1109/TNSRE.2016.2628057.

Zitao Shuai, Zongzhe Xu, David Yang, Wei Wang, and Yuzhe Yang. OSF: On pre-training and scaling of sleep foundation models. arXiv preprint arXiv:2603.00190, 2026. URL https://arxiv.org/abs/2603.00190.

James R. Stieger, Stephen A. Engel, and Bin He. Continuous sensorimotor rhythm based brain computer interface learning in a large population. *Scientific Data*, 8:98, 2021. doi: 10.1038/s41597-021-00883-1. URL https://doi.org/10.1038/s41597-021-00883-1.

Michael Tangermann, Klaus-Robert Müller, Ad Aertsen, Niels Birbaumer, Christoph Braun, Clemens Brunner, Robert Leeb, Carsten Mehring, Kai J. Miller, Gernot R. Müller-Putz, Guido Nolte, Gert Pfurtscheller, Hubert Preissl, Gerwin Schalk, Alois Schlögl, Carmen Vidaurre, Stephan Waldert, and Benjamin Blankertz. Review of the BCI competition IV. *Frontiers in Neuroscience*, 6:55, 2012. doi: 10.3389/fnins.2012.00055. URL https://doi.org/10.3389/fnins.2012.00055.

Anna Tegon, Thorir Mar Ingolfsson, Xiaying Wang, Luca Benini, and Yawei Li. FEMBA: Efficient and scalable EEG analysis with a bidirectional mamba foundation model. arXiv preprint arXiv:2502.06438, 2025. URL https://arxiv.org/abs/2502.06438.

James Theiler, Stephen Eubank, André Longtin, Bryan Galdrikian, and J. Doyne Farmer. Testing for nonlinearity in time series: The method of surrogate data. *Physica D: Nonlinear Phenomena*, 58(1–4):77–94, 1992. doi: 10.1016/0167-2789(92)90102-S. URL https://doi.org/10.1016/0167-2789(92)90102-S.

Guangyu Wang, Wenchao Liu, Yuhong He, Cong Xu, Lin Ma, and Haifeng Li. EEGPT: Pretrained transformer for universal and reliable representation of EEG signals. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/4540d267eeec4e5dbd9dae9448f0b739-Abstract-Conference.html.

Jiquan Wang, Sha Zhao, Zhiling Luo, Yangxuan Zhou, Haiteng Jiang, Shijian Li, Tao Li, and Gang Pan. CBraMod: A criss-cross brain foundation model for EEG decoding. In *International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2412.07236.

Jiquan Wang, Sha Zhao, Zhiling Luo, Yangxuan Zhou, Shijian Li, and Gang Pan. EEGMamba: An EEG foundation model with mamba. *Neural Networks*, 192:107816, 2025b. doi: 10.1016/j.neunet.2025.107816. URL https://doi.org/10.1016/j.neunet.2025.107816.

Xujia Wang, Xuhui Liu, Xi Liu, Qian Si, Zhaoliang Xu, Yang Li, and Xiantong Zhen. EEG-DINO: Learning EEG foundation models via hierarchical self-distillation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2025*, volume 15960 of *Lecture Notes in Computer Science*, pp. 196–205. Springer Nature Switzerland, 2025c. doi: 10.1007/978-3-032-04927-8_19. URL https://doi.org/10.1007/978-3-032-04927-8_19.

Yihe Wang, Nan Huang, Nadia Mammone, Marco Cecchi, and Xiang Zhang. LEAD: An EEG foundation model for alzheimer's disease detection. arXiv preprint arXiv:2502.01678, 2025d. URL https://arxiv.org/abs/2502.01678.

Christopher Warner, Jonas Mago, JR Huml, and Beren Millidge. ZUNA1.1: A more flexible EEG foundation model for Denoising and Super-resolution. arXiv preprint arXiv:2607.27308, 2026. URL https://arxiv.org/abs/2607.27308.

Peter D. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967. doi: 10.1109/TAU.1967.1161901. URL https://doi.org/10.1109/TAU.1967.1161901.

John Wieting and Douwe Kiela. No Training Required: Exploring Random Encoders for Sentence Classification. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BkgPajAcY7.

Qinfan Xiao, Ziyun Cui, Chi Zhang, Siqi Chen, Wen Wu, Andrew Thwaites, Alexandra Woolgar, Bowen Zhou, and Chao Zhang. BrainOmni: A brain foundation model for unified EEG and MEG signals. arXiv preprint arXiv:2505.18185, 2025. URL https://arxiv.org/abs/2505.18185.

Wei Xiong, Jiangtong Li, Jie Li, Kun Zhu, and Changjun Jiang. EEG-FM-Bench: A comprehensive benchmark for the systematic evaluation and diagnostic analyses of EEG foundation models. arXiv preprint arXiv:2508.17742v3, 2026. URL https://arxiv.org/abs/2508.17742v3. Version 3, revised 15 June 2026; first posted in 2025.

Chaoqi Yang, M. Brandon Westover, and Jimeng Sun. BIOT: Biosignal transformer for cross-data learning in the wild. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-3420. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f6b30f3e2dd9cb53bbf2024402d02295-Abstract.html.

Liuyin Yang, Qiang Sun, Ang Li, and Marc M. Van Hulle. Are EEG foundation models worth it? comparative evaluation with traditional decoders in diverse BCI tasks. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/f0f39b7686634fc81ca0112566b2c05f-Abstract-Conference.html.

Junjie Yu, Zihan Deng, Jianyu Zhang, Junrong Mu, Jiahui An, Wenxiao Ma, Ziling Lu, Yue Wang, Yan Zhu, Kexin Lou, and Quanying Liu. Understanding and Correcting Low-Frequency Bias in EEG Foundation Model. arXiv preprint arXiv:2608.01898, 2026. URL https://arxiv.org/abs/2608.01898.

Wei-Long Zheng and Bao-Liang Lu. A multimodal approach to estimating vigilance using EEG and forehead EOG. *Journal of Neural Engineering*, 14(2):026017, 2017. doi: 10.1088/1741-2552/aa5a98. URL https://doi.org/10.1088/1741-2552/aa5a98.

Yuchen Zhou, Jiamin Wu, Zichen Ren, Zhouheng Yao, Weiheng Lu, Kunyu Peng, Qihao Zheng, Chunfeng Song, Wanli Ouyang, and Chao Gou. CSBrain: A cross-scale spatiotemporal brain foundation model for EEG decoding. arXiv preprint arXiv:2506.23075, 2025. URL https://arxiv.org/abs/2506.23075.

Igor Zyma, Sergii Tukaev, Ivan Seleznov, Ken Kiyono, Anton Popov, Mariia Chernykh, and Oleksii Shpenkov. Electroencephalograms during mental arithmetic task performance. *Data*, 4(1):14, 2019. doi: 10.3390/data4010014. URL https://doi.org/10.3390/data4010014.

A Experimental Setup for Section 2

This appendix describes the encoder benchmark in Section 2. The main results use attention pooling and classical descriptors reduced by PCA. We report the other readouts and the experiments without PCA separately, so that each result is tied to the protocol that produced it.

A.1 Data and subject splits

We evaluate 34 datasets from eight families (Table 3). The five main families are motor imagery, ERP, SSVEP, sleep, and clinical tasks, covering 27 entries. Emotion, workload, and imagined speech provide seven additional entries. Each corpus is capped at 6,000 windows, sampled without replacement with seed 20260906. All representations compared within a corpus use the same retained windows and subject partitions.

The common preprocessing uses EEG channels, a 0.5–45 Hz band-pass, resampling to 200 Hz, and four-second windows. Each channel is standardized within each window by subtracting its temporal mean and dividing by its population standard deviation, with a floor of 10^{-8}. The classical descriptors and encoder inputs therefore start from the same normalized signal.

We use the common 19-channel montage whenever all its channels are available; otherwise, we use the native montage or a predefined subset. The common montage is FP1, FP2, F7, F3, FZ, F4, F8, T7, C3, CZ, C4, T8, P7, P3, PZ, P4, P8, O1, and O2. Channel counts for the six intervention datasets are reported in Table 15.

All windows and recordings from one person remain in the same partition. Datasets with at least ten subjects use one fixed split of approximately 45% training, 15% validation, and 40% test subjects; otherwise, we use leave-one-subject-out evaluation (LOSO). The split seed is 20260908. For subject-level classification, splits are stratified by the subject's label; TUAB uses the majority label of each person's recordings. BNCI2014-001 and Nakanishi2015 each use nine leave-one-subject-out folds, with six training, two validation, and one test subject per fold. The full annotation budget means all training and validation labels in the retained subset, not all recordings in the original dataset.

A.2 Frozen encoders and coverage

Table 4 lists the 23 encoder configurations. We use published checkpoints of REVE-base (El Ouahidi et al., 2025), CBraMod-base (Wang et al., 2025a), LUNA-base (Döner et al., 2025), EEGPT-large (Wang et al., 2024), LaBraM-base (Jiang et al., 2024), CSBrain-base (Zhou et al., 2025), EEGMamba-base (Wang et al., 2025b), FEMBA-base (Tegon et al., 2025), ST-EEGFormer-base (Yang et al., 2026), BIOT-six18 (Yang et al., 2023), BENDR-base (Kostas et al., 2021), Neuro-GPT-base (Cui et al., 2024), NeuroLM-base (Jiang et al., 2025), NeuroRVQ-base (Barmpas et al., 2025), CodeBrain (Ma et al., 2025), BrainOmni-base (Xiao et al., 2025), EEG-DINO-small and EEG-DINO-medium (Wang et al., 2025c), Signal-JEPA (Guetschel et al., 2024), LEAD (Wang et al., 2025d), OSF-base (Shuai et al., 2026), and ZUNA-1.1 (Warner et al., 2026). Each published checkpoint is paired with one randomly initialized instance of the same architecture. Both encoders remain frozen. This design isolates the contribution of pretraining to the learned representations: fine-tuning would modify these representations through task-specific adaptation, obscuring which properties were acquired during pretraining. Our objective is to characterize what pretraining has learned rather than to maximize downstream performance. The pipeline checks that parameter names, parameter shapes, and output-grid shapes agree between the two versions. The comparisons therefore change the encoder weights while keeping the architecture and downstream training procedure fixed.

Model-specific adapters receive the same windows, resampled with a polyphase FIR filter when required.

Table 3: Data retained for Section 2. Counts describe the evaluated subsets after preprocessing and subsampling.

Family	Dataset / task	Windows	Subjects	Classes
Motor imagery	BNCI2014-001 (Tangermann et al., 2012)	5184	9	4
	Cho2017 (Cho et al., 2017)	5997	52	2
	Dreyer2023 (Dreyer et al., 2023)	5978	87	2
	Lee2019 MI (Lee et al., 2019)	5994	54	2
	PhysioNet MI (Schalk, 2009; Goldberger et al., 2000)	6000	109	5
	Schirrmeister2017 (Schirrmeister et al., 2017)	6000	14	4
	Stieger2021 (Stieger et al., 2021)	6000	62	4
ERP	BNCI2014-009 (Aricò et al., 2014)	6000	10	2
	BNCI2015-003 (Guger et al., 2009)	6000	10	2
	ERP CORE MMN (Kappenman et al., 2021)	6000	40	2
	ERP CORE N170 (Kappenman et al., 2021)	6000	40	2
	ERP CORE N2pc (Kappenman et al., 2021)	6000	40	2
	Lee2019 ERP (Lee et al., 2019)	5994	54	2
SSVEP	BETA (Liu et al., 2020)	5995	55	40
	Kalunga2016 (Kalunga et al., 2016)	960	12	4
	Lee2019 SSVEP (Lee et al., 2019)	5994	54	4
	MAMEM3 (Nikolopoulos, 2021)	1375	11	5
	Nakanishi2015 (Nakanishi et al., 2015)	1620	9	12
Sleep	HMC (Alvarez-Estevez & Rijsman, 2021)	6000	151	5
	ISRUC (Khalighi et al., 2016)	6000	98	5
	Sleep-EDF (Kemp et al., 2000)	6000	78	5
Clinical	ADFTD (Miltiadous et al., 2023)	6000	88	3
	CHB-MIT (Guttag, 2010)	5432	23	2
	Mumtaz (Mumtaz, 2016)	4329	58	2
	PD Iowa (Anjum et al., 2024)	5904	149	2
	TUAB (Obeid & Picone, 2016)	6000	2326	2
	TUEV (Obeid & Picone, 2016)	6000	370	6
Emotion	DREAMER (Katsigiannis & Ramzan, 2018)	6000	23	2
	FACED (Chen et al., 2023)	6000	123	9
	SEED-V (Liu et al., 2022)	6000	16	5
Workload	EEGMAT (Zyma et al., 2019)	2134	36	2
	SEED-VIG (Zheng & Lu, 2017)	6000	21	3
	Shin2017B (Shin et al., 2017)	1740	29	2
Imagined speech	BCIC2020-3 (Jeong et al., 2022)	6000	15	5

Table 4: Encoder configurations and available model–corpus pairs. D is the token feature width. A, M, and E denote attention, global mean, and electrode-wise pooling. The maximum coverage is 34 corpora per configuration.

Encoder	Configuration	D	A	M	E
REVE	base	512	32	32	32
CBraMod	base	200	34	34	34
LUNA	base	256	32	32	0
EEGPT	large	512	34	34	0
LaBraM	base	200	31	31	31
CSBrain	base	200	34	34	34
EEGMamba	base	200	34	34	34
FEMBA	base	385	21	21	0
ST-EEGFormer	base	768	5	31	31
ZUNA	1.1 output	32	32	32	32
BIOT	six18	256	26	26	0
BENDR	base	512	32	32	0
Neuro-GPT	base	1024	32	32	0
NeuroLM	base	768	31	31	31
NeuroRVQ	base	800	31	31	31
CodeBrain	released	200	34	34	34
BrainOmni	base	512	32	32	0
EEG-DINO (small)	small	200	28	28	28
EEG-DINO (medium)	medium	512	28	28	28
Signal-JEPA	released	64	21	21	21
LEAD	released	128	32	32	32
OSF	base	768	34	34	0
ZUNA (hidden)	1.1 hidden	1024	2	32	32

A.3 Classical signal descriptors

Descriptors are computed separately for each normalized window. Band-based descriptors use six frequency bands: $[1, 4)$, $[4, 8)$, $[8, 13)$, $[13, 20)$, $[20, 30)$, and $[30, 45)$ Hz. We evaluate the following groups.

Band power. For each channel, we compute the full-window FFT, sum its power within each band, and take the natural logarithm after adding 10^{-12}. This gives six values per channel.

Spectrum. We concatenate log Welch power spectra (Welch, 1967) estimated with one-, two-, and four-second Hann segments, 50% overlap, and mean removal within each segment. Frequencies from 0.5 to 45 Hz are retained. Power is floored at 10^{-20} before taking the natural logarithm.

Covariance. We filter each window in each band with a fourth-order forward–backward Butterworth filter. We then compute $XX^\top/T$ on the whole window, its two halves, and its four quarters, where T is the number of samples in the relevant interval. These are second-moment matrices; the filtered subwindows are not centered again. Each matrix is shrunk toward the trace-scaled identity with coefficient 0.001, then mapped by the matrix logarithm, with eigenvalues floored at 10^{-12}. We retain the upper triangle and multiply off-diagonal entries by $\sqrt{2}$.

Coherence. We estimate complex coherency, $S_{ij}/\sqrt{S_{ii}S_{jj}}$, using one-second Hann segments, 50% overlap, and segment-wise mean removal. We average within each frequency band and retain the real and imaginary parts for every distinct channel pair. This descriptor is complex coherency, rather than magnitude-squared coherence alone.

The five descriptor variants are `bp` (band power), `spec` (spectrum), `cov` (covariance), `speccov` (spectrum and covariance), and `full` (spectrum, covariance, and coherence). The main results use `full`.

For C channels, the spectrum, covariance, and coherence groups contain $314C$, $42C(C+1)/2$, and $12C(C-1)/2$ values, respectively. The full descriptor therefore contains 15,998 values for

19 channels and 2,946 values for six channels. No subject-specific reference or alignment is fitted when computing these per-window descriptors.

Standardization and PCA. For each fold and descriptor variant, columns are standardized on the training and validation windows, without labels. Columns with standard deviation at most 10^{-6} are set to zero. Each group is divided by the square root of its number of active columns times the number of groups, giving the groups equal total weight. PCA is fitted to this representation on the same non-test windows.

If the encoder readout has width d, H_d retains the leading $\min(d, r)$ components, where r is the numerical rank. This dimensionality cap limits the imbalance between the handcrafted and encoder feature blocks and reduces downstream classifier training costs. The rank threshold is 10^{-8} times the largest eigenvalue of the centered Gram matrix. PCA scores are not whitened. Validation windows thus contribute to this label-free preprocessing, but test windows contribute to neither standardization nor PCA. Crucially, repeating the comparisons without PCA preserves the substantial reduction in the average pretraining advantage, although individual residual gains differ (Appendix B.5). Dimensionality reduction therefore improves computational efficiency while preserving the main qualitative conclusion.

A.4 Readouts and classifier fitting

We compare four arms: Z_0, Z_P, $[H_d, Z_0]$, and $[H_d, Z_P]$. The first two use features from the random and pretrained encoders. The other two also receive the same classical descriptors. Each arm selects its own hyperparameters on validation performance, using the same search space as its paired arm.

Main readout: attention pooling. We use single-query attention pooling as the main readout to accommodate the different token-grid sizes produced by the evaluated encoders while keeping the downstream head compact. Concatenating all tokens would make classifier size grow with the number of tokens, introducing substantial variation in downstream capacity across architectures and complicating the interpretation of their scores. Attention pooling instead produces one D-dimensional vector per window and allows task-dependent weighting of tokens using only D additional query parameters. This choice addresses the sensitivity to classifier design highlighted by EEG-FM-Bench (Xiong et al., 2026). It removes dependence of head size on token count, while the pretrained and random versions of each encoder retain exactly matched heads; feature widths and head sizes can still differ across architectures.

For tokens $z_i \in \mathbb{R}^D$, a single learned query q gives

$$a_i = \frac{\exp(q^\top z_i/\sqrt{D})}{\sum_j \exp(q^\top z_j/\sqrt{D})}, \qquad z_{\text{pool}} = \sum_i a_i z_i. \tag{3}$$

A linear classifier receives z_{pool}, or its concatenation with H_d. Here $d = D$. Only the query and classifier are trained; the encoder remains frozen. The query starts at zero, giving uniform attention at initialization.

Token features are standardized per dimension over the fitting windows and tokens. The PCA scores are centered on the fitting windows and scaled by one common scalar to unit average component variance. This scaling does not whiten individual components.

We use full-batch Adam, learning rate 0.01, and cross-entropy loss. The L2 penalty on the query and classification weights is selected from $\{0.001, 0.01, 0.1\}$. Validation κ is evaluated every five steps, up to 300 steps, to select the penalty and training duration. Ties between penalties favor stronger regularization; ties between steps retain the earlier step. The query and classifier are then reinitialized and refitted on training plus validation windows for the selected number of steps. We repeat this procedure with head seeds 0, 1, and 2 and average their test scores, rather than voting over their predictions.

Attention produces one vector per window, regardless of the number of tokens. It does not give every model the same feature width or head size. For K classes, the encoder-only head has $D + K(D+1)$ trainable parameters. With d_H retained classical components, it has $D + K(D + d_H + 1)$. Head sizes are matched between pretrained and random versions of each model, not across different models.

Global mean and electrode-wise pooling. Global mean pooling averages all tokens, giving width $d = D$. Electrode-wise pooling averages over time within each electrode and concatenates the results, giving width $d = CD$ when an electrode axis is available. In the experiments accompanying the main PCA results, these two readouts use linear ridge regression to one-hot targets. Predictions are the class with the largest score.

Encoder features are standardized on the fitting data and divided by the square root of their number of active dimensions. PCA scores are centered without component-wise rescaling. Ridge uses centered linear kernels and regularization $n_{\text{fit}}\alpha$, with $\alpha \in \{10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. For combined arms, the kernel is $(1-w)K_H + wK_Z$, with $w \in \{0, 0.1, 0.5, 0.9, 1\}$. The endpoints allow the classifier to ignore either block. Validation κ selects the hyperparameters; ties favor larger α, then smaller w. The selected classifier is refitted on training plus validation windows.

These comparisons change the pooling, classifier fitting procedure, and, for electrode-wise pooling, the width of H_d. They therefore do not isolate the effect of pooling alone. We address this with uniform-pooling controls and a further experiment using cross-entropy classifiers for all three readouts (Appendix B.6).

A.5 Scores, averages, and uncertainty

We report Cohen's κ over all held-out test windows in a model–corpus pair. For leave-one-subject-out evaluation, out-of-fold predictions are concatenated before scoring. Differences use the same test observations for the compared arms. The two contrasts are $\Delta_{\text{pre}} = s(Z_P) - s(Z_0)$ and $\Delta_H = s([H_d, Z_P]) - s([H_d, Z_0])$, as in Equations 1 and 2.

For each pair, 95% confidence intervals use 2,000 paired bootstrap resamples of test subjects. All windows from a sampled subject stay together, and the same resample is used for both arms. Attention scores are averaged across head seeds within each resample.

Each small model-level point in the main figure averages that model's available corpora within a family. The family summary instead averages all available model–corpus pairs. Its 95% interval uses 2,000 bootstrap resamples of the rows and columns of the model-by-corpus matrix, sampled independently with replacement, and averages the non-missing entries. This family bootstrap does not resample subjects again. The overall summary gives equal weight to the five main family means; it is not an average of every pair across all families.

B Additional Results for Section 2

B.1 Absolute scores and pretraining gains

Tables 5 and 6 give the absolute scores and paired gains for the main attention readout. Equal weighting of the five main families gives $\Delta_{\text{pre}} = +0.0698$ and $\Delta_H = +0.0024$, reported as $+0.070$ and $+0.002$ in the main text.

The absolute scores matter when interpreting a small difference. On ERP, both combined arms have mean $\kappa = 0.021$. Their similar scores do not establish that the classical descriptors capture everything needed for the task. Conversely, small average differences can hide positive and negative results: among the 652 attention pairs, 89 have a subject-bootstrap interval for Δ_H entirely above zero, and 67 entirely below zero. These are descriptive counts without multiple-comparison correction.

Absolute rankings do not isolate the contribution of pretraining. Absolute downstream scores and gains from pretrained weights can lead to different conclusions about the same models. Table 7 illustrates this distinction on four corpora using the main attention-pooling protocol, encoder features only, and the full annotation budget. Within each corpus, all entries use the same labeled windows and subject partitions. Both random and pretrained encoders remain frozen; only the attention query and linear classifier are trained. Thus, "random" refers to the encoder weights, not to an untrained downstream classifier.

On Stieger2021, randomly initialized REVE scores 0.256, above pretrained EEG-DINO (medium) at 0.125. Yet both architectures benefit from pretraining by almost the same amount: $+0.047$ and

Table 5: Absolute test κ with attention pooling and the full descriptor H_d. Values average the available pairs within each family. n is the number of model–evaluation dataset pairs.

Family	n	Z_0	Z_P	$[H_d, Z_0]$	$[H_d, Z_P]$
Motor imagery	138	0.075	0.104	0.220	0.221
ERP	113	0.018	0.015	0.021	0.021
SSVEP	100	0.120	0.175	0.387	0.389
Sleep	50	0.269	0.431	0.564	0.571
Clinical	115	0.258	0.363	0.511	0.514
Emotion	62	0.020	0.037	0.112	0.113
Workload	53	0.054	0.134	0.176	0.190
Imagined speech	21	-0.001	0.005	0.014	0.016

Table 6: Pretraining gains with attention pooling. Intervals are the family bootstrap intervals described in Appendix A.5. Bounds very close to zero retain an extra decimal place to show their sign. The last three families are additional evaluations.

Family	Δ_{pre}	95% interval	Δ_H	95% interval
Motor imagery	$+0.029$	$[-0.007, +0.069]$	$+0.000$	$[-0.005, +0.006]$
ERP	-0.002	$[-0.015, +0.012]$	$+0.000$	$[-0.010, +0.013]$
SSVEP	$+0.055$	$[-0.0003, +0.118]$	$+0.002$	$[-0.013, +0.016]$
Sleep	$+0.162$	$[+0.094, +0.281]$	$+0.007$	$[-0.0001, +0.021]$
Clinical	$+0.105$	$[+0.051, +0.171]$	$+0.002$	$[-0.016, +0.020]$
Emotion	$+0.018$	$[+0.007, +0.030]$	$+0.001$	$[-0.002, +0.004]$
Workload	$+0.080$	$[+0.036, +0.120]$	$+0.014$	$[+0.003, +0.026]$
Imagined speech	$+0.006$	$[+0.002, +0.010]$	$+0.002$	$[-0.002, +0.005]$

$+0.048$, respectively. Both produce 512-dimensional features and use 2,564 trainable readout parameters, so this example does not arise from a difference in head size. The much higher absolute score of pretrained REVE (0.304 versus 0.125) coexists with a similar within-architecture pretraining gain.

On Lee2019 SSVEP, random BIOT reaches 0.866, above pretrained REVE (0.816) and CBraMod (0.674). Nevertheless, pretraining improves REVE by $+0.233$ and CBraMod by $+0.611$ relative to their own random encoders. REVE therefore has the higher pretrained score, while CBraMod has the larger gain from pretrained weights. BIOT provides the converse example: its pretrained score of 0.645 accompanies a decrease of 0.220 from its random reference. A high absolute score can therefore coexist with either a substantial positive or a negative pretraining gain.

The distinction also appears in sleep and clinical evaluation. On ISRUC, random EEGPT scores 0.448, above pretrained CSBrain at 0.417. Pretrained EEGPT also ranks higher (0.494), although its gain is smaller: $+0.046$ versus $+0.246$ for CSBrain. On TUAB, random BIOT scores 0.477, above pretrained CSBrain at 0.415, while their respective pretraining gains are $+0.071$ and $+0.227$. These comparisons show that a pretrained encoder can benefit substantially from pretraining even when a different architecture with random weights scores higher. The cross-architecture orderings reported here are descriptive comparisons of observed scores, not claims of statistically significant superiority.

For two architectures A and B evaluated on the same corpus, the distinction can be written exactly as

$$s(Z_P^A) - s(Z_P^B) = \big[s(Z_0^A) - s(Z_0^B)\big] \quad + \big[\Delta_{\mathrm{pre}}^A - \Delta_{\mathrm{pre}}^B\big]. \tag{4}$$

An absolute ranking therefore combines differences already present with random encoders and differences in the gains from pretraining. Random-encoder performance reflects the architecture, input representation, initialization, and supervised readout together; it is not a measure of architecture alone. Tokenization, channel handling, positional information, feature width, and compatibility with the readout can all affect the resulting pipeline.

Absolute scores remain useful for selecting a pipeline under a specified evaluation protocol, but cannot by themselves identify what pretraining contributes. We therefore report them alongside paired pretrained–random contrasts with the architecture and downstream training procedure held fixed.

Table 7: Selected examples contrasting absolute performance and the contribution of pretraining. Scores are test Cohen's κ, averaged over three downstream-head seeds. D is the pooled feature width. Each gain compares pretrained and random weights of the same architecture; its interval uses 2,000 paired test-subject bootstrap resamples (Appendix A.5). Differences are computed before rounding. These intervals concern within-architecture gains, not cross-architecture score differences, and do not include variation across encoder initializations or pretraining runs.

Corpus	Encoder	D	$s(Z_0)$	$s(Z_P)$	Δ_{pre} [95% interval]
Stieger2021	REVE	512	0.256	0.304	$+0.047$ $[+0.019, +0.080]$
	EEG-DINO (medium)	512	0.077	0.125	$+0.048$ $[+0.024, +0.074]$
Lee2019 SSVEP	BIOT	256	0.866	0.645	-0.220 $[-0.280, -0.165]$
	REVE	512	0.583	0.816	$+0.233$ $[+0.186, +0.281]$
	CBraMod	200	0.063	0.674	$+0.611$ $[+0.537, +0.680]$
ISRUC	EEGPT	512	0.448	0.494	$+0.046$ $[+0.020, +0.072]$
	CSBrain	200	0.171	0.417	$+0.246$ $[+0.209, +0.285]$
TUAB	BIOT	256	0.477	0.549	$+0.071$ $[+0.037, +0.106]$
	CSBrain	200	0.188	0.415	$+0.227$ $[+0.184, +0.269]$

The same reasoning motivates Δ_H: once both classifiers receive identical classical descriptors, their paired difference measures the remaining predictive benefit of pretrained weights. Neither contrast ranks pretraining objectives in isolation, since the released models also differ in their pretraining data and training recipes.

B.2 Which classical descriptors reduce the gain?

Table 8 reports each descriptor variant under each readout. With attention, spectra alone reduce the sleep and clinical gains to about $+0.016$. Adding covariance reduces them to $+0.006$ and $+0.001$, respectively. Adding coherence gives $+0.007$ and $+0.002$, so it does not reduce the remaining gain further in these two families.

These comparisons need not decrease monotonically. Adding a group changes the relative weight of the existing groups and can change which directions PCA retains. The variants are finite descriptions of the signal, not an exhaustive representation of every spectral and spatial property.

Table 8: Descriptor ablation. Without H, values are Δ_{pre}; the other columns give Δ_H for the indicated descriptor reduced to the readout width. Each row averages all available pairs for that readout and family. A dash means the combination was not evaluated. Global mean and electrode-wise readouts in this table use ridge; attention uses cross-entropy.

Readout	Family	Without H	bp	spec	cov	speccov	full
Attention	Motor imagery	$+0.029$	—	$+0.006$	—	-0.001	$+0.000$
	ERP	-0.002	—	$+0.002$	—	$+0.000$	$+0.000$
	SSVEP	$+0.055$	—	-0.002	—	-0.001	$+0.002$
	Sleep	$+0.162$	—	$+0.016$	—	$+0.006$	$+0.007$
	Clinical	$+0.105$	—	$+0.016$	—	$+0.001$	$+0.002$
Global mean	Motor imagery	$+0.015$	-0.001	$+0.003$	-0.002	$+0.002$	-0.003
	ERP	-0.006	-0.004	-0.002	-0.005	-0.004	-0.004
	SSVEP	$+0.020$	-0.002	$+0.001$	-0.007	-0.005	$+0.006$
	Sleep	$+0.149$	$+0.033$	$+0.034$	$+0.024$	$+0.013$	$+0.018$
	Clinical	$+0.090$	$+0.043$	$+0.034$	$+0.040$	$+0.028$	$+0.030$
Electrode-wise	Motor imagery	$+0.047$	$+0.024$	$+0.029$	-0.002	$+0.003$	$+0.005$
	ERP	-0.011	-0.010	-0.010	-0.010	-0.011	-0.012
	SSVEP	$+0.016$	$+0.004$	$+0.003$	$+0.004$	$+0.001$	$+0.007$
	Sleep	$+0.185$	$+0.039$	$+0.039$	$+0.026$	$+0.020$	$+0.020$
	Clinical	$+0.119$	$+0.041$	$+0.025$	$+0.047$	$+0.003$	$+0.035$

B.3 RESULTS BY CORPUS AND ENCODER

Table 9 gives the four absolute scores and two contrasts for every corpus, averaging available encoders. Tables 10 and 11 give encoder-level means in the five main families. Coverage differs across encoders, so these means should not be treated as a ranking on a common set of datasets. A missing entry indicates no available result in that family.

Table 9: Results by corpus with attention pooling and H_d from the full descriptor. Each row averages n encoder configurations. Contrasts are computed before rounding.

Corpus / task	n	Z_0	Z_P	$[H_d, Z_0]$	$[H_d, Z_P]$	Δ_{pre}	Δ_H
BNCI2014-001	16	0.056	0.095	0.187	0.193	+0.039	+0.006
Cho2017	21	0.066	0.080	0.164	0.162	+0.014	-0.002
Dreyer2023	17	0.052	0.076	0.233	0.233	+0.024	+0.000
Lee2019 MI	21	0.039	0.056	0.237	0.239	+0.017	+0.003
PhysioNet MI	21	0.077	0.081	0.190	0.186	+0.005	-0.004
Schirrmeister2017	21	0.130	0.208	0.315	0.320	+0.079	+0.004
Stieger2021	21	0.095	0.123	0.213	0.210	+0.027	-0.003
BNCI2014-009	19	0.057	0.053	0.054	0.052	-0.003	-0.002
BNCI2015-003	19	-0.002	0.000	0.003	0.004	+0.002	+0.001
ERP CORE MMN	18	0.002	0.001	-0.004	-0.009	-0.001	-0.004
ERP CORE N170	18	0.018	0.022	0.018	0.028	+0.004	+0.010
ERP CORE N2pc	18	0.026	0.013	0.052	0.051	-0.013	-0.001
Lee2019 ERP	21	0.006	0.005	0.004	0.001	-0.002	-0.003
BETA	21	0.049	0.069	0.221	0.220	+0.020	-0.002
Kalunga2016	19	0.072	0.168	0.362	0.382	+0.096	+0.020
Lee2019 SSVEP	21	0.271	0.377	0.654	0.653	+0.106	-0.001
MAMEM3	20	0.050	0.051	0.109	0.107	+0.000	-0.002
Nakanishi2015	19	0.152	0.207	0.593	0.588	+0.055	-0.005
HMC	21	0.218	0.361	0.516	0.522	+0.143	+0.006
ISRUC	21	0.298	0.432	0.591	0.595	+0.134	+0.004
Sleep-EDF	8	0.327	0.610	0.619	0.640	+0.284	+0.021
ADFTD	21	0.070	0.161	0.314	0.321	+0.092	+0.007
CHB-MIT	10	0.257	0.345	0.497	0.523	+0.088	+0.026
Mumtaz	21	0.528	0.651	0.847	0.830	+0.123	-0.018
PD Iowa	21	0.132	0.175	0.332	0.311	+0.044	-0.021
TUAB	21	0.298	0.437	0.568	0.578	+0.139	+0.010
TUEV	21	0.262	0.398	0.501	0.524	+0.136	+0.023
DREAMER	20	0.007	0.022	0.106	0.107	+0.015	+0.001
FACED	21	0.027	0.042	0.112	0.112	+0.015	+0.000
SEED-V	21	0.024	0.048	0.119	0.120	+0.023	+0.001
EEGMAT	21	0.069	0.165	0.181	0.197	+0.096	+0.016
SEED-VIG	18	0.066	0.145	0.169	0.180	+0.078	+0.011
Shin2017B	14	0.015	0.071	0.178	0.192	+0.056	+0.014
BCIC2020-3	21	-0.001	0.005	0.014	0.016	+0.006	+0.002

B.4 COMPARING READOUTS ON THE SAME MODEL–DATASET PAIRS

The three readouts have different coverage. We therefore repeat the summary using only the 409 model–dataset pairs available under all three, including 325 pairs in the five main families (Table 12). The reduction in gain remains visible on this common set. The residual still depends on the readout: in clinical tasks it is $+0.002$ with attention, $+0.036$ with global mean pooling, and $+0.037$ with electrode-wise pooling. Matching coverage removes differences in which pairs are evaluated; it does not remove differences in classifier fitting or in H_d's width.

B.5 USING CLASSICAL DESCRIPTORS WITHOUT PCA

Because we used PCA to reduce compute time and to place the descriptor groups on an equal footing, this section checks that it did not introduce any evaluation artifact. To that end, we repeat the combined-arm comparison on the full descriptor matrix without reducing its dimension, and show that the results are unchanged. Two choices are involved here and should not be confused: "full descriptor" refers to the set of descriptor groups used (spectrum, covariance, and coherence), while "without PCA" refers to keeping all of their columns rather than projecting them onto principal components.

Table 10: Gain from pretraining without classical descriptors, Δ_{pre}, by encoder and family. Attention pooling; each entry averages that encoder's available corpora.

Encoder	Motor imagery	ERP	SSVEP	Sleep	Clinical
REVE	+0.160	+0.084	+0.088	+0.247	+0.189
CBraMod	+0.085	+0.002	+0.190	+0.271	+0.163
LUNA	+0.049	+0.000	−0.031	−0.082	−0.010
EEGPT	−0.020	+0.003	+0.074	+0.038	−0.015
LaBraM	+0.044	−0.015	+0.022	+0.039	+0.071
CSBrain	+0.032	+0.004	+0.141	+0.312	+0.186
EEGMamba	+0.033	−0.011	+0.098	+0.181	+0.053
FEMBA	+0.028	+0.010	+0.115	—	+0.054
ST-EEGFormer	—	+0.022	+0.228	+0.086	—
ZUNA	−0.024	+0.010	+0.042	+0.052	+0.050
BIOT	−0.093	−0.009	−0.310	—	+0.069
BENDR	−0.004	−0.006	+0.028	−0.088	−0.053
Neuro-GPT	−0.166	−0.031	−0.038	+0.154	+0.079
NeuroLM	+0.136	+0.014	+0.147	+0.299	+0.350
NeuroRVQ	−0.009	−0.026	−0.060	+0.016	+0.014
CodeBrain	+0.028	−0.016	−0.015	+0.303	+0.096
BrainOmni	+0.170	+0.004	+0.223	+0.443	+0.475
EEG-DINO (small)	+0.020	−0.025	+0.008	+0.247	+0.128
EEG-DINO (medium)	+0.080	−0.023	+0.040	+0.213	+0.127
Signal-JEPA	−0.044	+0.002	−0.016	−0.104	−0.130
LEAD	+0.034	−0.006	+0.104	+0.184	+0.155
OSF	+0.045	−0.031	+0.099	+0.275	+0.159
ZUNA (hidden)	—	—	—	+0.043	—

Table 11: Remaining pretraining gain, Δ_H, by encoder and family. Attention pooling with the full descriptor reduced to H_d; each entry averages that encoder's available corpora.

Encoder	Motor imagery	ERP	SSVEP	Sleep	Clinical
REVE	+0.006	+0.066	+0.013	+0.012	−0.003
CBraMod	+0.012	+0.007	+0.018	+0.015	+0.011
LUNA	+0.007	−0.002	−0.010	−0.009	−0.025
EEGPT	+0.003	+0.000	+0.009	+0.006	+0.008
LaBraM	+0.008	−0.016	+0.002	−0.012	+0.005
CSBrain	+0.002	+0.020	+0.012	−0.006	+0.004
EEGMamba	+0.008	−0.014	−0.005	+0.005	+0.007
FEMBA	+0.005	+0.006	+0.003	—	+0.007
ST-EEGFormer	—	−0.007	+0.026	+0.023	—
ZUNA	+0.001	+0.002	+0.002	+0.002	−0.014
BIOT	−0.015	+0.001	−0.127	—	+0.042
BENDR	+0.008	−0.009	+0.007	−0.005	+0.003
Neuro-GPT	−0.027	−0.026	−0.019	+0.001	−0.015
NeuroLM	−0.003	+0.009	−0.022	+0.000	−0.014
NeuroRVQ	−0.004	−0.021	+0.011	−0.002	+0.015
CodeBrain	+0.008	+0.002	+0.002	−0.005	−0.010
BrainOmni	+0.002	+0.002	+0.002	+0.001	+0.003
EEG-DINO (small)	−0.002	−0.018	+0.008	+0.038	−0.026
EEG-DINO (medium)	+0.001	−0.009	+0.016	+0.026	+0.000
Signal-JEPA	−0.009	−0.004	−0.010	+0.005	+0.001
LEAD	+0.004	+0.005	+0.017	+0.012	+0.058
OSF	−0.009	−0.012	+0.043	+0.013	−0.010
ZUNA (hidden)	—	—	—	+0.016	—

Table 12: Readout comparison on the same pairs. Each readout uses the main PCA protocol, including its original classifier fitting procedure. n is identical across readouts within a row.

		Attention		Global mean		Electrode-wise	
Family	n	Δ_{pre}	Δ_H	Δ_{pre}	Δ_H	Δ_{pre}	Δ_H
Motor imagery	85	$+0.046$	$+0.003$	$+0.026$	$+0.001$	$+0.061$	$+0.010$
ERP	68	$+0.002$	$+0.004$	-0.003	-0.002	-0.008	-0.009
SSVEP	65	$+0.068$	$+0.006$	$+0.027$	$+0.010$	$+0.024$	$+0.009$
Sleep	36	$+0.175$	$+0.009$	$+0.169$	$+0.020$	$+0.185$	$+0.020$
Clinical	71	$+0.113$	$+0.002$	$+0.110$	$+0.036$	$+0.130$	$+0.037$
Emotion	39	$+0.024$	$+0.002$	$+0.026$	$+0.000$	$+0.047$	-0.001
Workload	32	$+0.103$	$+0.019$	$+0.105$	$+0.012$	$+0.102$	$+0.011$
Imagined speech	13	$+0.007$	$+0.004$	$+0.010$	$+0.012$	$+0.006$	$+0.007$

Groups are standardized and given equal total weight as above. In these no-PCA experiments, this preprocessing is fitted on training windows during selection and on training plus validation windows for the final fit. For attention, the concatenated descriptor matrix is scaled to total variance D before entering the classifier. This matches the total variance of standardized token dimensions, but does not guarantee equal variance for the descriptor block and the pooled encoder vector.

Table 13 compares only pairs available in both versions. With global mean pooling and the same ridge search, the mean change in Δ_H over 230 common pairs is -0.0010. The attention comparison contains 611 common pairs. Its clinical residual increases from $+0.002$ to $+0.013$, while the sleep residual changes from $+0.007$ to $+0.009$. Thus, removing PCA preserves the broad reduction in pretraining gain, but does not give identical residuals. These results also include the stated change in how descriptor preprocessing is fitted.

Table 13: Pretraining gain conditional on classical descriptors, with and without PCA. Each row uses exactly the same model–corpus pairs on both sides. The global mean comparison uses ridge in both versions; the attention comparison uses cross-entropy in both.

Readout	Family	n	Δ_H, PCA	Δ_H, no PCA
Attention	Motor imagery	137	$+0.000$	-0.002
	ERP	113	$+0.000$	-0.002
	SSVEP	60	$+0.005$	$+0.002$
	Sleep	50	$+0.007$	$+0.009$
	Clinical	115	$+0.002$	$+0.013$
	Emotion	62	$+0.001$	-0.001
	Workload	53	$+0.014$	$+0.007$
	Imagined speech	21	$+0.002$	$+0.000$
Global mean	Motor imagery	44	-0.006	-0.002
	ERP	37	-0.004	-0.007
	SSVEP	21	-0.006	-0.006
	Sleep	48	$+0.016$	$+0.016$
	Clinical	39	$+0.017$	$+0.013$
	Emotion	41	$+0.000$	-0.002

B.6 Separating pooling from classifier fitting

Uniform pooling with the attention training procedure. We replace learned attention by fixed uniform token weights and keep the same linear head, normalization, optimizer, regularization search, and validation selection. This control is evaluated without H. For pretrained encoders, learned attention improves average κ over this control by 0.031 on motor imagery and 0.051 on SSVEP; the gains on ERP, sleep, and clinical tasks are 0.002, 0.006, and 0.007. For random encoders, the corresponding gains are 0.016, 0.030, -0.001, -0.004, and 0.014, respectively. These results show that pooling can affect decoding, but do not isolate the effect of learned attention when H is also supplied.

A common classifier fitting procedure for all three readouts. In a further evaluation without PCA, global mean and electrode-wise features are read by the same cross-entropy training procedure as attention, with no learned query for these deterministic poolings. We use the same learning rate, L2 search, validation selection, refitting procedure, and three head seeds. Descriptor groups have equal weight and are scaled to total variance equal to the encoder readout width.

Table 14 reports the 340 pairs available under all three readouts in this evaluation, including 263 in the five main families. Classical descriptors again reduce the average pretraining gain, while some residuals remain, especially on SSVEP and workload. This comparison supports the overall reduction under a common classifier fitting procedure; it does not require the residual to be zero for every task or readout.

Table 14: Common-pair comparison using cross-entropy classifiers for all three readouts and classical descriptors without PCA. Each row uses the same pairs across its six score columns. This is a separate evaluation from the main PCA results.

		Attention		Global mean		Electrode-wise	
Family	n	Δ_{pre}	Δ_H	Δ_{pre}	Δ_H	Δ_{pre}	Δ_H
Motor imagery	76	$+0.044$	-0.001	$+0.023$	$+0.003$	$+0.058$	$+0.010$
ERP	62	-0.005	-0.003	-0.003	-0.005	-0.002	-0.006
SSVEP	28	$+0.081$	$+0.012$	$+0.043$	$+0.035$	$+0.041$	$+0.015$
Sleep	32	$+0.181$	$+0.013$	$+0.161$	$+0.015$	$+0.177$	$+0.015$
Clinical	65	$+0.099$	$+0.010$	$+0.093$	-0.007	$+0.098$	-0.003
Emotion	32	$+0.022$	-0.001	$+0.023$	$+0.001$	$+0.036$	$+0.006$
Workload	32	$+0.103$	$+0.017$	$+0.092$	$+0.015$	$+0.091$	$+0.014$
Imagined speech	13	$+0.007$	$+0.000$	$+0.002$	$+0.000$	$+0.006$	-0.001

C Experimental setup for Section 3

We use two complementary experiments. First, we change the signal used for self-supervised pretraining, using different synthetic signals, then evaluate the resulting models on real EEG. The main comparison uses masked signal reconstruction; an extension compares masked reconstruction, JEPA-style latent prediction, and their combination while keeping the REVE-small encoder architecture fixed. A separate control increases the encoder size to REVE-base under the reconstruction objective. Second, we train EEGNetv4 from scratch and apply the same signal transformation to its training, validation, and test data. The first experiment measures transfer from pretraining; the second measures supervised decoding of each signal version. They share the downstream windows and subject partitions. The transformations and their numerical validation are described in Appendix D; results for the additional objectives appear in Appendix E.2.

C.1 Data and subject splits

Pretraining uses TUEG recordings with the 19-channel montage listed in Appendix A. Real windows are filtered to 0.5–45 Hz, sampled at 200 Hz, and standardized separately for each channel within each four-second window. Each window contains 800 samples per channel. Synthetic versions are generated from these preprocessed windows. For all TUEG pretraining experiments conducted in this work, we excluded recordings belonging to TUAB or any other downstream evaluation dataset. Both real and synthetic pretraining conditions use this same filtered corpus, ensuring no recording overlap between pretraining and downstream evaluation.

The six downstream corpora and their retained sample sizes are given in Table 15. All methods use the same retained windows. BCI IV-2a uses nine leave-one-subject-out folds, with six training, two validation, and one test subject in each fold. The other corpora use one fixed subject split, approximately 45%/15%/40% for training, validation, and testing (seed 20260908). TUAB splits are stratified using each person's majority label. All windows from a person stay in one partition within a fold. The window-sampling seed is 20260906.

We retain 22 channels for BCI IV-2a, six for ISRUC, and the common 19-channel montage for the remaining corpora. The BCI IV-2a channels are Fz, FC3, FC1, FCz, FC2, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CP1, CPz, CP2, CP4, P1, Pz, P2, and POz. The ISRUC channels are F3, C3, O1, F4,

Table 15: Retained downstream data. The last column counts training, validation, and test subjects. BCI IV-2a uses nine folds; the other corpora use one fixed split.

Corpus	Windows	Subjects	Channels	Classes	Subject split
BCI IV-2a	5,184	9	22	4	6/2/1 (each fold)
PhysioNet MI	6,000	109	19	5	49/16/44
Lee2019 ERP	5,994	54	19	2	24/8/22
Lee2019 SSVEP	5,994	54	19	4	24/8/22
ISRUC	6,000	98	6	5	44/15/39
TUAB	6,000	2326	19	2	1046/349/931

C4, and O2. TUEG is additionally evaluated on 1,024 fixed real windows for reconstruction only. This TUEG evaluation is a reference on the pretraining corpus, not evidence of transfer to an unseen corpus. Subject separation in downstream evaluation does not, by itself, establish separation from pretraining subjects or recordings.

C.2 Architecture and masked-reconstruction pretraining

The five MAE signal conditions use the same REVE-small MAE architecture. Its encoder has four Transformer layers, eight attention heads, and 512-dimensional tokens. The decoder has two layers, eight heads, and width 512. A token represents 200 samples from one electrode. Consecutive patches overlap by 20 samples, giving four patches per channel, starting at 0, 0.9, 1.8, and 2.7 seconds. Thus, patch extraction covers the first 740 samples of the 800-sample input; the signal transformation acts on the full window before patch extraction.

Training randomly masks 75% of electrode–patch tokens. The loss is the mean absolute error (L1) on the masked target patches. It is not the squared error used to report reconstruction scores. For synthetic conditions, a new signal is drawn whenever a window is presented, before masking. Both the visible input and the reconstruction target come from this same transformed window.

We use AdamW with a constant learning rate of 10^{-3}, weight decay 0.05, $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, and batch size 2,048. The standard-budget comparison uses approximately 20.8 million window presentations per run; checkpoint counts range from 20,819,968 to 20,847,310. Real EEG, PSD + cross-spectra, Covariance, and PSD each use three REVE-small MAE pretraining seeds. Mean + variance uses one pretraining run. Real EEG and PSD + cross-spectra are also evaluated with one run each at 104,236,550 presentations, five times the nominal standard budget of 20,847,310. These counts include repeated presentations of source windows.

C.3 Additional objectives and encoder-size control

Objective comparison at fixed encoder architecture. We retain the REVE-small encoder, tokenization, positional encoding, 75% masking rate, and two-layer decoder/predictor described above, and vary the pretraining objective. The three objectives are masked signal reconstruction (MAE), masked latent prediction (JEPA), and their sum (JEPA + reconstruction). The downstream encoder and its token grid are the same architecture in all three cases. The latent objectives add an exponential-moving-average (EMA) teacher and a latent output projection used only during pretraining; neither is used by the downstream readout.

Let u denote the real or transformed input window and $\mathcal{M}$ its masked token indices. The student encoder, with parameters θ, encodes the visible tokens, and the two-layer predictor produces an output h_i at each masked position. A teacher encoder with parameters $\bar{\theta}$ receives the complete version of the same window. Its final-layer token t_i is standardized over the feature dimension:

$$\bar{t}_i = \frac{t_i - \mathrm{mean}_d(t_i)}{\sqrt{\mathrm{var}_d(t_i) + 10^{-6}}}, \qquad \bar{\theta} \leftarrow 0.999\bar{\theta} + 0.001\theta, \tag{5}$$

where the variance uses denominator D. The teacher is initialized from the student encoder, receives no gradients, and uses clean electrode positions without the student's training-time positional jitter. We predict $\bar{t}_i$ through a linear latent projection g_P, with smooth L1 loss averaged over masked

tokens and features:

$$L_P = \frac{1}{|\mathcal{M}|D} \sum_{i \in \mathcal{M}} \sum_{d=1}^{D} \ell_{\mathrm{SL1}}\big([g_P(h_i)]_d - \bar{t}_{id}\big), \qquad \ell_{\mathrm{SL1}}(e) = \begin{cases} e^2/2, & |e| < 1, \\ |e| - 1/2, & |e| \geq 1. \end{cases} \tag{6}$$

Writing L_R for the masked-patch L1 reconstruction loss, we train with

$$L_{\mathrm{MAE}} = L_R, \qquad L_{\mathrm{JEPA}} = L_P, \qquad L_{\mathrm{JEPA+recon}} = L_P + L_R. \tag{7}$$

Both terms of the combined objective use the same masked positions and unit coefficients. Pure JEPA uses no signal-reconstruction loss. No additional anti-collapse regularizer is applied. These are controlled implementations of latent prediction and latent prediction combined with reconstruction, rather than reproductions of a complete published JEPA or EEGPT training recipe.

Signal conditions and size control. For each objective, we compare real EEG, PSD + cross-spectra, Covariance, and PSD. Synthetic windows are drawn afresh before masking; the student input, teacher input, and any signal target derive from the same draw. At a fixed signal condition, the REVE-small comparison changes the objective and its pretraining-only prediction components while retaining the encoder architecture, data, masking, optimizer settings, and downstream evaluation. At a fixed objective, the real–synthetic comparison changes the pretraining signal.

The separate size control uses REVE-base: 22 Transformer layers, eight attention heads, width 512, and approximately 69.2 million encoder parameters, compared with four layers and 12.7 million parameters for REVE-small. It retains the same patch layout, masking rate, and two-layer decoder and uses masked reconstruction under the same four signal conditions. The REVE-small grid comprises 36 pretraining runs (three objectives times four signals times three seeds); REVE-base uses four runs (one per signal). The mean-and-variance condition is evaluated only for REVE-small MAE, with one pretraining run.

Training budget and initialization. All conditions use the TUEG store and AdamW settings above, with batch size 2,048. Each REVE-small objective–signal condition uses three pretraining seeds, indexed 0, 1, and 2; each REVE-base signal condition uses one run. Standard-budget checkpoint counts lie between 20,819,968 and 20,847,310 window presentations, a difference of 0.13%. The presentation budgets are therefore approximately matched; computational cost differs because the latent objectives require a teacher forward pass and REVE-base has a deeper encoder. Initialization and stochastic training draws vary across pretraining seeds, while the order of windows in the pre-shuffled store remains fixed. The individual random checkpoints A and B are reported as descriptive references; the aggregate random-encoder reference summarizes three encoder initializations. The comparisons control architecture and training recipe without assuming identical starting weights or stochastic trajectories across all signal conditions.

Shared downstream evaluation. Every configuration uses the retained windows and subject partitions in Table 15 and the three frozen readouts in Appendix C.5. All stages of classifier fitting and evaluation use complete real EEG windows, without supplying an additional classical-descriptor block. Every pretraining seed is evaluated separately with this same protocol. The MAE results in the main signal comparison and the objective comparison use the same three runs per condition.

We also evaluate the published REVE-base encoder as an external frozen-transfer reference under this same downstream protocol. Its original pretraining data and computational budget are not matched to the controlled runs. It is assigned no reconstruction score in this evaluation.

C.4 Reconstruction on real EEG

Every model with a trained signal-reconstruction head reconstructs real EEG, regardless of its pretraining signal. This includes the MAEs, REVE-small with JEPA + reconstruction, and the REVE-base MAEs. Pure JEPA has no trained signal-reconstruction head and is not assigned a reconstruction score. We use a random mask of 75% of tokens, with the same evaluation mask for every model on a given window. Masks use seed 20260910, incremented by $1009b$ for reconstruction batch b, with batches of 128 windows. Predicted patches are mapped back to the signal before scoring, averaging predictions where masked patches overlap. We score only samples covered by a masked patch and by no visible patch; samples outside the extracted patches are excluded.

For each corpus, the reconstruction score is

$$R_0^2 = 1 - \frac{\sum_w \sum_{(c,t)\in M_w} (x_{wct} - \hat{x}_{wct})^2}{\sum_w \sum_{(c,t)\in M_w} x_{wct}^2}, \tag{8}$$

where M_w is the set of scored hidden samples. This compares the prediction with zero; the denominator is not centered by the mean of the hidden samples. For each pretraining run and corpus, we sum errors and target energies over windows before taking their ratio. For the overall reconstruction score, we average the six corpus scores equally within each pretraining run, then report the mean and sample standard deviation across the three pretraining runs when available. Corpus-specific entries instead summarize the three reconstruction scores for that corpus. The archive also contains the mean and standard deviation of separate per-window ratios; these are different summaries and are not substituted for the pooled score.

C.5 Frozen-feature and supervised readouts

The frozen encoder receives complete real windows without masking. We compare three ways to read its token grid: averaging all tokens (*mean pooling*); averaging time within each electrode and concatenating the electrode vectors (*electrodes kept*); and learning one attention query over all tokens (*attention pooling*). All three are followed by a linear classifier. No encoder parameter is updated during downstream fitting.

For mean pooling and electrodes kept, the classifier is ridge regression on one-hot labels. Feature scaling is fitted on training data, with constant columns removed from the effective feature count. The linear kernel is divided by the number of active features. We select $\alpha \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ by validation κ, using a ridge penalty of $n\alpha$ for n fitting windows. We then refit scaling and the classifier on training and validation windows together. Class prediction uses the largest output.

For attention pooling, feature means and sample standard deviations are fitted over training tokens. Standard deviations at most 10^{-6} are replaced by one. A learned query q gives token z_i weight $a_i = \mathrm{softmax}_i(q^\top z_i/\sqrt{D})$; the classifier receives $\sum_i a_i z_i$. Here $D = 512$ for all REVE configurations in this appendix. The query is randomly initialized, with each component drawn from $\mathcal{N}(0, 1/D)$. We train the query and linear head with unweighted cross-entropy, Adam, learning rate 5×10^{-4}, weight decay 10^{-3}, and batch size 128. Training lasts at most 80 epochs, with patience 10 on validation κ; an increase greater than 10^{-4} updates the retained state. We evaluate the best validation checkpoint without refitting on validation data. The three attention seeds are 20260908, 20260909, and 20260910.

EEGNetv4 is trained from scratch with AdamW, learning rate 10^{-3}, weight decay 10^{-4}, batch size 128, unweighted cross-entropy, and gradient-norm clipping at 1. Training lasts at most 100 epochs, with patience 15 on validation κ. The best validation checkpoint is used for testing. The implementation uses the EEGNetv4 defaults of eight temporal filters, depth multiplier two, 16 output filters, and a 64-sample temporal kernel. We use three seed indices $s \in \{0, 1, 2\}$; the seed in fold f is $1000s + f$. Synthetic training windows are drawn afresh in every batch; validation and test use fixed draws for each seed and fold.

Table 16 gives the trainable parameter counts. For C channels and K classes, the three frozen readouts have $513K$, $512CK + K$, and $512 + 513K$ parameters. These counts exclude the pretrained encoder and apply to both REVE-small and REVE-base because their token widths are equal. The similar sizes of attention pooling and EEGNetv4 do not make their architectures, training procedures, or total computation equivalent.

C.6 Scores and uncertainty

We compute Cohen's κ from the concatenated test predictions within each corpus. For leave-one-subject-out evaluation, each subject contributes predictions from its test fold. For each pretrained encoder and corpus, attention-pooling scores first average the three classifier initializations; we average scores, not predictions. The two ridge readouts yield one score per corpus and pretrained encoder.

Table 16: Trainable classifier parameters. Frozen-readout counts exclude the encoder; EEGNetv4 counts include the entire supervised network.

Corpus	Mean pool	Electrodes kept	Attention	EEGNetv4
BCI IV-2a	2,052	45,060	2,564	3,060
PhysioNet MI	2,565	48,645	3,077	3,413
Lee2019 ERP	1,026	19,458	1,538	2,210
Lee2019 SSVEP	2,052	38,916	2,564	3,012
ISRUC	2,565	15,365	3,077	3,205
TUAB	1,026	19,458	1,538	2,210

Let $\kappa_{s,d,h}$ denote the attention score for pretraining seed s, corpus d, and classifier seed h. The score for one pretraining run is

$$K_s = \frac{1}{6}\sum_{d=1}^{6}\left(\frac{1}{3}\sum_{h=1}^{3}\kappa_{s,d,h}\right). \tag{9}$$

For conditions with three pretraining seeds, we report $\bar{K} \pm s_K$, where

$$\bar{K} = \frac{1}{3}\sum_{s=0}^{2} K_s, \qquad s_K = \sqrt{\frac{1}{2}\sum_{s=0}^{2}(K_s - \bar{K})^2}. \tag{10}$$

Thus, the reported SD measures variation across pretraining seeds of the equal-weight six-corpus mean. It is not the SD across corpora, a standard error, or a confidence interval. For mean pooling and electrodes kept, the same aggregation applies without a classifier-seed average. Corpus-specific REVE-small entries report mean and sample SD across the three pretraining seeds for that corpus, after averaging classifier seeds where applicable. The three classifier initializations per encoder are not treated as independent pretraining runs. Random-encoder summaries use the same procedure across three encoder initializations.

For EEGNetv4, we compute a six-corpus mean separately for each supervised training seed, then report the mean and sample SD across the three supervised seeds where an SD is shown. In the task-family figure, motor imagery averages BCI IV-2a and PhysioNet MI within each seed; the other families contain one corpus each. Frozen-transfer error bars summarize the three pretraining seeds after averaging classifier seeds, while EEGNetv4 error bars summarize the three supervised training seeds. Conditions with one pretraining run have no estimated pretraining-seed SD.

For the REVE-small real–synthetic contrasts, we compute the synthetic-minus-real difference within each corresponding pretraining-run index, then report the mean and sample SD of the three differences. These SDs are computed from the differences themselves, rather than by adding or averaging the separate conditions' SDs. All contrasts use unrounded scores.

Where subject-bootstrap intervals are retained for supervised EEGNetv4 or single-run controls, they are 95% percentile intervals from 2,000 resamples of whole test subjects, paired across conditions. Models and predictions remain fixed, and classifier-seed scores are averaged within each resample where applicable. These intervals do not measure variability across pretraining runs. The single-run REVE-base contrast uses bootstrap seed 20260920, with separate random streams for the six corpora and shared subject draws across conditions within each corpus. The two Lee2019 tasks are resampled separately. These intervals condition on the fitted models, readouts, subject partitions, and six retained corpora; they do not estimate uncertainty over a population of corpora.

D Construction and validation of the synthetic signals

The transformations act separately on each preprocessed window, before masking. For pretraining, each window has 19 channels and $T = 800$ samples at 200 Hz; the supervised controls use the channels available in each downstream corpus. Let $x_c(t)$ be channel c, $X_c(k)$ its discrete Fourier transform, and $\widetilde{x}_c(t)$ the transformed signal. A fresh draw is generated at every training presentation. The visible input and its target always come from the same draw. No filtering or normalization is applied after the transformation.

D.1 PSD + CROSS-SPECTRA

For every positive Fourier frequency except Nyquist, we draw $\phi_k \sim \mathcal{U}[0, 2\pi)$ and apply the same phase rotation to all channels:

$$\widetilde{X}_c(k) = X_c(k)e^{i\phi_k}. \tag{11}$$

The DC and Nyquist components remain unchanged; negative frequencies follow conjugate symmetry, giving a real-valued inverse transform. For every channel pair,

$$\widetilde{X}_c(k)\widetilde{X}_d(k)^* = X_c(k)X_d(k)^*. \tag{12}$$

The transformation therefore preserves the whole-window power spectra and complex cross-spectra, including relative phases between channels. It also preserves means, variances, and zero-lag covariance, up to numerical precision.

These identities concern the full four-second window. They do not preserve the original waveform, event timing, or spectra estimated within shorter intervals. Phase randomization changes how frequency components combine in time. It does not imply that every localized fluctuation disappears, that energy becomes exactly uniform in time in each draw, or that all higher-order dependencies are removed. Because Fourier amplitudes remain fixed at their source-window values, the transformed signals are not exact Gaussian-process draws. Their preserved cross-spectra also contain more detail than the finite descriptor set in Appendix A.3.

D.2 PSD

Here the phases $\phi_{c,k}$ are drawn independently for every channel and frequency. Whole-window channel power spectra, means, and variances remain unchanged. Cross-channel phase relationships generally change, so the original complex cross-spectra and off-diagonal covariance are no longer preserved. This does not remove every cross-channel statistic: for example, the magnitude $|X_c(k)X_d(k)^*|$ remains fixed by the two preserved amplitudes. DC and Nyquist components are again unchanged.

D.3 COVARIANCE

Let μ be the vector of channel means and $\Sigma = (x-\mu)(x-\mu)^\top/T$ the window covariance. With L the Cholesky factor of $\Sigma + \epsilon I$, where $\epsilon = 10^{-3}\,\mathrm{tr}(\Sigma)/C + 10^{-6}$, we draw

$$\widetilde{x}(t) = \mu + L\eta(t), \qquad \eta(t) \sim \mathcal{N}(0, I), \tag{13}$$

independently at each time point. The output is temporally white and matches the regularized covariance in expectation. Finite windows do not reproduce it exactly. In particular, this control does not preserve the original band-limited spectrum.

D.4 MEAN + VARIANCE

We draw $\widetilde{x}_c(t) = \mu_c + s_c\eta_c(t)$ independently across channels and time, where s_c is the source window's sample standard deviation, using denominator $T-1$. The target means and variances are matched in expectation, without readjustment after sampling. Since the input is already standardized per channel, this condition is close to independent standardized white noise.

D.5 NUMERICAL PRESERVATION CHECKS

We apply the pretraining transformations to 128 archived windows from each of three evaluation sets (TUEG, Stieger, and a mixed set), all with 19 channels and 800 samples. For statistic s, the relative error is $\|s(\widetilde{x}) - s(x)\|_F / \|s(x)\|_F$. Table 17 first pools squared norms over the windows of each set, then averages the three set-level errors. Figure 6 also shows the per-window ratios; a pooled ratio need not equal their mean. Distances between different real windows provide a descriptive reference.

The set-level errors for exactly preserved Fourier statistics are below 4×10^{-7}. The covariance control has relative covariance error 0.069, and the Gaussian variance errors are approximately 0.050, reflecting finite-sample variation and, for covariance, regularization. Independent phases substantially

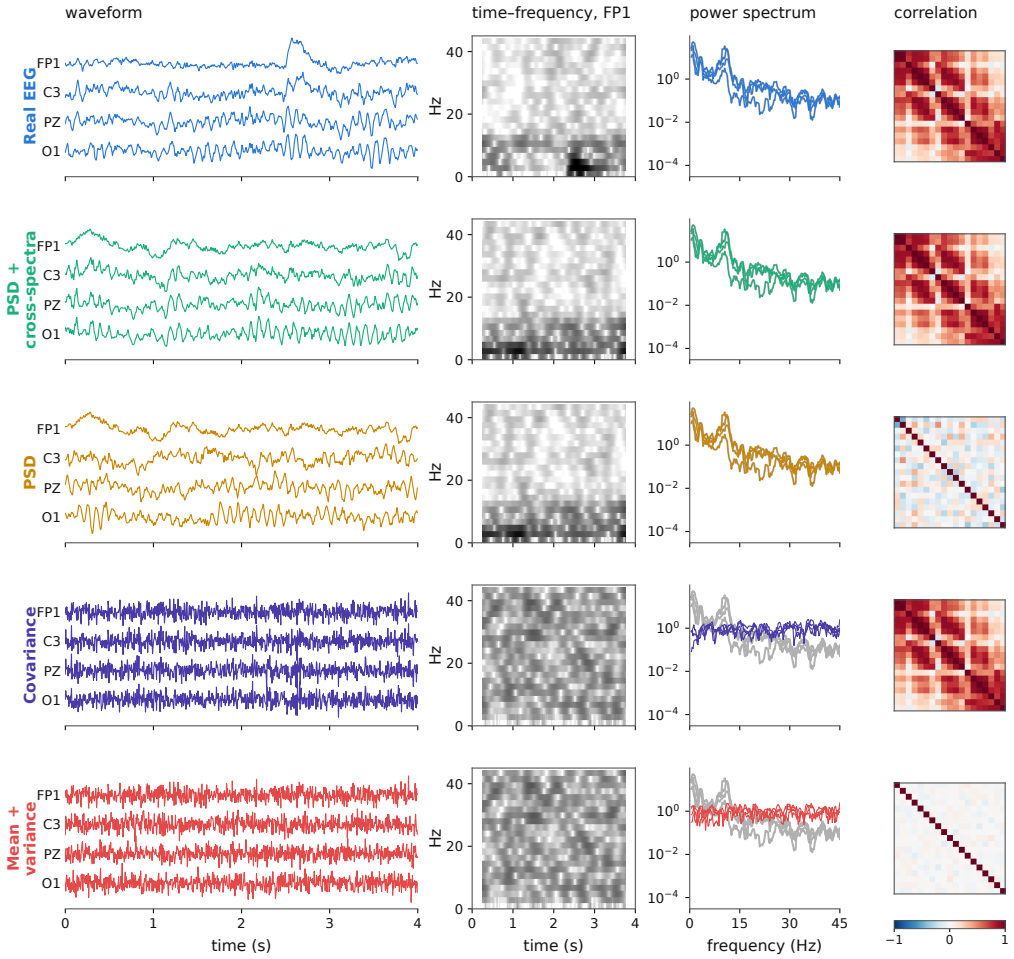


Figure 4: **One real TUEG window and its four synthetic versions.** Rows show the real window and one draw of each transformation. Columns show four channels, time–frequency power on FP1 (0.5-second segments), smoothed whole-window power spectra (real reference in grey), and the 19-channel correlation matrix. The localized frontal transient is altered by phase randomization while the whole-window spectra remain unchanged. Shared phases also preserve the correlation matrix. Gaussian controls target the full covariance or only the channel variances, in expectation.

Table 17: Numerical preservation checks. Entries are relative Frobenius errors pooled within each evaluation set and averaged over the three sets. Fourier products use the complete four-second window; Welch uses one-second segments. The last row compares two different real windows from the same set.

Signal version	Power spec.	Cross-spec.	Covariance	Variances	Welch 1 s	Waveform
PSD + cross-spectra	1.4×10^{-7}	1.3×10^{-7}	3.5×10^{-7}	5.8×10^{-8}	0.246	1.425
PSD	1.4×10^{-7}	1.359	0.975	5.9×10^{-8}	0.250	1.414
Covariance	1.000	1.011	0.069	0.050	0.984	1.415
Mean + variance	1.000	1.015	0.926	0.050	0.983	1.415
Two real windows	1.059	1.288	0.779	0.013	0.813	1.408

change the complex cross-spectra and zero-lag covariance; Gaussian inputs substantially change the spectra. Waveform distances are approximately 1.41 in all four conditions.

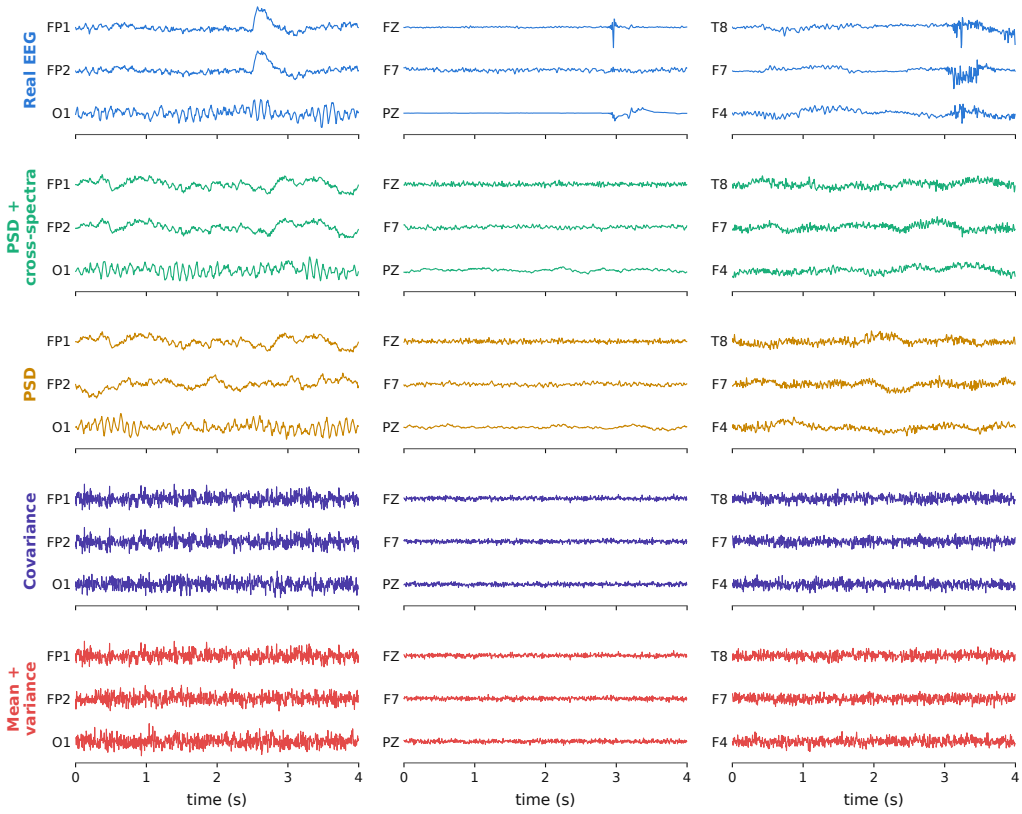


Figure 5: **Examples of changes in waveform and temporal localization.** Columns show a frontal blink-like transient, a sharp transient on FZ (both TUEG), and a burst of high-frequency activity on T8 and F7 (mixed evaluation set). Three channels are displayed per window. Rows show the real signal and one draw of each synthetic version, using the colours of Figure 4. These examples illustrate altered event morphology and timing; they are not a quantitative test that every event or physiological feature is removed.

One-second Welch spectra are not exactly preserved by either phase transformation (relative errors 0.246 and 0.250). The identity applies to the full-window Fourier products, while Welch estimation uses shorter, tapered segments. Similarly, filtering a finite window can introduce boundary effects in band-limited descriptors.

E Additional results for Section 3

E.1 Results by corpus and readout

Tables 18–20 give the signal conditions, individual initialization references A and B, and extended-budget checkpoints for the six corpora. At the standard budget, the Real EEG, PSD + cross-spectra, Covariance, and PSD MAE entries summarize three REVE-small pretraining seeds. The raw-linear reference applies the same ridge procedure to the flattened real window, without an encoder. EEG-Netv4 entries use the signal version indicated by the row for both training and evaluation. All MAE entries use real EEG for evaluation. All frozen-readout results use the protocol of Appendix C.5. Appendix E.2 compares the pretraining objectives and the larger REVE encoder.

Changing the readout changes absolute performance, but the average comparison between real EEG and PSD + cross-spectra remains close. At the standard budget, their mean-pooling scores are 0.312 and 0.311; with electrodes kept, they are 0.415 and 0.406; with attention, they are 0.347 and 0.349. Each score averages three pretraining seeds and weights the six corpora equally. Electrode-wise

Table 18: Motor imagery. Reconstruction is pooled R_0^2 on real EEG; the three frozen-readout columns report κ on real EEG. For the four standard-budget conditions Real EEG, PSD + cross-spectra, Covariance, and PSD, these entries give mean $\pm$ sample SD across three REVE-small pretraining seeds. Attention first averages three classifier seeds within each pretraining run and corpus. Mean + variance, each individual initialization reference, and each $5\times$ condition use one encoder run and are reported as point estimates. EEGNetv4 entries remain means across three supervised training seeds on the row's signal version. Dashes denote inapplicable evaluations, not zero scores.

Condition	R_0^2	Mean pool	Electrodes kept	Attention	EEGNetv4
BCI IV-2a					
Real EEG	0.890 $\pm$0.001	0.155 $\pm$0.008	0.305 $\pm$0.009	0.111 $\pm$0.012	0.440
PSD + cross-spectra	0.890 $\pm$0.001	0.160 $\pm$0.012	0.296 $\pm$0.011	0.134 $\pm$0.023	0.168
Covariance	0.874 $\pm$0.004	0.139 $\pm$0.011	0.244 $\pm$0.004	0.144 $\pm$0.006	0.089
PSD	0.058 $\pm$0.000	0.094 $\pm$0.006	0.242 $\pm$0.008	0.082 $\pm$0.006	0.037
Mean + variance	0.001	0.036	0.126	0.063	0.000
Initialization A	-0.315	0.111	0.181	0.111	—
Initialization B	-0.325	0.105	0.199	0.105	—
Real EEG, $5\times$	0.887	0.169	0.320	0.129	—
PSD + cross-spectra, $5\times$	0.888	0.171	0.298	0.160	—
Raw linear	—	0.270 (raw-window ridge)			—
PhysioNet MI					
Real EEG	0.715 $\pm$0.001	0.162 $\pm$0.031	0.267 $\pm$0.016	0.124 $\pm$0.002	0.395
PSD + cross-spectra	0.707 $\pm$0.000	0.135 $\pm$0.012	0.240 $\pm$0.007	0.115 $\pm$0.003	0.075
Covariance	0.627 $\pm$0.003	0.151 $\pm$0.014	0.221 $\pm$0.017	0.095 $\pm$0.006	0.075
PSD	0.062 $\pm$0.001	0.105 $\pm$0.020	0.267 $\pm$0.009	0.113 $\pm$0.003	0.040
Mean + variance	0.000	0.037	0.147	0.042	0.011
Initialization A	-0.319	0.135	0.207	0.032	—
Initialization B	-0.344	0.131	0.214	0.035	—
Real EEG, $5\times$	0.721	0.148	0.273	0.135	—
PSD + cross-spectra, $5\times$	0.712	0.187	0.242	0.127	—
Raw linear	—	0.272 (raw-window ridge)			—

Table 19: ERP and SSVEP. Reconstruction is pooled R_0^2 on real EEG; the three frozen-readout columns report κ on real EEG. For the four standard-budget conditions Real EEG, PSD + cross-spectra, Covariance, and PSD, these entries give mean $\pm$ sample SD across three REVE-small pretraining seeds. Attention first averages three classifier seeds within each pretraining run and corpus. Mean + variance, each individual initialization reference, and each $5\times$ condition use one encoder run and are reported as point estimates. EEGNetv4 entries remain means across three supervised training seeds on the row's signal version. Dashes denote inapplicable evaluations, not zero scores.

Condition	R_0^2	Mean pool	Electrodes kept	Attention	EEGNetv4
Lee2019 ERP					
Real EEG	0.726 $\pm$0.000	0.003 $\pm$0.010	0.037 $\pm$0.017	-0.013 $\pm$0.006	0.341
PSD + cross-spectra	0.712 $\pm$0.002	0.005 $\pm$0.009	0.032 $\pm$0.023	-0.005 $\pm$0.006	0.000
Covariance	0.608 $\pm$0.012	0.002 $\pm$0.013	0.073 $\pm$0.047	0.008 $\pm$0.014	0.003
PSD	0.061 $\pm$0.001	0.012 $\pm$0.003	0.092 $\pm$0.031	-0.003 $\pm$0.014	0.000
Mean + variance	0.000	0.000	0.068	0.020	0.000
Initialization A	-0.328	-0.002	0.081	0.000	—
Initialization B	-0.349	0.000	0.099	0.012	—
Real EEG, $5\times$	0.733	0.033	0.130	0.003	—
PSD + cross-spectra, $5\times$	0.717	-0.007	0.035	-0.011	—
Lee2019 SSVEP					
Real EEG	0.729 $\pm$0.000	0.416 $\pm$0.007	0.733 $\pm$0.005	0.724 $\pm$0.019	0.894
PSD + cross-spectra	0.714 $\pm$0.001	0.425 $\pm$0.015	0.723 $\pm$0.007	0.721 $\pm$0.010	0.882
Covariance	0.628 $\pm$0.008	0.181 $\pm$0.020	0.633 $\pm$0.005	0.652 $\pm$0.002	0.086
PSD	0.051 $\pm$0.001	0.336 $\pm$0.024	0.658 $\pm$0.005	0.580 $\pm$0.016	0.658
Mean + variance	0.000	0.111	0.624	0.386	0.000
Initialization A	-0.343	0.337	0.651	0.454	—
Initialization B	-0.361	0.312	0.666	0.473	—
Real EEG, $5\times$	0.737	0.475	0.764	0.765	—
PSD + cross-spectra, $5\times$	0.719	0.487	0.765	0.747	—

Table 20: Sleep and clinical EEG. Reconstruction is pooled R_0^2 on real EEG; the three frozen-readout columns report κ on real EEG. For the four standard-budget conditions Real EEG, PSD + cross-spectra, Covariance, and PSD, these entries give mean $\pm$ sample SD across three REVE-small pretraining seeds. Attention first averages three classifier seeds within each pretraining run and corpus. Mean + variance, each individual initialization reference, and each $5\times$ condition use one encoder run and are reported as point estimates. EEGNetv4 entries remain means across three supervised training seeds on the row's signal version. Dashes denote inapplicable evaluations, not zero scores.

Condition	R_0^2	Mean pool	Electrodes kept	Attention	EEGNetv4
ISRUC					
Real EEG	0.215 $\pm$0.014	0.538 $\pm$0.003	0.535 $\pm$0.005	0.541 $\pm$0.005	0.537
PSD + cross-spectra	0.207 $\pm$0.014	0.544 $\pm$0.006	0.546 $\pm$0.004	0.541 $\pm$0.005	0.554
Covariance	0.115 $\pm$0.017	0.274 $\pm$0.005	0.278 $\pm$0.005	0.278 $\pm$0.003	0.162
PSD	0.024 $\pm$0.003	0.473 $\pm$0.008	0.471 $\pm$0.006	0.460 $\pm$0.002	0.507
Mean + variance	0.000	0.100	0.090	0.167	0.003
Initialization A	-0.319	0.321	0.275	0.090	—
Initialization B	-0.342	0.307	0.273	0.045	—
Real EEG, $5\times$	0.192	0.547	0.545	0.561	—
PSD + cross-spectra, $5\times$	0.177	0.579	0.576	0.567	—
Raw linear	—	0.026 (raw-window ridge)			—
TUAB					
Real EEG	0.664 $\pm$0.002	0.597 $\pm$0.006	0.610 $\pm$0.008	0.593 $\pm$0.008	0.532
PSD + cross-spectra	0.659 $\pm$0.002	0.599 $\pm$0.004	0.597 $\pm$0.007	0.587 $\pm$0.007	0.561
Covariance	0.565 $\pm$0.004	0.392 $\pm$0.004	0.420 $\pm$0.013	0.396 $\pm$0.010	0.297
PSD	0.062 $\pm$0.000	0.477 $\pm$0.008	0.489 $\pm$0.008	0.469 $\pm$0.006	0.460
Mean + variance	0.000	0.199	0.085	0.247	-0.017
Initialization A	-0.322	0.331	0.191	0.191	—
Initialization B	-0.344	0.344	0.226	0.235	—
Real EEG, $5\times$	0.675	0.640	0.626	0.632	—
PSD + cross-spectra, $5\times$	0.669	0.620	0.635	0.615	—
Raw linear	—	-0.008 (raw-window ridge)			—

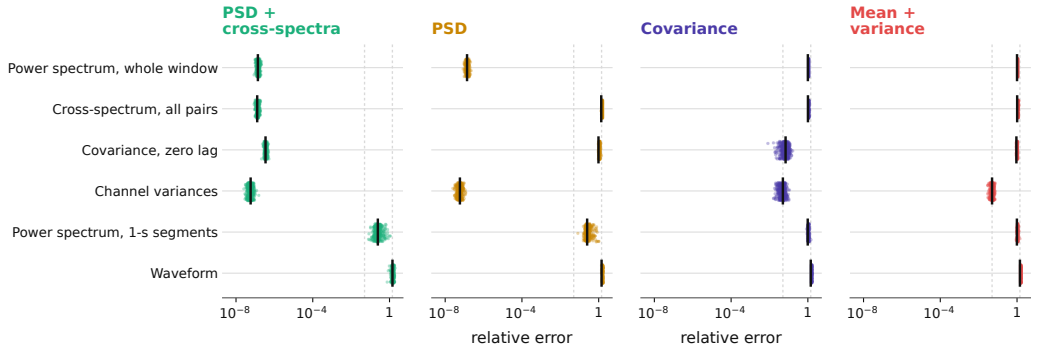


Figure 6: **Per-window preservation errors.** Each point compares a synthetic window with its source (128 windows from each of three sets). Black bars summarize the pooled errors. The reference $\sqrt{2/T} \approx 0.05$ gives the scale of relative sampling fluctuations of a Gaussian variance estimate; it is not a general threshold for covariance-matrix error. The reference $\sqrt{2}$ is the relative distance between orthogonal signals of equal energy. The small Fourier errors are consistent with the exact identities in Equation 12 evaluated at finite precision.

pooling improves the mean score while also increasing the number of classifier parameters. Its gain cannot be assigned solely to either the retained structure or the larger classifier from this comparison alone.

The random-encoder reference over three encoder initializations reaches 0.146 ± 0.008 in mean attention κ. Both real-EEG and PSD + cross-spectra pretraining remain well above this descriptive reference. The individual initialization references A and B are also reported by corpus; these reference comparisons do not assume identical starting weights across all signal conditions.

There are meaningful differences within this overall pattern. With electrodes kept, real EEG exceeds PSD + cross-spectra by 0.028 on PhysioNet MI, whereas the sign reverses under some other corpus–readout combinations. Mean-pooled real and PSD + cross-spectra features give near-zero scores on Lee2019 ERP; other readouts recover modest scores. These results limit what this probe demonstrates about ERP transfer. They do not establish that the embeddings contain no ERP information.

For supervised EEGNetv4, changing real EEG to PSD + cross-spectra reduces κ from 0.440 to 0.168 on BCI IV-2a, from 0.395 to 0.075 on PhysioNet MI, and from 0.341 to 0.000 on Lee2019 ERP. The corresponding real-minus-synthetic subject-bootstrap intervals are $[0.229, 0.316]$, $[0.271, 0.372]$, and $[0.266, 0.416]$. SSVEP changes only from 0.894 to 0.882; ISRUC and TUAB are slightly higher on the transformed signal. The supervised penalty is therefore concentrated in motor imagery and ERP under this protocol.

E.2 The signal comparison across pretraining objectives

Table 21 reports the four-signal grid for REVE-small under MAE, JEPA, and JEPA + reconstruction, with three pretraining seeds per condition, together with the single-run REVE-base MAE size control. The objective comparison retains the same REVE-small encoder architecture, while the size control retains the MAE objective. Both use the data and evaluation procedure of the main experiment (Appendix C.3). Table 22 isolates the effect of replacing real EEG with PSD + cross-spectra at each fixed encoder size, objective, and readout.

With attention pooling, real-EEG and PSD + cross-spectra pretraining give 0.347 and 0.349 under MAE, 0.354 and 0.349 under pure JEPA, and 0.352 and 0.349 under JEPA + reconstruction, averaging three pretraining seeds per condition. The real–synthetic proximity therefore persists when the target is a learned latent representation and no raw-sample reconstruction loss is used. It also persists when latent prediction and reconstruction are combined. Under REVE-base MAE, the corresponding scores are 0.368 and 0.364, extending the observation to a deeper encoder.

Across the three REVE-small objectives and all three readouts, the synthetic-minus-real differences in six-corpus mean κ range from -0.013 to $+0.002$ after averaging pretraining seeds (computed before rounding). Individual run differences range from -0.024 to $+0.010$. The REVE-base size control also gives small mean differences. These results support a close average comparison across the tested objectives and readouts, rather than dependence on masked raw-signal reconstruction alone; they are not a formal equivalence test.

Table 21: Objective and encoder-size controls. All transfer scores use real EEG and the readouts of Appendix C.5. Each run score averages six corpora equally; attention first averages three classifier seeds within each corpus and pretraining run. REVE-small entries give mean ± sample SD across three pretraining seeds. R_0^2 is the pooled reconstruction score within each corpus, averaged across corpora within each run. REVE-base entries are point estimates from one pretraining run. Dashes denote no evaluated trained reconstruction head. The REVE-small random reference summarizes three encoder initializations; the REVE-base random reference uses one.

Encoder / objective	Pretraining signal	R_0^2	Mean pool	Electrodes kept	Attention
REVE-small / MAE	Real EEG	0.657 ±0.003	0.312 ±0.005	0.415 ±0.002	0.347 ±0.005
	PSD + cross-spectra	0.648 ±0.003	0.311 ±0.007	0.406 ±0.003	0.349 ±0.005
	Covariance	0.570 ±0.008	0.190 ±0.005	0.311 ±0.008	0.262 ±0.006
	PSD	0.053 ±0.001	0.250 ±0.002	0.370 ±0.008	0.283 ±0.006
REVE-small / JEPA	Real EEG	—	0.323 ±0.002	0.423 ±0.010	0.354 ±0.002
	PSD + cross-spectra	—	0.311 ±0.007	0.413 ±0.007	0.349 ±0.003
	Covariance	—	0.199 ±0.003	0.312 ±0.006	0.266 ±0.004
	PSD	—	0.270 ±0.003	0.375 ±0.005	0.307 ±0.004
REVE-small / JEPA + recon.	Real EEG	0.655 ±0.001	0.317 ±0.010	0.417 ±0.001	0.352 ±0.004
	PSD + cross-spectra	0.647 ±0.000	0.319 ±0.006	0.412 ±0.003	0.349 ±0.003
	Covariance	0.574 ±0.006	0.194 ±0.007	0.314 ±0.008	0.270 ±0.005
	PSD	0.055 ±0.001	0.256 ±0.005	0.381 ±0.011	0.293 ±0.003
REVE-base / MAE	Real EEG	0.661	0.327	0.424	0.368
	PSD + cross-spectra	0.653	0.330	0.420	0.364
	Covariance	0.574	0.193	0.300	0.265
	PSD	0.053	0.252	0.369	0.286
Random encoder references					
REVE-small	—	—	0.203 ±0.005	0.266 ±0.003	0.146 ±0.008
REVE-base	—	—	0.200	0.297	0.181

The reduced-statistic controls give a consistent separation across objectives. For REVE-small, preserving covariance alone or channel spectra alone yields lower six-corpus mean transfer than real EEG in all three pretraining runs under every objective and readout. With attention, mean deficits relative to real EEG are 0.082–0.088 for Covariance and 0.047–0.063 for PSD. REVE-base shows the same ordering. Thus, the experiment shows both the proximity of real EEG to PSD + cross-spectra and the lower average transfer obtained when preserving only one of these statistical components.

For configurations with a reconstruction head, real-EEG and PSD + cross-spectra pretraining also give close pooled reconstruction scores on real downstream EEG: 0.657 versus 0.648 for REVE-small MAE, 0.655 versus 0.647 for JEPA + reconstruction, and 0.661 versus 0.653 for REVE-base MAE. The REVE-small values average three pretraining seeds. These comparisons use the same 75% mask and patch geometry. Pure JEPA is evaluated through transfer, without attributing a signal-reconstruction score to its latent predictor.

Table 22: PSD + cross-spectra minus real-EEG pretraining at fixed encoder size, objective, and readout. Differences use unrounded scores and average the six corpora equally. REVE-small entries report mean ± sample SD of the three differences computed within corresponding pretraining-run indices. For the single-run REVE-base control, brackets retain 95% intervals from 2,000 paired test-subject bootstrap resamples, with independent resampling across corpora; those intervals exclude pretraining-run uncertainty.

Encoder	Objective	Mean pool	Electrodes kept	Attention
REVE-small	MAE	-0.001 ± 0.005	-0.009 ± 0.005	$+0.002$ ± 0.008
REVE-small	JEPA	-0.013 ± 0.005	-0.010 ± 0.017	-0.005 ± 0.005
REVE-small	JEPA + recon.	$+0.001$ ± 0.013	-0.005 ± 0.002	-0.003 ± 0.004
REVE-base	MAE	$+0.003$ $[-0.008, +0.015]$	-0.004 $[-0.013, +0.005]$	-0.003 $[-0.012, +0.007]$

Results by corpus. Table 23 gives the attention scores underlying the real–synthetic comparison. For all four encoder–objective configurations, the absolute difference in mean score is below 0.025 on every corpus; REVE-small means average three pretraining seeds. Thus, the six-corpus means do not conceal a large mean attention-pooling gap on one task. Corpus differences can be larger with other readouts: for example, under JEPA the PSD + cross-spectra condition is 0.029 below real EEG on PhysioNet MI when electrodes are kept. The results support a close average comparison, with some dependence on corpus and readout.

Attention scores on Lee2019 ERP remain close to zero under all four configurations. Similarity in that regime provides little evidence about preserving useful ERP transfer, and does not imply that the embeddings contain no ERP information. By contrast, JEPA achieves substantial transfer on SSVEP, ISRUC, and TUAB under both real and PSD + cross-spectra pretraining. The agreement between signal conditions is therefore not simply an agreement between uninformative encoders on every task.

Table 23: Attention-pooling transfer by corpus. Synthetic denotes PSD + cross-spectra pretraining; all readouts are fitted and tested on real EEG. REVE-small entries report mean ± sample SD across three pretraining seeds, after averaging three classifier seeds within each pretraining run and corpus. REVE-base entries average three classifier seeds for one pretraining run. BCI and PhysioNet denote the two motor-imagery corpora; ERP and SSVEP denote the Lee2019 tasks.

Encoder / objective	Signal	BCI	PhysioNet	ERP	SSVEP	ISRUC	TUAB
REVE-small / MAE	Real	0.111 ± 0.012	0.124 ± 0.002	-0.013 ± 0.006	0.724 ± 0.019	0.541 ± 0.005	0.593 ± 0.008
	Synthetic	0.134 ± 0.023	0.115 ± 0.003	-0.005 ± 0.006	0.721 ± 0.010	0.541 ± 0.005	0.587 ± 0.007
REVE-small / JEPA	Real	0.089 ± 0.008	0.127 ± 0.010	-0.001 ± 0.007	0.725 ± 0.014	0.582 ± 0.009	0.604 ± 0.006
	Synthetic	0.099 ± 0.021	0.127 ± 0.007	-0.005 ± 0.011	0.708 ± 0.024	0.577 ± 0.004	0.590 ± 0.008
REVE-small / JEPA + recon.	Real	0.113 ± 0.027	0.125 ± 0.004	-0.003 ± 0.013	0.713 ± 0.008	0.557 ± 0.012	0.607 ± 0.007
	Synthetic	0.117 ± 0.017	0.122 ± 0.007	0.000 ± 0.008	0.725 ± 0.010	0.549 ± 0.006	0.584 ± 0.007
REVE-base / MAE	Real	0.134	0.135	0.004	0.753	0.552	0.627
	Synthetic	0.143	0.132	-0.005	0.743	0.569	0.603

Published-checkpoint reference. As an external reference point, we also evaluate the published REVE-base checkpoint under the same downstream protocol. It reaches a mean attention transfer of 0.347 (Table 24), close to the 0.368 of our real-EEG MAE and the 0.364 of our PSD + cross-spectra MAE at the same encoder size. Our controlled runs therefore fall in the same range of average downstream performance as the public model. Note, however, that the published checkpoint was

pretrained on different data with a different compute budget, so this comparison is a sanity check on absolute performance, not evidence of matched pretraining efficiency.

Table 24: Published REVE-base and the controlled REVE-base MAEs under the same frozen-readout protocol on real EEG. Values are mean κ across six corpora; attention averages three classifier seeds. Pretraining data and compute are not matched to the published checkpoint.

REVE-base pretraining	Mean pool	Electrodes kept	Attention
Published checkpoint	0.299	0.418	0.347
Real EEG	0.327	0.424	0.368
PSD + cross-spectra	0.330	0.420	0.364

E.3 Five times the pretraining budget

Table 25: Effect of five times the pretraining budget. Reconstruction uses pooled R_0^2 within each corpus; downstream values are equal-weight means across six corpora. Transfer columns report κ. Standard-budget entries report mean $\pm$ sample SD across three REVE-small pretraining seeds, after averaging classifier seeds where applicable. Each $5\times$ condition uses one pretraining run and is reported as a point estimate. The unequal numbers of pretraining runs make this a descriptive budget comparison.

Condition	R_0^2 TUEG	R_0^2 Six corpora	Mean pool	Electrodes kept	Attention
Real EEG	0.683 $\pm$0.001	0.657 $\pm$0.003	0.312 $\pm$0.005	0.415 $\pm$0.002	0.347 $\pm$0.005
Real EEG, $5\times$	0.695	0.657	0.335	0.443	0.371
PSD + cross-spectra	0.671 $\pm$0.000	0.648 $\pm$0.003	0.311 $\pm$0.007	0.406 $\pm$0.003	0.349 $\pm$0.005
PSD + cross-spectra, $5\times$	0.681	0.647	0.340	0.425	0.368

The extended-budget models achieve modestly higher mean transfer (Table 25). With attention pooling, real EEG reaches 0.371 at the larger budget, compared with the standard-budget three-seed mean of 0.347; PSD + cross-spectra reaches 0.368, compared with 0.349. Their scores remain close. These observations support persistence of the real-versus-synthetic comparison at the larger budget; they do not support the claim that further pretraining never helps. Each extended-budget condition has one pretraining run, so its pretraining-seed variability is not estimated.

F Experimental setup for Section 4

The question: once a useful representation has been learned, what keeps it useful? A model can reconstruct an EEG signal accurately without making the target task easy to predict from its features. The experiments in Section 4 therefore ask a practical question: *if supervision first teaches an encoder a useful representation, can reconstruction alone preserve that benefit?* This matters because acquisition and retention are different requirements. A representation must first become useful, and it must remain useful as training continues.

The experiment follows this question directly. We first train with reconstruction and task labels together. We then make two identical copies of the resulting training state: one **keeps supervision**, while the other **removes supervision** and continues with reconstruction alone. A third model, trained with **reconstruction only from the beginning**, provides a reference at the same training budget. The two continuation branches see the same subjects and windows, in the same order, with the same reconstruction masks. New classifiers fitted to each frozen encoder on separate subjects measure what its representation can support at each checkpoint.

This design makes the comparison informative: it tests the retention of an already acquired predictive advantage, checks whether continued supervision helps preserve it, and measures whether either outcome comes with a substantial change in reconstruction loss. Each corpus is a separate experiment with a randomly initialized encoder.

F.1 Data and subject separation

We use Stieger 2021 for four-class motor imagery, Lee2019 ERP for binary event-related potential decoding, ISRUC for five-class sleep staging, and TUAB for normal versus abnormal EEG classification. These provide two event-related tasks and two state-related tasks. Signals and labels from different corpora are never mixed.

We use the benchmark preprocessing: 4-s windows at 200 Hz, standardized within each window and channel. The montage contains 19 electrodes for Stieger, ERP, and TUAB, and six for ISRUC. The fixed evaluation panel contains at most 6,000 windows per corpus (5,994 for ERP). A separate encoder-training stream is drawn from the corpus's labeled pool, capped at 60,000 windows before restricting it to encoder-training subjects.

Subjects are partitioned once, with seed 20260908, into three disjoint groups:

- **Encoder training (A, approximately 40%):** all data used to update the encoder, whether or not labels are used.
- **Readout development (B, 30%):** data used to fit and select the classifiers that evaluate frozen representations. These subjects are further divided into fitting and validation groups in an approximately 2:1 ratio.
- **Testing (C, 30%):** subjects held out for evaluation, never used to fit or select a model.

All sessions of a subject remain in the same group. TUAB splits are stratified by each subject's majority label in the evaluation panel. Table 26 gives the realized counts. No window from B or C enters encoder training, even without its label. Consequently, the readouts test whether the learned representation is useful for new subjects. The task classes are shared across groups; transfer to new tasks or classes is not tested.

Table 26: Subject groups and evaluation windows. Encoder-training subjects (A), readout-development subjects (B), and test subjects (C) are disjoint. B is divided into fitting and validation subjects. Window counts describe the fixed evaluation panel, not the encoder-training stream.

Corpus	Classes	Training subjects (A)	Readout subjects (B) fit / val.	Test subjects (C)	Readout windows fit / val.	Test windows
Stieger 2021	4	25	13 / 6	18	1160 / 703	1801
Lee2019 ERP	2	22	11 / 5	16	1221 / 555	1776
ISRUC	5	39	19 / 10	30	1166 / 631	1818
TUAB	2	930	465 / 233	698	1200 / 571	1803

The finite A stream contains roughly 24,000 windows per corpus, which are repeatedly shuffled and presented during training. An implementation "epoch" contains 4,096,000 presentations and therefore includes many passes through this finite set. Presentation counts measure exposure, not distinct examples. Incomplete batches at the end of each shuffle are dropped, and a new permutation is drawn.

F.2 Training: learn with supervision, then keep or remove it

Architecture and initialization. We use the small REVE-style masked autoencoder: embedding width 512, four encoder transformer layers, eight attention heads, and a two-layer decoder. Patches contain 200 samples with 20-sample overlap, giving four temporal patches per electrode. Random masking removes 75% of electrode–patch tokens. Each independent encoder training run begins from random weights. Within each corpus and encoder seed, the reconstruction-only model and the model trained with supervision share exactly the same initialization. Encoder seeds 0, 1, and 2 correspond to random seeds.

What the model is trained to do. The reconstruction objective asks the model to recover masked signal patches. The supervised objective asks its features to predict the task label. Let f_θ and g_ϕ denote the encoder and reconstruction decoder, and let m denote the reconstruction mask. The reconstruction loss L_{recon} is the mean absolute error on masked patches, using the window-standardized

signal as the target, without additional target standardization within each patch. A linear classifier q_ψ predicts the label from the mean of the encoder tokens of the complete window:

$$L_{\text{label}}(x, y) = \text{CE}\left(q_\psi\left(\frac{1}{L}\sum_{\ell=1}^{L} f_\theta(x)_\ell\right), y\right), \tag{14}$$

$$L_{\text{joint}}(x, y, m) = L_{\text{recon}}(x, m) + \lambda L_{\text{label}}(x, y). \tag{15}$$

Both terms use the same batch of A windows: a masked view for reconstruction and a complete view for classification. The main setting is $\lambda = 0.02$; a control uses stronger supervision, $\lambda = 0.1$, on encoder seed 0. The encoder and decoder are optimized together. The training-time classifier is never used to score frozen representations.

Step 1: establish what supervision adds. We train two models from the shared random initialization: one with reconstruction only, and one with reconstruction plus task supervision. At the end of this first phase, denoted t_{split}, their frozen representations are compared on held-out subjects. This establishes whether supervision has produced a predictive advantage worth tracking. We call the supervised model at this point the **supervised start**: it is the shared starting checkpoint for the next step.

Step 2: test whether reconstruction preserves that advantage. We copy the complete supervised start into two continuation branches:

- **Supervision kept:** continue training with reconstruction and task supervision together.
- **Supervision removed:** set the supervision weight to zero and continue with reconstruction alone.

Both branches inherit the same encoder, decoder, classifier, optimizer moments and step counters, training-loop state, and random-number state. The unused classifier remains present when supervision is removed, but contributes no supervised gradient. The **reconstruction-only reference** continues uninterrupted to the same final exposure, denoted t_{end}.

The two reconstruction-only continuations have different histories: the reference has never used labels, whereas the supervision-removed branch starts from a representation that has already learned from labels. This distinction is central to the retention question.

Optimizer controls: repeat the comparison with a fresh optimizer. To assess the role of optimizer history, we repeat both continuations from the same supervised start, clearing optimizer moments and step counters in both. We also continue the reconstruction-only reference from its first-phase checkpoint with a fresh optimizer. These conditions keep the same descriptive names; the optimizer setting is listed separately in Table 27.

The main reconstruction-only, supervised-learning, supervision-kept, and supervision-removed conditions are run on all three encoder seeds for every corpus. The three optimizer-reset controls use encoder seed 0 and $\lambda = 0.02$. The stronger-supervision control repeats supervised learning and both continuations on seed 0, using the same reconstruction-only reference.

Optimization and training budget. Training uses AdamW with constant learning rate 10^{-3}, weight decay 0.05, $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, batch size 2,048, and bfloat16 mixed precision. The first phase lasts 2,000 updates, or $t_{\text{split}} = 4,096,000$ window presentations. Continuation lasts 4,000 additional updates, or 8,192,000 presentations, giving a final cumulative budget of $t_{\text{end}} = 12,288,000$. These budgets are fixed independently of test performance.

Checkpoints are saved at first-phase steps 16, 64, 250, 500, 1,000, and 2,000, and at 16, 64, 250, 500, 1,000, 2,000, 3,000, and 4,000 updates after the split. The initial state and continuation starting states are also saved. In the tables, *end of first phase* and *end of training* refer to t_{split} and t_{end}, respectively.

What is matched between the continuations. A private generator determines data order from encoder seed and stream epoch. Mask and position-noise generators are re-seeded before each update

Table 27: Training conditions in plain language. "Supervised start" is the shared checkpoint at the end of the first phase. Reconstruction continues in every condition. The main retention comparison is *supervision kept* versus *supervision removed*, with the complete optimizer state copied into both branches. Reset controls change the optimizer setting in both branches.

Condition	Starting point	Training objective	Optimizer
Reconstruction only (reference)	Random initialization	Reconstruction	Continuous
Supervised learning (first phase)	Random initialization	Reconstruction and labels	Continuous
Supervision kept	Supervised start	Reconstruction and labels	Copied
Supervision removed	Supervised start	Reconstruction	Copied
Controls with a fresh optimizer			
Supervision kept	Supervised start	Reconstruction and labels	Reset
Supervision removed	Supervised start	Reconstruction	Reset
Reconstruction only	Reference at end of first phase	Reconstruction	Reset

using encoder seed and cumulative update number. The split occurs at a stream-epoch boundary, and resumed branches start at the corresponding next epoch. Thus, the additional supervised forward pass cannot alter later reconstruction masks or data order. The implementation checks mask identity and batch fingerprints. Saved readout scores and reconstruction losses at the start of each continuation equal those of the parent checkpoint.

The branches are matched in data exposure and update count. Their computation differs: training with labels adds a forward and backward pass through the encoder on complete windows. A token-layer count estimates approximately 2.3 times the work per update compared with reconstruction alone; this is not a measured FLOP count. No control matching that additional computation through extra reconstruction views was run.

F.3 Evaluation: can a new classifier use the frozen representation?

At every checkpoint, we freeze the encoder and fit a new classifier, or *readout*, on group B. This classifier is independent of the one used during encoder training. It tests whether the current representation remains useful even when a new classifier is allowed to adapt to it. Consequently, a decline cannot be explained solely by the original classification head becoming outdated.

Encoder features, denoted Z, are extracted from complete, unmasked windows. All feature standardization, dimensionality reduction, and classifier selection use only the designated B fitting and validation data; C is used only for evaluation. We evaluate both the encoder features alone and their combination with classical EEG descriptors, denoted H. Giving the same descriptors to both competing readouts tests whether the encoder comparison still matters when these established features are available.

Primary attention readout. Token features are standardized per embedding dimension over fitting windows and token positions. One learned query q pools the standardized tokens z_ℓ:

$$\alpha_\ell = \frac{\exp(q^\top z_\ell/\sqrt{512})}{\sum_j \exp(q^\top z_j/\sqrt{512})}, \qquad z_{\mathrm{pool}} = \sum_\ell \alpha_\ell z_\ell. \tag{16}$$

The query is initialized with independent $\mathcal{N}(0, 1/512)$ components. A linear classifier receives either z_{pool} alone or its concatenation with the classical descriptor block. Query and classifier are trained by cross-entropy using Adam, learning rate 5×10^{-4}, weight decay 10^{-3}, batches of 128, and at most 80 epochs. Validation pooled κ selects the checkpoint, with patience 10 and an improvement threshold of 10^{-4}. The best fitting-subject model is retained without refitting on validation subjects. Three readout initializations use seeds 20260908, 20260909, and 20260910.

Classical descriptors: a compact block and the full block. H concatenates multi-resolution log-power spectra, band-specific log-covariance descriptors, and coherency descriptors from the benchmark. Their respective widths are 5,966, 7,980, and 2,052 on the 19-electrode corpora (15,998 total), and 1,884, 882, and 180 on ISRUC (2,946 total). Each group is standardized on fitting windows; near-constant columns are set to zero. Dividing each group by the square root of three times its number of active columns gives equal total variance to the three groups. The concatenation is scaled by $\sqrt{d_Z}$, where d_Z is the readout's encoder-feature width.

The primary analysis uses H_{512}, the leading 512 principal directions of this group-balanced block, fitted on the readout-fitting windows. Principal-component scores retain their relative variance and are not whitened or independently re-standardized. We also evaluate full-width H: an advantage measured with a compact descriptor block need not persist when all descriptors are supplied. The descriptors alone are evaluated under the same classifier recipe, using an all-zero placeholder token in place of encoder features.

Alternative ways to pool encoder features. We evaluate two fixed alternatives to attention: global mean pooling ($d_Z = 512$), and temporal mean pooling followed by concatenation of electrodes ($d_Z = 512C_{\text{electrodes}}$). The latter retains a separate feature block for each electrode.

Their linear classifiers minimize cross-entropy plus an explicit squared-weight penalty, using full-batch Adam with learning rate 10^{-2} for at most 300 steps. Penalties in $\{10^{-3}, 10^{-2}, 10^{-1}\}$ and stopping steps are selected by validation κ, evaluated every five updates. Ties between penalty values favor the larger penalty. Each selected classifier is refitted on all of B for the selected number of steps, including refitting feature transformations on B. Three classifier initializations use seeds 0, 1, and 2. Global mean pooling is evaluated with both full H and H_{512}; electrode-preserving pooling uses full H.

F.4 What each comparison tells us

Overall performance and paired subject comparisons. We report two complementary summaries. The **pooled score**, κ_{pool}, is Cohen's κ over all test windows in C. It is computed separately for each of the three fitted readout heads and then averaged; predictions are not ensembled. A **paired subject comparison** first computes the score difference within each eligible subject and then averages these differences, giving each eligible subject equal weight.

For the latter, let $s_c^b(t; V)$ be the mean readout-head κ within test subject c, for condition b, checkpoint t, and readout input V. The eligible set $\mathcal{C}_*$ contains subjects with at least two test windows and at least two observed target classes. It includes all 18 Stieger, 16 ERP, and 30 ISRUC test subjects, but only 18 of the 698 TUAB test subjects. A paired subject comparison therefore need not equal the difference of two pooled scores. TUAB's paired estimates concern only this small subset with both labels and cannot provide uncertainty intervals for its full test population.

In the equations below, the superscripts recon, start, kept, and $\mathrm{removed}$ refer to the reconstruction-only reference, the supervised start, and the two continuation conditions. The supervised start is always evaluated at t_{split}; its time argument is omitted.

1. Initial gain: did supervision make the representation more useful? At the end of the first phase, we compare the supervised start with the reconstruction-only reference using encoder features alone:

$$\Delta_Z = \frac{1}{|\mathcal{C}_*|} \sum_{c \in \mathcal{C}_*} \left[s_c^{\mathrm{start}}(Z) - s_c^{\mathrm{recon}}(t_{\text{split}}; Z) \right]. \tag{17}$$

A positive value establishes an acquired predictive advantage under the tested readout. This is the benefit whose subsequent retention we examine.

2. Gain with descriptors: does the encoder advantage remain when classical features are available? At any checkpoint, we give both readouts the same descriptor block and compare condition b with the reconstruction-only reference at the same training budget:

$$G_H^b(t) = \frac{1}{|\mathcal{C}_*|} \sum_{c \in \mathcal{C}_*} \left[s_c^b(t; [H_{512}, Z]) - s_c^{\mathrm{recon}}(t; [H_{512}, Z]) \right]. \tag{18}$$

Full-descriptor controls replace H_{512} by H. This *gain with descriptors* compares two encoders while holding the supplied descriptors fixed. Improvement over the descriptors alone is a separate quantity, defined using pooled scores:

$$A^b_{H,\mathrm{pool}}(t) = \kappa_{\mathrm{pool}}([H, Z_b(t)]) - \kappa_{\mathrm{pool}}(H). \tag{19}$$

The same definition applies to the compact block by replacing H with H_{512}.

3. Change after removal: does performance actually fall? We compare the supervision-removed branch with its own starting representation:

$$\mathrm{Change}_Z(t) = \frac{1}{|\mathcal{C}_*|} \sum_{c \in \mathcal{C}_*} \left[s_c^{\mathrm{removed}}(t; Z) - s_c^{\mathrm{start}}(Z) \right]. \tag{20}$$

A negative value means that predictive performance has declined since the supervised start.

4. Benefit of keeping supervision: how far apart do the continuations become? We compare the two branches at the same checkpoint:

$$\mathrm{Gap}_Z(t) = \frac{1}{|\mathcal{C}_*|} \sum_{c \in \mathcal{C}_*} \left[s_c^{\mathrm{kept}}(t; Z) - s_c^{\mathrm{removed}}(t; Z) \right]. \tag{21}$$

A positive gap favors keeping supervision. The change and the gap answer different questions: a gap could arise because one branch improves while the other stays stable. Reporting the change from the shared starting point is therefore necessary to establish an actual decline. The optimizer-reset comparison uses the same gap definition, with both branches reset.

How much of the initial advantage remains? When the initial gain is substantial, we also report the fraction of the initial pooled advantage that remains above the reconstruction-only reference:

$$F_b(t) = \frac{\kappa_{\mathrm{pool}}(Z_b(t)) - \kappa_{\mathrm{pool}}(Z_{\mathrm{recon}}(t))}{\kappa_{\mathrm{pool}}(Z_{\mathrm{start}}) - \kappa_{\mathrm{pool}}(Z_{\mathrm{recon}}(t_{\mathrm{split}}))}. \tag{22}$$

Both numerator and denominator use pooled scores. This is a fraction of the measured predictive advantage, not a fraction of all task information in the encoder. It can exceed 100% if the advantage over the reference grows. We interpret it only for Stieger and ERP, where the denominator is clearly positive.

Uncertainty. Within an encoder seed, 95% intervals use 2,000 paired bootstrap resamples of the eligible test subjects, averaging the per-subject differences. These intervals condition on the trained encoder and fitted readouts. Encoder-training variability is reported separately as individual seed results or mean $\pm$ sample standard deviation across the three encoder seeds (divisor $3 - 1$). The three classifier initializations are not treated as additional encoder-training replications. Intervals are pointwise and do not include variability from alternative subject partitions.

Reconstruction quality: is the predictive benefit compatible with the reconstruction objective? We also evaluate reconstruction loss. This tests whether representations with different predictive performance can nevertheless reconstruct comparably well. The model's own masked-patch L1 loss is measured in evaluation mode on fixed banks drawn from A, B, and C. A shares subject identities with encoder training; B and C contain subjects unseen during encoder training. Bank sizes are given in Table 28.

Each bank is split into batches of at most 32 windows. Masks are fixed by batch index and seed across all conditions and checkpoints. The stored bank loss is the unweighted mean of these batch losses, including a final smaller batch when present. Each batch contributes one loss value, called a *loss cell* below.

Let $\overline{L}^b_B(t)$ be the stored reconstruction loss on bank B for condition b. Its relative excess over the reconstruction-only reference at the same budget is

$$E^b_B(t) = \frac{\overline{L}^b_B(t)}{\overline{L}^{\mathrm{recon}}_B(t)} - 1. \tag{23}$$

Table 28: Reconstruction-bank sizes and numbers of evaluation batches (loss cells), in encoder-training / readout-development / test order (A / B / C). Each batch contains 32 windows except a possible final smaller batch.

Corpus	Windows (A / B / C)	Batches (A / B / C)
Stieger 2021	2048 / 1863 / 1801	64 / 59 / 57
Lee2019 ERP	2048 / 1776 / 1776	64 / 56 / 56
ISRUC	2048 / 1797 / 1818	64 / 57 / 57
TUAB	2048 / 1771 / 1803	64 / 56 / 57

Negative values mean better reconstruction than the reference. The 95% interval uses 2,000 paired bootstrap resamples of loss cells, recomputing the ratio in each resample. The pre-specified compatibility criterion requires the upper interval bound to be at most $+1\%$ on bank B. These intervals describe variability over the fixed evaluation batches; they are not subject-bootstrap intervals and do not establish population-wide equivalence. We apply the criterion to the supervised start and to both continuation branches at the end of training.

F.5 Main results: what is learned, and what remains?

Supervision produces a substantial advantage on the two event-related tasks. Tables 29 and 30 report the absolute attention-readout scores for every encoder seed. At the end of the first phase, mean pooled κ increases from 0.074 to 0.367 on Stieger and from 0.005 to 0.277 on ERP when supervision is added. The paired subject-level initial gains are $+0.267 \pm 0.008$ and $+0.259 \pm 0.015$. With H_{512} supplied to both classifiers, the gains remain $+0.203 \pm 0.004$ and $+0.258 \pm 0.018$ (Table 32). These gains show that the architecture can learn features that improve prediction on new subjects, including when the supplied descriptor block is already available. There is therefore a clear acquired benefit whose retention can be tested.

The advantage is smaller on ISRUC: pooled κ increases from 0.402 to 0.461, and the subject-level gain with descriptors is $+0.021 \pm 0.015$. TUAB shows little pooled improvement, from 0.497 to 0.507; its pooled score with H_{512} slightly decreases, from 0.533 to 0.526. Its paired estimates concern a small eligible subset and differ from these pooled comparisons (Table 31). The strongest evidence for a substantial acquired advantage is thus concentrated in Stieger and ERP.

Removing supervision erodes much of that advantage. By the end of training, Stieger's pooled scores are 0.291 ± 0.020 with supervision kept, 0.146 ± 0.019 with supervision removed, and 0.065 ± 0.001 for the reconstruction-only reference. ERP reaches 0.260 ± 0.013, 0.013 ± 0.009, and -0.003 ± 0.017, respectively.

Two comparisons establish what happened. First, after supervision is removed, the subject-level change from the supervised start is -0.206 ± 0.014 on Stieger and -0.256 ± 0.015 on ERP: performance actually falls. Second, keeping supervision produces a final advantage over removing it of $+0.138 \pm 0.022$ and $+0.240 \pm 0.016$. For every encoder seed on both tasks, the change interval lies below zero and the gap interval lies above zero (Table 31).

The same pattern appears when classical descriptors are available. The final gain over the reconstruction-only reference is $+0.028 \pm 0.010$ on Stieger and $+0.002 \pm 0.007$ on ERP after supervision is removed, compared with $+0.150 \pm 0.015$ and $+0.205 \pm 0.041$ when it is kept. Across seeds, removing supervision retains 22–32% of the initial pooled advantage on Stieger and 2–14% on ERP. Keeping it retains 70–82% and 79–115%, respectively.

Keeping supervision therefore helps preserve the learned advantage, but does not guarantee constant performance: on Stieger, this branch also declines from its starting score. Table 33 reports every scheduled continuation checkpoint for both tasks.

ISRUC shows a smaller subject-level decline after removal (-0.026 ± 0.003), while the gap between keeping and removing supervision varies across seeds ($+0.013 \pm 0.022$), with little difference remaining when descriptors are supplied. TUAB also lacks a consistent subject-level retention gap. Neither corpus provides an initial advantage with descriptors comparable to Stieger or ERP.

Table 29: Performance using encoder features alone (Z), with the attention readout and $\lambda = 0.02$. Each entry is pooled κ over all test windows, averaged over three independently fitted readout heads for one encoder seed. The first phase ends after 4.096 million presentations; training ends after 12.288 million. Both continuation branches begin from the supervised start. The reconstruction-only reference never uses labels.

Corpus	Seed	End of first phase		End of training		
		Reconstruction only	Supervised start	Reconstruction only	Supervision kept	Supervision removed
Stieger 2021	0	0.073	0.367	0.065	0.273	0.151
	1	0.073	0.375	0.065	0.312	0.161
	2	0.076	0.360	0.064	0.290	0.125
Lee2019 ERP	0	0.002	0.274	−0.006	0.263	0.003
	1	0.012	0.303	0.015	0.245	0.021
	2	0.001	0.253	−0.019	0.271	0.016
ISRUC	0	0.417	0.456	0.427	0.463	0.427
	1	0.393	0.455	0.413	0.443	0.432
	2	0.396	0.472	0.409	0.437	0.448
TUAB	0	0.492	0.509	0.501	0.516	0.483
	1	0.508	0.511	0.525	0.524	0.498
	2	0.491	0.499	0.520	0.495	0.504

Table 30: Performance using classical descriptors and encoder features together ($[H_{512}, Z]$), with the attention readout and $\lambda = 0.02$. Each entry is pooled κ over all test windows, averaged over three independently fitted readout heads for one encoder seed. The first phase ends after 4.096 million presentations; training ends after 12.288 million. The corresponding descriptor-only scores are 0.130 (Stieger), −0.031 (ERP), 0.544 (ISRUC), and 0.532 (TUAB).

Corpus	Seed	End of first phase		End of training		
		Reconstruction only	Supervised start	Reconstruction only	Supervision kept	Supervision removed
Stieger 2021	0	0.140	0.363	0.133	0.281	0.173
	1	0.153	0.370	0.135	0.315	0.159
	2	0.142	0.356	0.134	0.292	0.171
Lee2019 ERP	0	0.012	0.275	0.013	0.202	0.015
	1	0.001	0.295	0.017	0.211	0.007
	2	−0.018	0.239	−0.007	0.260	0.005
ISRUC	0	0.545	0.547	0.549	0.553	0.546
	1	0.527	0.552	0.534	0.552	0.553
	2	0.532	0.565	0.536	0.550	0.562
TUAB	0	0.530	0.524	0.538	0.523	0.527
	1	0.536	0.541	0.548	0.534	0.529
	2	0.533	0.512	0.536	0.543	0.543

Table 31: What supervision adds and what happens after its removal: paired subject-level comparisons with 95% bootstrap intervals, attention readout, and $\lambda = 0.02$. Initial gains compare the supervised start with the reconstruction-only reference. "Change after removal" compares the final supervision-removed score with its starting score; negative values indicate decline. "Kept − removed" compares the final continuation scores. All columns use Z alone except "with descriptors", which uses $[H_{512}, Z]$. These are averages of within-subject differences, not differences of pooled scores. TUAB intervals concern only 18 eligible subjects.

Corpus	Seed	End of first phase		End of training	
		Initial gain (Z alone)	Initial gain with descriptors	Change after removal	Kept − removed
Stieger 2021	0	$+0.264$ $[+0.193, +0.335]$	$+0.204$ $[+0.150, +0.261]$	-0.194 $[-0.246, -0.139]$	$+0.114$ $[+0.086, +0.143]$
	1	$+0.276$ $[+0.221, +0.333]$	$+0.205$ $[+0.156, +0.258]$	-0.203 $[-0.239, -0.167]$	$+0.144$ $[+0.107, +0.183]$
	2	$+0.260$ $[+0.190, +0.335]$	$+0.199$ $[+0.141, +0.257]$	-0.221 $[-0.277, -0.169]$	$+0.155$ $[+0.109, +0.203]$
Lee2019 ERP	0	$+0.262$ $[+0.196, +0.330]$	$+0.249$ $[+0.186, +0.311]$	-0.260 $[-0.328, -0.193]$	$+0.246$ $[+0.190, +0.300]$
	1	$+0.272$ $[+0.195, +0.346]$	$+0.279$ $[+0.185, +0.371]$	-0.269 $[-0.337, -0.193]$	$+0.222$ $[+0.167, +0.278]$
	2	$+0.243$ $[+0.191, +0.294]$	$+0.246$ $[+0.184, +0.309]$	-0.240 $[-0.284, -0.191]$	$+0.253$ $[+0.182, +0.323]$
ISRUC	0	$+0.041$ $[+0.024, +0.057]$	$+0.005$ $[-0.015, +0.026]$	-0.029 $[-0.043, -0.015]$	$+0.036$ $[+0.019, +0.053]$
	1	$+0.067$ $[+0.046, +0.088]$	$+0.024$ $[+0.005, +0.044]$	-0.025 $[-0.046, -0.003]$	$+0.012$ $[-0.006, +0.028]$
	2	$+0.076$ $[+0.051, +0.102]$	$+0.034$ $[+0.016, +0.052]$	-0.024 $[-0.041, -0.008]$	-0.008 $[-0.025, +0.012]$
TUAB	0	$+0.020$ $[-0.072, +0.114]$	$+0.008$ $[-0.084, +0.083]$	$+0.044$ $[-0.065, +0.176]$	-0.061 $[-0.191, +0.025]$
	1	-0.041 $[-0.127, +0.036]$	-0.047 $[-0.138, +0.028]$	$+0.019$ $[-0.034, +0.071]$	$+0.025$ $[-0.029, +0.085]$
	2	-0.078 $[-0.147, -0.006]$	-0.015 $[-0.075, +0.040]$	$+0.097$ $[+0.030, +0.170]$	-0.033 $[-0.103, +0.028]$

Table 32: Gain over the reconstruction-only reference when the same classical descriptors (H_{512}) are available to both readouts. The reference is evaluated at the same training budget as the condition shown. Values are paired subject-level gains, mean ± sample SD across three encoder seeds; $\lambda = 0.02$. This SD measures variability across encoder seeds and is separate from subject-bootstrap uncertainty. TUAB estimates concern only its 18 eligible subjects.

Corpus	Supervised start (end of first phase)	Supervision kept (end of training)	Supervision removed (end of training)
Stieger 2021	$+0.203 \pm 0.004$	$+0.150 \pm 0.015$	$+0.028 \pm 0.010$
Lee2019 ERP	$+0.258 \pm 0.018$	$+0.205 \pm 0.041$	$+0.002 \pm 0.007$
ISRUC	$+0.021 \pm 0.015$	$+0.012 \pm 0.007$	$+0.014 \pm 0.015$
TUAB	-0.018 ± 0.028	-0.005 ± 0.018	-0.004 ± 0.006

Table 33: How predictive performance evolves during continuation on the two event-related tasks. Values are pooled κ on encoder features alone, using the attention readout and $\lambda = 0.02$, reported as mean $\pm$ sample SD across three encoder seeds. At zero added updates, both continuation branches are exactly the supervised start. Added presentations count repeated window exposures after the first phase, not distinct windows or cumulative training exposure.

Corpus	Added updates	Added presentations (millions)	Reconstruction only	Supervision kept	Supervision removed
Stieger 2021	0	0.000	0.074 ± 0.002	0.367 ± 0.007	0.367 ± 0.007
	16	0.033	0.069 ± 0.013	0.370 ± 0.007	0.349 ± 0.005
	64	0.131	0.081 ± 0.008	0.363 ± 0.004	0.330 ± 0.004
	250	0.512	0.074 ± 0.006	0.364 ± 0.006	0.290 ± 0.004
	500	1.024	0.069 ± 0.007	0.346 ± 0.017	0.253 ± 0.023
	1000	2.048	0.076 ± 0.002	0.339 ± 0.018	0.214 ± 0.014
	2000	4.096	0.074 ± 0.009	0.319 ± 0.005	0.166 ± 0.016
	3000	6.144	0.069 ± 0.002	0.293 ± 0.026	0.151 ± 0.022
	4000	8.192	0.065 ± 0.001	0.291 ± 0.020	0.146 ± 0.019
Lee2019 ERP	0	0.000	0.005 ± 0.006	0.277 ± 0.025	0.277 ± 0.025
	16	0.033	0.000 ± 0.007	0.290 ± 0.001	0.267 ± 0.029
	64	0.131	0.004 ± 0.003	0.274 ± 0.032	0.240 ± 0.030
	250	0.512	0.003 ± 0.008	0.301 ± 0.021	0.094 ± 0.030
	500	1.024	-0.003 ± 0.006	0.294 ± 0.027	0.054 ± 0.017
	1000	2.048	-0.002 ± 0.002	0.278 ± 0.036	0.019 ± 0.010
	2000	4.096	0.009 ± 0.006	0.265 ± 0.019	0.004 ± 0.004
	3000	6.144	-0.004 ± 0.011	0.240 ± 0.031	0.009 ± 0.006
	4000	8.192	-0.003 ± 0.017	0.260 ± 0.013	0.013 ± 0.009

The useful representation can coexist with comparable reconstruction quality. All 12 comparisons of the supervised start with the reconstruction-only reference satisfy the bank-B compatibility criterion at $\lambda = 0.02$ (Table 34). At the end of training, 23 of the 24 continuation comparisons satisfy it. The exception is TUAB with supervision removed, seed 1: the excess is 0.84%, with an upper interval bound of 1.06%, narrowly above the threshold.

Every Stieger and ERP comparison satisfies the criterion. On these two tasks, both acquiring a large predictive advantage and retaining much of it are compatible with reconstruction loss within the chosen tolerance. This gives the retention result its practical relevance: losing the acquired advantage is not necessary to meet that reconstruction criterion. Comparable reconstruction loss can accompany markedly different predictive performance. Table 35 gives the corresponding excesses on A and C.

F.6 Controls: optimizer history, supervision strength, and readout choice

Does the result depend on inherited optimizer history? Repeating both continuations with a fresh optimizer preserves the direction of the event-task result (Table 36). The subject-level gap favoring supervision kept over supervision removed is $+0.129$ on Stieger and $+0.201$ on ERP, compared with $+0.114$ and $+0.246$ for the corresponding seed-0 pair with inherited optimizer state. The magnitudes differ, particularly for ERP, but both reset gaps have positive 95% intervals.

When supervision is removed, the endpoint pooled score is similar with a reset and an inherited optimizer: 0.149 versus 0.151 on Stieger, and 0.024 versus 0.003 on ERP. Resetting the reconstruction-only reference changes its endpoint pooled score by less than 0.017 in absolute value on every corpus. Thus, the decline occurs both when optimizer history is preserved and when it is cleared in both branches. The result is not specific to either optimizer setting.

What changes with stronger supervision? Figure 7 compares the full training trajectories for $\lambda = 0.02$ and $\lambda = 0.1$ on the same encoder seed, showing how supervision strength affects both the supervised start and the subsequent continuations.

Table 34: Relative reconstruction excess on held-out bank B, in percent, with paired 95% batch-bootstrap intervals. Each condition is compared with the reconstruction-only reference at the same training budget. Negative values mean lower L1 loss. At $\lambda = 0.02$, the upper bound is below the pre-specified $+1\%$ margin in all 12 first-phase comparisons and 23 of 24 continuation comparisons. The exception is TUAB with supervision removed, seed 1. Resampling units are fixed evaluation batches, not independent subjects.

Corpus	Seed	Supervised start (end of first phase)	Supervision kept (end of training)	Supervision removed (end of training)
Stieger 2021	0	$+0.34$ $[+0.20, +0.48]$	$+0.45$ $[+0.23, +0.68]$	$+0.23$ $[+0.03, +0.44]$
	1	$+0.74$ $[+0.61, +0.89]$	-0.01 $[-0.18, +0.18]$	$+0.23$ $[+0.04, +0.44]$
	2	$+0.60$ $[+0.46, +0.74]$	-0.02 $[-0.20, +0.17]$	$+0.12$ $[-0.07, +0.30]$
Lee2019 ERP	0	$+0.00$ $[-0.10, +0.11]$	$+0.29$ $[+0.14, +0.43]$	$+0.48$ $[+0.33, +0.63]$
	1	$+0.20$ $[+0.11, +0.30]$	-0.15 $[-0.33, +0.03]$	$+0.08$ $[-0.10, +0.28]$
	2	$+0.11$ $[+0.02, +0.20]$	-0.09 $[-0.23, +0.04]$	$+0.29$ $[+0.15, +0.42]$
ISRUC	0	-0.55 $[-0.84, -0.24]$	-0.94 $[-1.42, -0.53]$	-0.63 $[-1.06, -0.25]$
	1	-0.65 $[-0.93, -0.38]$	-0.78 $[-1.12, -0.43]$	-0.96 $[-1.29, -0.62]$
	2	-0.66 $[-0.94, -0.37]$	-0.45 $[-0.77, -0.13]$	-0.19 $[-0.55, +0.15]$
TUAB	0	$+0.39$ $[+0.25, +0.52]$	$+0.21$ $[+0.01, +0.40]$	$+0.26$ $[+0.07, +0.45]$
	1	$+0.83$ $[+0.69, +0.96]$	$+0.69$ $[+0.44, +0.93]$	$+0.84$ $[+0.60, +1.06]$
	2	$+0.46$ $[+0.30, +0.60]$	$+0.48$ $[+0.25, +0.71]$	$+0.57$ $[+0.32, +0.82]$

Table 35: Relative reconstruction excess on banks A and C, in percent, compared with the reconstruction-only reference at the same training budget. Values are mean $\pm$ sample SD across three encoder seeds, with $\lambda = 0.02$. The supervised start is evaluated at the end of the first phase; the two continuations are evaluated at the end of training. Bank A shares encoder-training subject identities; bank C contains test subjects.

Corpus	Supervised start		Supervision kept		Supervision removed	
	Training (A)	Test (C)	Training (A)	Test (C)	Training (A)	Test (C)
Stieger 2021	$+0.921 \pm 0.242$	$+0.490 \pm 0.071$	$+0.379 \pm 0.250$	-0.028 ± 0.447	$+0.212 \pm 0.069$	$+0.165 \pm 0.306$
Lee2019 ERP	$+0.314 \pm 0.055$	$+0.100 \pm 0.121$	$+0.361 \pm 0.148$	$+0.016 \pm 0.211$	$+0.639 \pm 0.157$	$+0.354 \pm 0.314$
ISRUC	$+0.708 \pm 0.235$	-0.552 ± 0.220	$+0.498 \pm 0.371$	-0.589 ± 0.428	$+0.458 \pm 0.568$	-0.454 ± 0.264
TUAB	$+0.633 \pm 0.155$	$+0.480 \pm 0.211$	$+0.529 \pm 0.205$	$+0.372 \pm 0.384$	$+0.589 \pm 0.237$	$+0.487 \pm 0.410$

Table 36: Optimizer-reset controls at the end of training, encoder seed 0 and $\lambda = 0.02$. All three conditions shown use a reset optimizer. The first three numeric columns give pooled attention-readout κ on Z. "Kept − removed" is the paired subject-level gap with its 95% interval. The final column gives the change in the reconstruction-only reference's pooled score caused by resetting its optimizer. TUAB's paired gap uses only 18 subjects.

Corpus	Supervision kept	Supervision removed	Reconstruction only	Kept − removed 95% CI	Reference: reset − inherited
Stieger 2021	0.288	0.149	0.082	$+0.129$ $[+0.098, +0.159]$	$+0.016$
Lee2019 ERP	0.248	0.024	-0.006	$+0.201$ $[+0.129, +0.273]$	$+0.000$
ISRUC	0.438	0.436	0.421	$+0.003$ $[-0.015, +0.022]$	-0.006
TUAB	0.514	0.492	0.487	-0.046 $[-0.101, +0.005]$	-0.014

With $\lambda = 0.1$, keeping supervision on Stieger gives a final pooled $\kappa = 0.329$, compared with 0.113 after removal, retaining 93% versus 17% of the initial advantage. ERP ends at 0.309 versus 0.058, retaining 101% versus 21%. The gains with descriptors also remain larger when supervision is kept (Table 37).

The reconstruction criterion is not satisfied throughout this control. At the supervised start, Stieger exceeds the reconstruction-only reference's bank-B loss by 2.07%, with a stored 95% interval of $[1.82, 2.31]\%$. ERP, ISRUC, and TUAB remain within the 1% tolerance. By the end of training, the supervision-kept branch has a lower bank-B reconstruction loss than the reference by 1.93%, 0.99%, 2.88%, and 2.16%, respectively for Stieger, ERP, ISRUC, and TUAB. Stronger supervision can therefore improve retention, although acquisition is not uniformly within the chosen reconstruction tolerance. This control uses only one encoder seed.

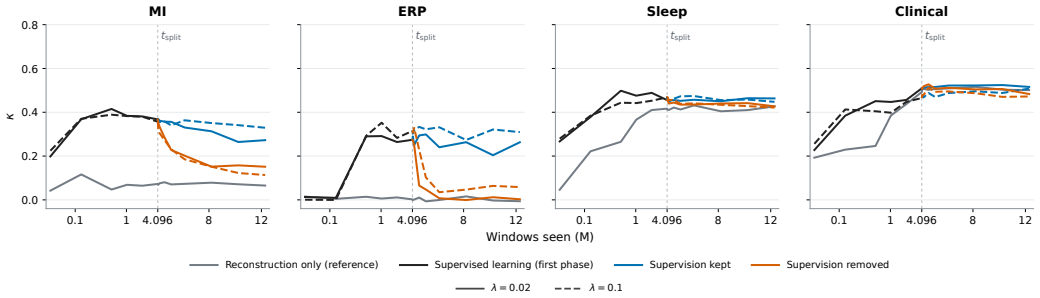


Figure 7: Effect of supervision strength on acquisition and retention. Panels show motor imagery (Stieger 2021), ERP (Lee2019 ERP), sleep staging (ISRUC), and clinical classification (TUAB). Curves report pooled test κ from the attention readout on encoder features Z, averaged over three readout initializations at each checkpoint. Solid and dashed curves use $\lambda = 0.02$ and $\lambda = 0.1$, respectively, both on encoder seed 0. The reconstruction-only reference is shared and shown once. For each supervision weight, the kept and removed branches inherit the complete training state at $t_{\text{split}} = 4.096 \times 10^6$ window presentations; optimizer-reset controls are not included. The horizontal axis is logarithmic before t_{split} and linear afterwards, giving equal width to both phases. The checkpoint at zero is omitted. This single-encoder-seed comparison does not estimate variability across encoder initializations.

Does the pattern persist with other descriptors and readouts? Table 38 reports the full-descriptor and fixed-pooling controls, using the same subjects, checkpoint states, and training budgets. With attention and full H, the initial gains with descriptors remain $+0.159 \pm 0.010$ on Stieger and $+0.223 \pm 0.017$ on ERP. After supervision is removed, the residual gains fall to $+0.031 \pm 0.011$ and -0.003 ± 0.005.

Table 37: Stronger supervision ($\lambda = 0.1$), encoder seed 0. Pooled scores use the attention readout on Z. Gains with descriptors are paired subject-level comparisons with the reconstruction-only reference at the same training budget, using $[H_{512}, Z]$ in both readouts. "Start" is the supervised start at the end of the first phase; "kept" and "removed" are evaluated at the end of training. TUAB gains concern only its 18 eligible subjects.

Corpus	Pooled score on encoder features			Gain with descriptors		
	Supervised start	Supervision kept	Supervision removed	Supervised start	Supervision kept	Supervision removed
Stieger 2021	0.357	0.329	0.113	+0.211	+0.185	+0.028
Lee2019 ERP	0.315	0.309	0.058	+0.285	+0.273	+0.027
ISRUC	0.468	0.448	0.421	+0.012	−0.001	−0.006
TUAB	0.465	0.508	0.472	+0.027	+0.026	+0.010

Global mean pooling also shows an acquired advantage followed by a decline after supervision removal. The electrode-preserving readout gives a weaker Stieger gap between keeping and removing supervision: $+0.027 \pm 0.030$, compared with $+0.158 \pm 0.039$ on ERP. Thus, the pattern extends beyond attention and the compact descriptor block, while its magnitude and consistency depend on the readout. All these classifiers are trained with cross-entropy. Their scores measure predictive performance accessible to the specified readouts; they do not measure all task information in the representation.

Table 38: Readout and descriptor controls, $\lambda = 0.02$. All entries are paired subject-level comparisons, mean ± sample SD across three encoder seeds. Gains with descriptors compare the indicated encoder with the reconstruction-only reference, supplying the listed descriptors to both readouts. "Change after removal" and "kept − removed" use Z alone. Initial gains are measured at the end of the first phase; all other columns use the end of training. Attention with full H shares the primary analysis's Z readout. The electrode-preserving control uses full H only. TUAB estimates concern only 18 eligible subjects.

Readout / descriptors	Initial gain with descriptors	Change after removal	Kept − removed	Final gain with descriptors (removed)
Stieger 2021				
Attention, full H	$+0.159 \pm 0.010$	-0.206 ± 0.014	$+0.138 \pm 0.022$	$+0.031 \pm 0.011$
Global mean, H_{512}	$+0.184 \pm 0.014$	-0.171 ± 0.007	$+0.110 \pm 0.019$	$+0.055 \pm 0.017$
Global mean, full H	$+0.159 \pm 0.028$	-0.171 ± 0.007	$+0.110 \pm 0.019$	$+0.055 \pm 0.016$
Electrodes kept, full H	$+0.055 \pm 0.023$	-0.053 ± 0.017	$+0.027 \pm 0.030$	$+0.009 \pm 0.053$
Lee2019 ERP				
Attention, full H	$+0.223 \pm 0.017$	-0.256 ± 0.015	$+0.240 \pm 0.016$	-0.003 ± 0.005
Global mean, H_{512}	$+0.278 \pm 0.029$	-0.269 ± 0.030	$+0.254 \pm 0.017$	$+0.014 \pm 0.028$
Global mean, full H	$+0.265 \pm 0.012$	-0.269 ± 0.030	$+0.254 \pm 0.017$	-0.005 ± 0.029
Electrodes kept, full H	$+0.168 \pm 0.008$	-0.180 ± 0.005	$+0.158 \pm 0.039$	$+0.017 \pm 0.038$
ISRUC				
Attention, full H	$+0.010 \pm 0.004$	-0.026 ± 0.003	$+0.013 \pm 0.022$	$+0.002 \pm 0.004$
Global mean, H_{512}	$+0.005 \pm 0.006$	-0.019 ± 0.025	$+0.000 \pm 0.017$	$+0.013 \pm 0.011$
Global mean, full H	$+0.017 \pm 0.018$	-0.019 ± 0.025	$+0.000 \pm 0.017$	$+0.011 \pm 0.004$
Electrodes kept, full H	$+0.001 \pm 0.011$	-0.025 ± 0.005	$+0.001 \pm 0.018$	$+0.013 \pm 0.020$
TUAB				
Attention, full H	$+0.004 \pm 0.024$	$+0.053 \pm 0.040$	-0.023 ± 0.044	-0.003 ± 0.002
Global mean, H_{512}	-0.005 ± 0.039	$+0.010 \pm 0.048$	-0.022 ± 0.033	-0.011 ± 0.018
Global mean, full H	-0.025 ± 0.014	$+0.010 \pm 0.048$	-0.022 ± 0.033	-0.018 ± 0.023
Electrodes kept, full H	$+0.013 \pm 0.005$	$+0.039 \pm 0.019$	-0.012 ± 0.061	$+0.017 \pm 0.045$

F.7 What the experiment establishes, and how to reproduce it

Why the design is informative. The principal comparison starts from exactly the same learned model and optimizer state, then keeps or removes the supervised term while preserving the reconstruction data, their order, their masks, and the update budget. It therefore measures the effect of continuing that supervision under this training procedure without changing subject exposure or optimizer initialization. Fitting a new readout at every checkpoint also prevents an outdated training-time classifier from being the sole explanation for a decline.

On the two event-related tasks, the results establish a concrete separation between learning a useful representation and retaining its usefulness. Supervision produces a substantial advantage on new subjects; reconstruction-only continuation erodes much of that advantage; and keeping supervision preserves more of it while remaining within the chosen reconstruction tolerance. This is why evaluating retention adds value to evaluating reconstruction quality: reconstruction loss alone does not reveal whether an acquired predictive capability remains accessible to the tested readouts.

Scope of the conclusion. The inference is limited to this architecture, training recipe, four corpora, and fixed subject partitions. The A pools are small relative to the presentation budget and undergo extensive reuse. Training with supervision uses additional computation. The optimizer-reset and stronger-supervision controls each use only one encoder seed. TUAB's subject-level inference is particularly restricted because most subjects lack both labels in the small evaluation sample.

Neither the reconstruction tolerance nor the available readouts establishes that all task information is preserved or lost. The experiment does not test transfer to unseen tasks or classes, and it does not establish that reconstruction is the unique limitation of other EEG foundation models.

Reproducibility. The implementation separates training, frozen evaluation, and analysis. Training saves the random initialization and the complete supervised start at t_{split}. Evaluation records per-head pooled scores, eligible per-subject scores, selected readout settings, and per-cell reconstruction losses at every checkpoint. The analysis reconstructs the initial-gain, continuation, optimizer-reset, and reconstruction comparisons from those records.

Reported endpoint comparisons use the fixed t_{split} and t_{end} budgets. Intermediate checkpoints describe the trajectory and are not used to choose a stopping point that maximizes test performance. All numerical tables in this appendix are based exclusively on the single-corpus experiments described above.

G DO THE CONCLUSIONS PERSIST WITH FEW DOWNSTREAM LABELS?

The main experiments use the full annotation budget of each retained benchmark subset. Because label efficiency is a central motivation for foundation models, we repeat the three analyses with $K \in \{32, 128\}$ labeled windows per class (validation labels included). The qualitative picture is unchanged: pretraining gains stay concentrated in sleep and clinical tasks and largely overlap with classical descriptors; pretraining on signals that preserve spectra and cross-spectra closely matches real-EEG pretraining; and reconstruction does not keep most of the extra motor-imagery and ERP information acquired under supervision. At $K = 32$, a small residual pretraining gain remains when both readouts receive classical features, but the five-family mean gain of adding pretrained features to classical descriptors alone is small at both budgets, with intervals that include zero.

G.1 LABEL BUDGETS AND EVALUATION PROTOCOL

What is reduced. We reuse the encoder checkpoints, EEG windows, subject partitions, and test windows of the corresponding main experiment. Only the labels available to the downstream classifier are reduced. In the retention experiment, the labels used to train the encoder on subject group A are unchanged; the budget applies to the fresh readout fitted on group B and evaluated on group C. These experiments therefore measure the label efficiency of frozen representations, not low-label encoder training or fine-tuning.

Sampling and validation. For each class, we draw without replacement $3K/4$ windows from the training subjects and $K/4$ from the validation subjects (Table 39). We use five label draws, nested so that the $K = 32$ subset is contained in the $K = 128$ subset. Draws and test windows are shared across the compared conditions. Model selection uses only the budgeted validation labels; the selected classifier is then refitted on all budgeted labels. The budget counts labeled windows, not annotated subjects or the screening cost of building class-stratified subsets.

Table 39: Nominal downstream label budgets, per class. For C classes, the total budget is CK windows. Validation is included, rather than supplied for free.

	$K = 32$	$K = 128$
Training labels per class	24	96
Validation labels per class	8	32
Total labels, binary task	64	256
Total labels, five-class task	160	640

Available labels and unlabeled recordings. All corpora reach the nominal $K = 32$ budget. At $K = 128$, classes with too few windows contribute all available windows of the relevant partition. In the published-encoder panel, actual per-class totals are 79–100 for BETA, 120 for Kalunga2016 and Nakanishi2015, 123–128 for Sleep-EDF, and 47–128 for TUEV; all other corpora reach 128. Token standardization, descriptor standardization, and PCA use all downstream training and validation windows without labels (all of B for retention), identically at every budget; test windows never enter these fits. This is therefore a label-scarce setting with access to extra unlabeled recordings, not a setting with only CK recordings.

Readout recipe. The readout uses a single learned attention query, initialized at zero, followed by a linear classifier. Full-batch Adam minimizes cross-entropy with learning rate 10^{-2}, at most 300 updates, and an L2 penalty chosen from $\{10^{-3}, 10^{-2}, 10^{-1}\}$. Validation κ is computed every five updates to select the penalty and stopping step; ties go to stronger regularization and earlier stopping. Test scores are averaged over head seeds 0, 1, and 2, then over label draws. For the surrogate and retention experiments, we also re-evaluate the full budget with this recipe, so that the low-label curves have a within-recipe reference. These full-budget scores need not match the main-text scores, which used the original readout recipes.

G.2 Published encoders: the pretraining gain still overlaps with classical descriptors

The panel covers 23 encoder configurations on 34 datasets and tasks. The attention comparison uses 515 model–corpus pairs across 27 corpora in the five main families, out of 519 in the full-budget panel; the full-budget reference is restricted to the same pairs. Scores average the available pairs within each family, then weight the five families equally.

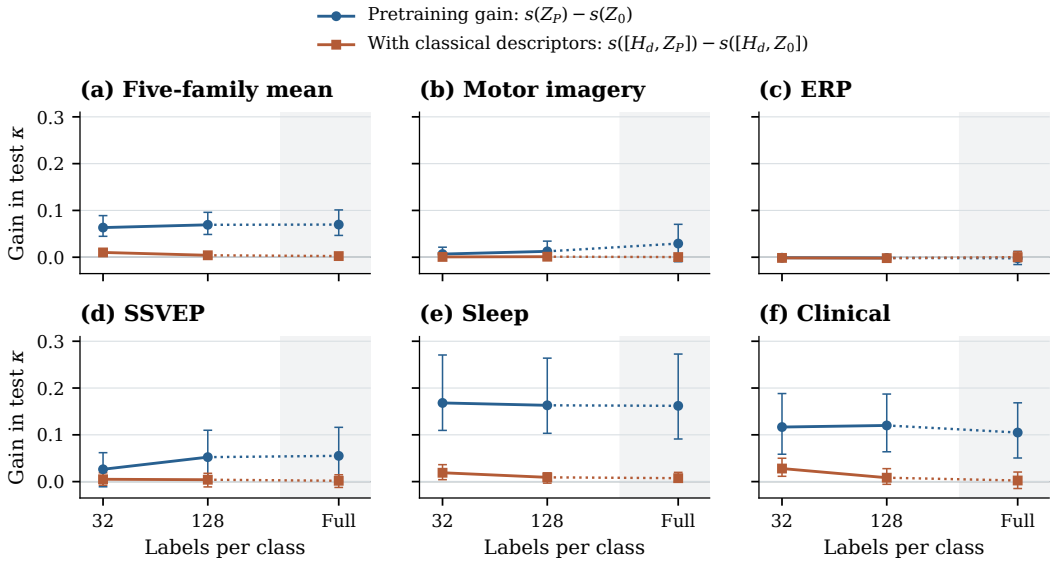


Figure 8: **Pretraining gains under restricted label budgets.** Attention readouts on the same 515 pairs, averaged over five label draws and three head seeds. Error bars are 95% intervals from 2,000 bootstrap resamples of models and corpora independently within each family; the overall mean weights families equally. They quantify matrix variation, not subject or draw uncertainty. Shaded "Full" points use the paper's matched pairs with corpus-dependent annotation counts; horizontal positions are categorical.

Table 40: Five-family means with 95% matrix-bootstrap intervals. $A_H(K)$ is the gain from adding pretrained features to H_d alone, defined below. The full-budget A_H is omitted because the paper's standalone H_d score used a different classifier recipe.

Labels / class	Δ_{pre}	Δ_H	A_H
32	$+0.063$ $[+0.045, +0.089]$	$+0.010$ $[+0.005, +0.017]$	$+0.004$ $[-0.001, +0.013]$
128	$+0.069$ $[+0.049, +0.096]$	$+0.004$ $[-0.001, +0.009]$	$+0.001$ $[-0.003, +0.006]$
Full (paper)	$+0.070$ $[+0.047, +0.101]$	$+0.002$ $[-0.003, +0.008]$	–

The average pretraining gain is $+0.063$ at 32 labels per class and $+0.069$ at 128, versus $+0.070$ at the full budget: it does not grow as labels become scarce. Classical descriptors remove about 84% and 94% of these gains, respectively. Motor imagery and ERP remain weakly served; sleep and clinical tasks account for most of the benefit (Figure 8).

G.3 Residual gains and comparison with standalone classical descriptors

A positive Δ_H does not by itself show that pretrained features improve on H_d alone, because appending random features can hurt the baseline. We therefore also measure

$$A_H(K) = s_K([H_d, Z_P]) - s_K(H_d), \tag{24}$$

with the same descriptor block, labels, classifier family, and selection recipe in both arms. Across the five families, Δ_H is $+0.010$ at $K = 32$ and $+0.004$ at $K = 128$, while A_H is $+0.004$ $[-0.001, +0.013]$ and $+0.001$ $[-0.003, +0.006]$. The positive residual contrast at $K = 32$ therefore does not establish a positive mean improvement over standalone classical descriptors. At $K = 128$, A_H is only $+0.003$ in sleep and $+0.001$ in clinical tasks.

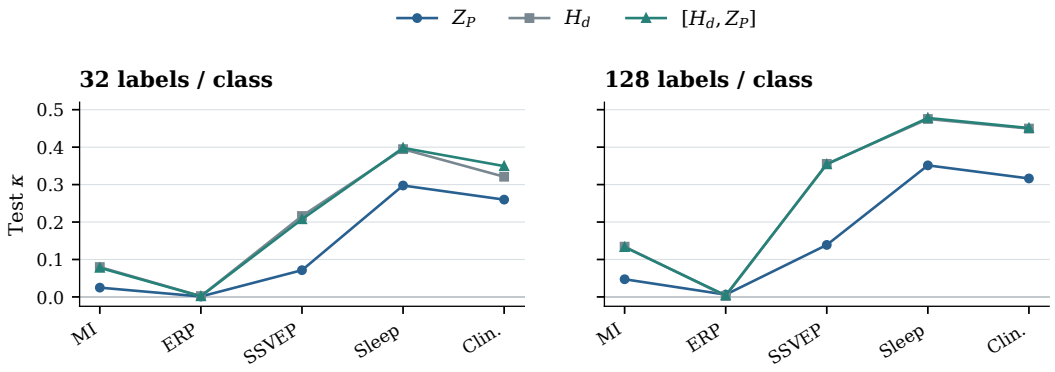


Figure 9: **Classical descriptors remain a strong low-label reference.** Absolute attention-readout scores, averaged over the available model–corpus pairs within each family. H_d alone uses cross-entropy fitting here and its width matches the descriptor block paired with each model. Each point averages five label draws and three head initializations. Lines connect task families only to aid visual comparison; they do not represent a continuous variable.

Absolute scores. In motor imagery, SSVEP, and sleep, H_d alone outperforms Z_P at both restricted budgets. ERP stays near chance. In short, classical descriptors account for most of the pretraining advantage at both budgets; the small residual contrast at $K = 32$ does not translate into a clear gain over descriptors alone.

G.4 Surrogate pretraining transfers with few downstream labels

We re-evaluate the real-EEG and surrogate pretraining checkpoints on the same six real-EEG datasets: BNCI2014-001, PhysionetMI, Lee2019 ERP, Lee2019 SSVEP, ISRUC, and TUAB. For each encoder seed, we average over head initializations and label draws, then weight the six corpora equally. Mean and standard deviation are across the three pretraining seeds, except for the mean-and-variance surrogate and the REVE-base controls, which have one run per condition.

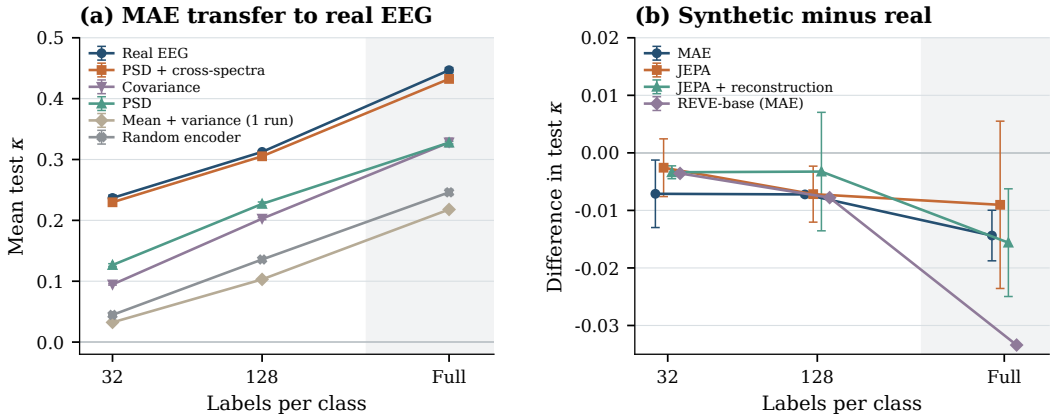


Figure 10: **The real-versus-synthetic transfer comparison persists at low label budgets.** (a) Six-corpus mean transfer of frozen REVE-small encoders pretrained by masked reconstruction (MAE). (b) Paired difference between pretraining with the PSD-plus-cross-spectra surrogate and with real EEG, under three objectives and the larger REVE-base architecture. Error bars show one standard deviation across three encoder seeds; the single-run controls have no uncertainty bar. All downstream inputs are real EEG. The full budget is re-evaluated with the same classifier recipe as the restricted budgets.

Table 41: Six-corpus mean transfer κ of REVE-small MAE encoders, with standard deviation across encoder seeds. The random encoder has three initializations; the mean-and-variance surrogate has one pretraining run. "Full" uses the new readout recipe throughout.

Pretraining signal	32 / class	128 / class	Full
Real EEG	0.237 ± 0.005	0.313 ± 0.001	0.447 ± 0.004
PSD + cross-spectra	0.230 ± 0.002	0.305 ± 0.001	0.432 ± 0.002
Covariance	0.095 ± 0.001	0.203 ± 0.002	0.328 ± 0.007
PSD	0.127 ± 0.002	0.227 ± 0.003	0.328 ± 0.002
Mean + variance (1 run)	0.032	0.103	0.218
Random encoder	0.045 ± 0.001	0.136 ± 0.001	0.246 ± 0.005

At 32 labels per class, real-EEG pretraining reaches 0.237 κ and the PSD-plus-cross-spectra surrogate 0.230, against 0.045 for random weights. At 128 labels per class, the scores are 0.313, 0.305, and 0.136. The paired synthetic-minus-real difference, computed before rounding, is -0.007 at both restricted budgets and -0.014 at the full budget under this readout. Covariance-only and PSD-only pretraining fall well below real EEG. The same close comparison holds under JEPA, JEPA plus reconstruction, and REVE-base (Figure 10b). These are small observed differences, not a formal equivalence test, but they support the main-text interpretation: most of the low-label transfer measured here can be learned from spectra and cross-spectra.

G.5 Reconstruction still loses the extra information acquired under supervision

We reuse the single-corpus bifurcation checkpoints ($\lambda = 0.02$, three encoder seeds). R is trained with reconstruction only. J adds task supervision until 4.096 million window presentations; from that point, JC keeps supervision and JR removes it, both running to 12.288 million presentations. Each checkpoint receives a fresh budgeted readout on B. Figure 11 reports pooled test κ, averaged over five label draws and three readout seeds per encoder.

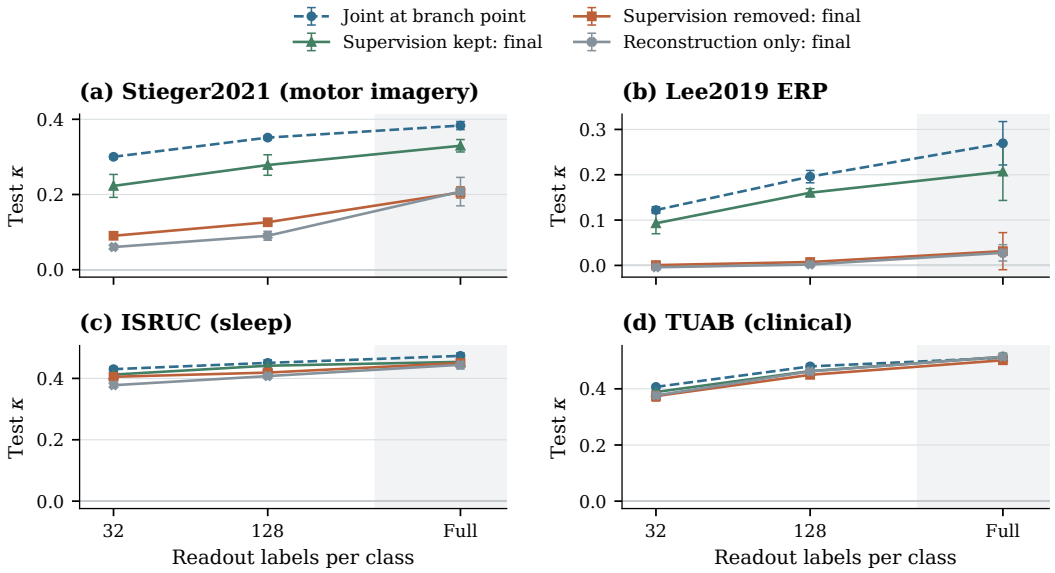


Figure 11: **Restricting readout labels does not remove the retention gap.** Attention-readout scores at the branch point and final exposure. Encoder training is unchanged; only fresh readout labels vary. Error bars show one standard deviation over three encoder seeds. Full-budget points use the same readout recipe as low-budget points, rather than the original retention readout.

On Stieger2021 at $K = 32$, branch-point scores are 0.059 for R and 0.300 for J; at final exposure, R, JC, and JR score 0.060, 0.223, and 0.090. On Lee2019 ERP, branch-point scores are 0.000 and 0.122, and final scores are -0.004, 0.093, and 0.001. Most of the supervised advantage is thus lost once supervision is removed. When both readouts also receive H_{512}, the final JR-over-R contrast stays small at $K = 32$ and $K = 128$, whereas keeping supervision preserves a larger contrast (Table 42).

Table 42: The same comparison when both attention readouts receive H_{512}. Entries are differences of pooled test κ, mean $\pm$ standard deviation across three encoder seeds. P_1 is the branch point and T is final exposure. They are not the subject-averaged contrasts reported elsewhere in the paper.

Corpus	K	$J(P_1) - R(P_1)$	$JC(T) - R(T)$	$JR(T) - R(T)$
Stieger2021	32	$+0.158 \pm 0.021$	$+0.071 \pm 0.022$	$+0.004 \pm 0.002$
Stieger2021	128	$+0.180 \pm 0.015$	$+0.110 \pm 0.024$	$+0.021 \pm 0.003$
Lee2019 ERP	32	$+0.096 \pm 0.008$	$+0.066 \pm 0.015$	$+0.003 \pm 0.004$
Lee2019 ERP	128	$+0.125 \pm 0.006$	$+0.098 \pm 0.019$	-0.003 ± 0.003

G.6 SUPERVISED DECODING CONTROLS AND SCOPE OF THE CONCLUSION

The supervised surrogate control depends on the budget. Frozen transfer should not be confused with training a supervised decoder directly on each version of the downstream signal. We repeat the EEGNetv4 control on the six corpora, training and testing either on real EEG or on its PSD-plus-cross-spectra surrogate. At $K = 32$, only eight validation windows per class exist, so validation-based stopping can select very short schedules. We therefore also run a fixed schedule of 1,000 updates on all budgeted labels, without validation-based selection (Figure 12).

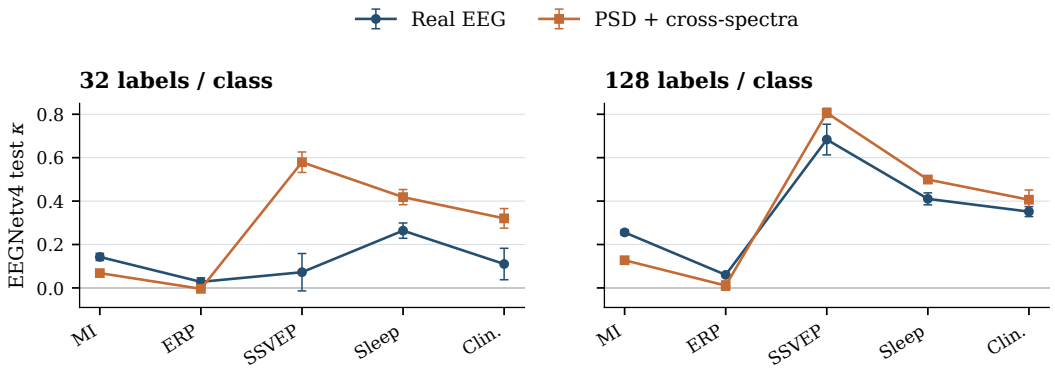


Figure 12: **The supervised surrogate control is not unchanged at low budgets.** EEGNetv4 trained from scratch for 1,000 updates on all available budgeted labels, with training and test inputs both real or both synthetic. Motor imagery averages BNCI2014-001 and PhysionetMI; each other family is represented by one corpus. Error bars are standard deviations over five draws, which jointly vary the label subset and network initialization. A fresh surrogate is sampled for each training batch; validation/test surrogates use fixed random seeds. These scores are distinct from frozen-encoder transfer.

The surrogate lowers motor-imagery decoding, but the mean degradation seen at the full budget does not hold for every low-label task. At 32 labels per class under the fixed schedule, synthetic versus real inputs give roughly 0.58 versus 0.07 for SSVEP, 0.42 versus 0.26 for ISRUC, and 0.32 versus 0.11 for TUAB. Fresh phase randomization at each batch also acts as a training-time augmentation, which may help these data-limited decoders, so these results cannot be read purely as an information-removal effect. ERP is near chance in both conditions, so this control says nothing about ERP information at these budgets.

What the low-label results establish. Reducing downstream labels does not make the evaluated encoders uniformly useful across task families. Pretraining still helps mainly where classical spectral and spatial structure is useful; synthetic pretraining that preserves this structure still transfers about as well as real-EEG pretraining; and reconstruction still loses most of the extra motor-imagery and ERP advantage installed by supervision. A small residual pretraining contrast remains at $K = 32$ when descriptors are given to both readouts, but the mean gain over standalone descriptors is small at both restricted budgets. The conclusion is about the structure and task dependence of the benefit; it does not claim that pretraining never helps with few labels.

Limits. The budgets restrict labeled windows, not annotated subjects, and assume access to extra non-test recordings for label-free preprocessing. All encoders stay frozen during downstream fitting. We do not extrapolate to fine-tuning, subject-budget scaling, or an annotation process that must first discover class membership. Full-budget values are corpus-dependent and are not a common number of labels. Matrix-bootstrap intervals, encoder-seed standard deviations, and label-draw standard deviations measure different sources of variation and are labeled separately throughout.

Table 43: **Where pretraining helps least, supervised decoding goes far beyond the descriptors.** Test Cohen's κ on the six datasets of Section 3 ($\kappa = 0$ at chance). The four frozen-encoder columns average the encoder configurations evaluated on each dataset (16 for BCI IV-2a, 21 for the others), with attention pooling. EEGNetv4 is trained from scratch on the same windows and subject splits; its score averages three training seeds.

		Frozen encoders (mean over models)				Supervised
Family	Dataset	Z_0	Z_P	$[H_d, Z_0]$	$[H_d, Z_P]$	EEGNetv4
Motor imagery	BCI IV-2a	0.056	0.095	0.187	0.193	0.440
	PhysioNet MI	0.077	0.081	0.190	0.186	0.395
ERP	Lee2019 ERP	0.006	0.005	0.004	0.001	0.341
SSVEP	Lee2019 SSVEP	0.271	0.377	0.654	0.653	0.894
Sleep	ISRUC	0.298	0.432	0.591	0.595	0.537
Clinical	TUAB	0.298	0.437	0.568	0.578	0.532

H WHERE PRETRAINING BARELY HELPS, MUCH REMAINS TO BE LEARNED

A residual gain near zero has two possible meanings. The descriptors may already capture what the task requires, leaving nothing for pretraining to add; or the task may depend on information that neither the descriptors nor the pretrained features provide. Telling them apart requires a reference for how well each task can be decoded from the same windows. We therefore train a supervised network, EEGNetv4 (Lawhern et al., 2018), from scratch on the same windows and splits of the six datasets we use again in Section 3 (Table 43).

On sleep staging (ISRUC) and abnormality detection (TUAB), $[H_d, Z_0]$ already scores at least as high as EEGNetv4, leaving little headroom relative to this supervised reference. On motor imagery and ERP, EEGNetv4 scores substantially higher than $[H_d, Z_0]$, while pretrained features add little on average beyond the descriptors. On Lee2019 ERP, descriptors and pretrained features remain near chance, while EEGNetv4 reaches $\kappa = 0.34$. SSVEP also shows a substantial gap that pretrained features do not close. The frozen readout can nevertheless detect substantial gains when task-relevant features are learned with supervision: in Section 4, with descriptors supplied to both heads, the paired comparison detects gains of 0.20 κ on motor imagery and 0.26 on ERP for an encoder trained with task labels.

Under these readouts, the supervised EEGNet reference reveals substantial remaining headroom on MI and ERP, while pretrained features add little on average beyond the classical descriptors.

I REPLICATION OF SECTION 3 RESULTS WITH LABRAM AND ITS NATIVE PRETRAINING OBJECTIVE

Why this experiment. Section 3 keeps the REVE-small encoder fixed and varies only the pretraining signal and objective. This isolates the effect of the signal, but it leaves open whether the finding depends on REVE. Here we test the same question on a second architecture, LaBraM-base (Jiang et al., 2024), trained with its own discrete prediction objective.

Why REVE is the main backbone. Pretraining EEG foundation models from scratch is expensive, so we could not run every architecture under every objective, signal, and seed. We chose REVE for the main comparisons because it is the strongest model in our five-family benchmark and a leading model in NeuralBench and EEG-FM-Bench (Banville et al., 2026; Xiong et al., 2026). With the REVE-small encoder fixed, we varied the objective across masked reconstruction (MAE), masked latent prediction (JEPA), and their combination (inspired by EEGPT (Wang et al., 2024)), which covers the main reconstruction and latent-prediction families.

The main grid is 3 objectives $\times$ 4 signal conditions $\times$ 3 seeds, i.e. 36 pretraining runs. We added four REVE-base MAE runs (one per signal condition; 69.2M vs. 12.7M parameters) and one REVE-small MAE run on the Mean + variance control, for 41 runs at the standard budget. Separate runs test a fivefold larger budget for real EEG and PSD + cross-spectra. Repeating all of this for a second architecture was out of budget, so we instead ran the targeted replication below.

Protocol. LaBraM predicts discrete codes from a learned VQ-NSP tokenizer. For each signal condition we train both the tokenizer and the encoder from scratch, with a separate tokenizer per condition. Both stages use the same TUEG data as the REVE experiments and about 20.8 million window presentations. We then freeze the encoder and evaluate transfer on the same six real-EEG dataset–task combinations with the same attention-pooling readout. Table 44 lists the number of completed runs per condition.

Table 44: Frozen transfer with LaBraM to real EEG. Each run's score averages three readout initializations and weights the six dataset–task combinations equally. Values are mean ± sample SD across n pretraining runs, or n encoder initializations for the random baseline. No SD is reported for the single-run Mean + variance condition. The Covariance results include one run with tokenizer collapse.

Pretraining signal	n	**Transfer** κ
Random initialization	3	0.144 ± 0.006
Real EEG	3	0.234 ± 0.008
PSD + cross-spectra	3	0.249 ± 0.014
PSD	2	0.202 ± 0.017
Covariance	2	0.181 ± 0.078
Mean + variance	1	0.118

Results. PSD + cross-spectra reaches a mean transfer score of 0.249 κ, against 0.234 for real EEG and 0.144 for random initialization. The paired synthetic-minus-real difference is $+0.015 \pm 0.019$ κ (mean ± sample SD over three seeds). PSD and Covariance score lower, and Mean + variance does not beat random initialization. The Covariance estimate is noisy and includes one run with tokenizer collapse.

The pattern matches Section 3: with a second architecture and a discrete prediction objective, pretraining on signals that preserve whole-window spectra and cross-spectra matches the average transfer of real EEG. As before, this is an aggregate result; ERP performance stays near chance in both conditions.

J Full fine-tuning evaluation

J.1 Why we mainly focus on frozen encoders

Our main focus in this paper is studying the predictive value accessible in an encoder before supervised adaptation. Our objective is to characterize what pretraining encodes in the weights rather than to maximize downstream performance. Keeping the encoder fixed and training only the readout makes this possible: the representation under evaluation is the one produced by pretraining, so the predictive signal we measure reflects what that representation makes accessible to a small readout. This choice lets us compare pretrained and randomly initialized representations, assess their predictive overlap with classical EEG descriptors, and track the effects of pretraining and continuation without allowing target labels to reshape the encoder. The conclusions concern the representations and readouts evaluated under this protocol.

Full fine-tuning evaluates the utility of a pretrained initialization when target labels also update the encoder. Because supervised training modifies the encoder, the resulting representation mixes what pretraining produced with what the target labels induced, and predictive signal can no longer be attributed to pretraining alone; this is why the interpretation and diagnosis in the main paper rely on frozen encoders. Fine-tuning instead addresses a different question: comparing pretrained and random initializations under the same adaptation protocol measures a benefit of pretraining as an initialization in this setting, including its interaction with supervised learning. We examine this question for REVE-base on five task families, with two label budgets.

Setup. We compare the published REVE-base checkpoint with a randomly initialized copy on PhysioNet MI Schalk (2009), ERR Lee et al. (2019), SSVEP, ISRUC, and TUAB, using the paper's windows, subject-disjoint splits, and attention-pooling head, without classical descriptors. We eval-

uate direct end-to-end training (FT) and head training with a frozen encoder followed by end-to-end training (LP-FT), at the full label budget and at 32 labeled windows per class. Results weight the five corpora equally. Within each adaptation protocol, the pretrained and random initializations receive the same optimization and hyperparameter search budget; encoder learning rates and checkpoints are selected separately on validation performance.

We use AdamW with a head learning rate of 10^{-3} and weight decay of 10^{-4}. Encoder learning rates are $\{3\times10^{-5}, 10^{-4}, 3\times10^{-4}, 10^{-3}\}$ at the full budget and $\{10^{-5}, 3\times10^{-5}, 10^{-4}\}$ at 32 labels per class. Full-budget learning-rate selection averages validation scores over three training seeds. The smaller budget uses five label draws, each with 24 training and 8 validation windows per class, and one training seed per draw. Fine-tuning does not refit on validation data, whereas the frozen references retain the main paper's train–validation refit. Comparisons between frozen and adapted scores should therefore be interpreted with this protocol difference in mind.

Absolute performance and the pretraining advantage. At the full label budget, FT improves mean absolute performance for both initializations relative to their frozen references ($+0.051$ κ for pretrained weights and $+0.061$ for random weights). The mean pretrained–random gap is $+0.176$ κ with frozen encoders, $+0.165$ after FT, and $+0.166$ after LP-FT (Figure 13a). At 32 labels per class, the corresponding gaps are $+0.209$, $+0.128$, and $+0.212$ (panel b). Thus, under the main attention-pooling readout, adaptation can improve downstream scores without a correspondingly larger average advantage from pretrained initialization.

This aggregate pattern does not establish equivalence between the protocols: task-specific gains vary, and with global mean pooling the full-budget gap increases from $+0.122$ κ frozen to $+0.200$ after FT. The effect of adaptation therefore depends on the readout and training procedure.

Target-task performance and transfer after adaptation. In a separate diagnostic, we fine-tune the pretrained encoder on SSVEP, freeze it again, and fit fresh readouts on ISRUC and TUAB. Across four encoder learning rates, SSVEP scores range from 0.861 to 0.890 κ, while transfer to the other two tasks varies more substantially (Figure 13c–d). At learning rates 10^{-4} and 10^{-3}, SSVEP scores are 0.890 and 0.883, but the mean fraction of the original pretraining gain retained on ISRUC and TUAB is 87% and 26%, respectively. The validation-selected rate is 10^{-4}. This single-seed diagnostic illustrates that similar target-task scores can coexist with different transfer performance after adaptation; it does not establish that fine-tuning inevitably degrades transfer.

Scope. These experiments cover one published checkpoint, one random encoder initialization, and five datasets under the tested adaptation budgets. The full campaign, including pooling controls and head-only runs, required approximately 62 GPU-hours. They provide a complementary assessment of pretraining as an initialization, while the frozen analyses characterize its accessible predictive value before task-specific encoder adaptation. The two evaluations thus address complementary aspects of the contribution of pretraining.

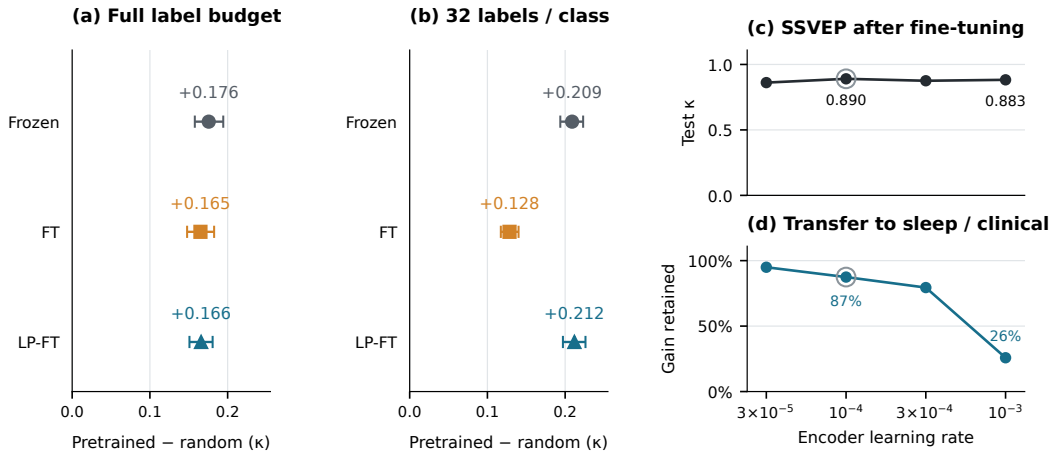


Figure 13: **Pretraining as a frozen representation and as an initialization for adaptation. (a–b)** Mean pretrained–random gap across five equally weighted corpora, with attention pooling and without classical descriptors. Error bars are 95% paired test-subject bootstrap intervals, conditional on the fitted runs. Frozen heads use three seeds; FT and LP-FT use three training seeds at the full budget and one seed for each of five label draws at 32 labels per class. **(c–d)** SSVEP fine-tuning from pretrained weights, with one training seed and four encoder learning rates; the ring marks the rate selected on SSVEP validation performance. Retained gain averages $(s_{\text{after}} - s_0)/(s_P - s_0)$ over ISRUC and TUAB, using fresh frozen readouts. Here, s_P and s_0 are the pretrained and random frozen references for each task, and s_{after} is the score after SSVEP fine-tuning.