SELF-SUPERVISED LEARNING WITH RECURSIVE QUADRATURE FILTERS FOR NEURAL TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

We develop self-supervised methods for training Recursive Quadrature Filter (RQF) encoders on unlabeled time series. RQFs are biologically motivated, complex-valued temporal filters that form a special class of diagonal state-space models (SSMs). Each RQF acts as a band-pass filter whose selectivity is determined by two learnable parameters: its tuning frequency and bandwidth. To train these encoders, we introduce a self-supervised learning objective based on prediction error; the encoder and a prediction head are trained to predict targets derived from the encoder's future states. In the *windowed* mode, the head uses the summary of the current window to predict the summary of a later window; in the *dense* mode, it uses the state at the current timestep to predict future log-power. Each RQF encoder is pretrained on a dataset and then a readout (e.g., a task-specific classifier) is trained on the output of the RQF encoder, with the encoder either frozen or fully fine-tuned. For multichannel EEG and intracranial recordings, a shared encoder processes each channel separately, and a spatial readout combines the channel representations using electrode locations. Across four neural benchmarks using scalp or intracranial EEG, the pretrained RQF system performs close to the state of the art while using a small fraction of the parameters of the competing models. For example, a 110k-parameter RQF system reaches 0.9130 ± 0.0036 test AUROC on TUAB, within the range achieved by state-of-the-art EEG foundation models with 1.0M–369M parameters.

1 INTRODUCTION

Self-supervised pretraining is widely used to learn general-purpose representations of physiological time series. The leading encoders for electroencephalography (EEG) often follow the recipe established in language and vision: raw signals are divided into fixed patches with learned embeddings, a Transformer is trained to predict masked patches or discrete codes, and the encoder is scaled to millions or hundreds of millions of parameters [1–3]. The resulting systems perform well across clinical benchmarks. However, because scale, learned input embeddings, and masked pretraining arrive together, it is difficult to tell how much of the result comes from capacity and how much from the recipe. Nor do these encoders explicitly model the oscillatory dynamics that are characteristic of neural recordings. This leaves open whether an encoder designed to model neural oscillations can achieve comparable performance with far fewer parameters and without masked pretraining.

Here, we develop self-supervised learning methods for Recursive Quadrature Filters (RQFs), a class of biologically motivated complex-valued temporal filters [4, 5]. Each RQF learns a tuning frequency and a bandwidth and behaves as a band-pass filter. We stack these filters in layers to form an *encoder*. Each filter maintains a complex-valued *state* that evolves as it processes the input. The states in the final layer form a vector representation of the signal at each timestep. This architecture is therefore inherently suited to physiological signals such as EEG and intracranial recordings, whose information is carried by rhythms in distinct frequency bands [6]. Because RQFs are a special case of diagonal state-space models (SSMs), they can be trained with the same computationally-efficient parallel-scan algorithms as modern diagonal SSMs. The recurrence of each RQF is *prospective*: it predicts how the oscillation it tracks will evolve (Section 3). In addition, the bottom-up input to each layer is itself prospective; the layer is driven by a fixed first-order look-ahead of its input that compensates for integration lag without adding learnable parameters [4].

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.

We capitalize on these prospective dynamics with self-supervised learning based on prediction error. A prediction head uses an earlier representation to predict a target computed from the encoder's states tens of milliseconds to seconds later. The loss compares this prediction with the actual target, so training requires no task labels. The *windowed* objective divides the sequence of final-layer states into windows, summarizes each with a deterministic operator or a learned pooler, and uses one window's summary to predict the summary of a later window. The *dense* objective instead uses the encoder state at an individual timestep to predict log-power over a future interval, without pooling the source states into a window summary. The self-supervised objective trains the encoder, including the phase-prior frequency and damping parameters of its RQF filters, together with the prediction head to predict targets derived from future states at longer horizons. Section 2 relates this objective to contrastive predictive coding (CPC), the joint-embedding predictive architecture (JEPA) framework, and previous self-supervised state-space encoders.

To assess RQFs trained with self-supervised prediction, we conducted experiments on four benchmarks. After pretraining, we train a task-specific readout with the encoder either held fixed (*frozen*) or updated jointly with the readout (*fully fine-tuned*). Keeping the encoder fixed tests whether its pretrained representations already provide useful information for the task.

We apply these evaluations to detect abnormal EEG recordings, sentence-onset detection, sleep staging, and motor imagery. On the Temple University Hospital Abnormal EEG Corpus (TUAB) [7], we compare our windowed system, pretrained only on TUAB, with larger EEG foundation models. We also compare otherwise identical models with and without pretraining across different amounts of labeled training data to determine when pretraining helps most. On BrainTreebank [8], we compare the frozen and fully fine-tuned encoder for sentence-onset detection. For five-class sleep staging, we test the windowed system on subgroup 1 of the sleep dataset from the Institute of Systems and Robotics at the University of Coimbra (ISRUC-SG1) [9]. Finally, we test cross-corpus transfer for motor imagery on Brain–Computer Interface Competition IV dataset 2a (BCI-IV-2a) [10], using weights pretrained on the Temple University Hospital EEG Corpus (TUEG) [7].

We find that small encoders perform surprisingly well. Across all four benchmarks the pretrained RQF system performs close to the state of the art with a small fraction of the parameters of the competing models (Figure 2). For example, with 110k parameters, windowed pretraining followed by full fine-tuning reaches a test area under the receiver operating characteristic curve (AUROC) of 0.9130 ± 0.0036 on TUAB, within the range reported by EEG foundation models with 1 million to 369 million parameters. On intracranial BrainTreebank, we achieve 0.9102 ± 0.0093 AUROC with the pretrained RQF encoder frozen, 5.7 points above the best model in BaRISTA's chronological evaluation, which uses 3.4 times as many parameters.

2 Related work

2.1 Predictive representation learning

Contrastive predictive coding (CPC) predicts future latent representations from an autoregressive context [11]. Later speech methods instead mask parts of the latent sequence: wav2vec 2.0 uses contrastive prediction of quantized targets, while HuBERT predicts clustered targets [12, 13]. Joint-embedding predictive architectures predict representations of masked image or video regions using a target encoder updated by an exponential moving average (EMA) [14, 15]. I-JEPA predicts masked image blocks from a single context block, and V-JEPA extends this approach to masked spatio-temporal regions of video. Both combine a vision Transformer context encoder, a lightweight predictor, and an EMA target encoder. The premise is that predicting in representation space lets these models discard irrelevant or unpredictable pixel-level detail.

Our objective predicts future representations within the same sequence. The windowed InfoNCE configuration is closely related to CPC. Its regression variant shares JEPA's feature-prediction formulation, but uses targets derived from later states of the same encoder instead of a separate target encoder. Predicting forward in time requires no input masking or construction of separate context and target views, and neither configuration requires a quantizer. The RQF architecture also provides part of the predictive structure: its recurrence extrapolates the oscillation it tracks by one time constant, while the learning objective trains prediction over horizons spanning many time constants. We use no moving-average teacher. In JEPA, this teacher helps prevent representational collapse; our objective

instead discourages collapse through detached targets, a variance penalty on the window summaries, and, in the InfoNCE configuration, in-batch negatives.

2.2 Self-supervision with state-space encoders

State-space encoders have also been trained with several self-supervised objectives. Audio Mamba uses input-dependent state transitions to reconstruct masked spectrogram patches. It outperforms a matched Transformer with fewer parameters: 4.8M and 5.4M parameters, respectively [16, 17].

Two aspects of Audio Mamba are relevant to our design: scan direction and input-dependent parameters. Its scan-direction ablation reports an aggregate normalized score of 76.3 for the unidirectional scan and 65.7 for the bidirectional scan. The authors attribute this difference to acoustic content evolving forward in time. The RQF encoder is likewise causal: its phase-rotation prior and first-order prospective input both extrapolate forward from present and past samples, with no backward scan. Audio Mamba inherits Mamba's input-dependent state-space parameters, although the paper does not isolate the contribution of this input dependence. Our RQF parameters, including the learned poles, remain input-independent in all experiments reported here.

For physiological signals, other approaches include CPC with a structured SSM for electrocardiography (ECG), a contrastive S4-family encoder for physiological waveforms, and predictive, joint-embedding, masked-reconstruction, and diffusion objectives for ECG and EEG [18–23]. CaMBrain is closely related in its use of causal state-space dynamics and prediction of future representations. It is a strictly causal Mamba-3 encoder for EEG that uses complex-valued, rotary state transitions and predicts latent representations several steps into the future [24]. CaMBrain uses an EMA teacher, masking, and input-dependent rotations. Our approach uses no teacher or masking, and the RQF encoder's learned poles are input-independent.

2.3 EEG and intracranial foundation models

The EEG models compared in Figure 2 use several pretraining objectives. LaBraM, CBraMod, and REVE use masked prediction with Transformer encoders. BIOT combines contrastive pretraining on unlabeled recordings with supervised pretraining across labeled datasets, while SAMBA uses masked reconstruction with a Mamba encoder–decoder [1–3, 25, 21]. These models, applied to the TUAB benchmark, range from 1M to 369M parameters. Our models are orders of magnitude smaller.

SAMBA provides a closely matched baseline on TUAB. Its TUAB variant, SAMBA-100s, is pretrained on TUAB itself and takes the same input configuration as our model: sixteen channels, each containing 100 seconds sampled at 128 Hz [21]. SAMBA transforms the raw signal through a learned spatial projection, a multi-branch temporal embedding, and a linear projection before the first Mamba2 block. Its deeper Mamba blocks operate on features downsampled by convolution and max pooling. The RQF recurrence, by comparison, operates directly on the waveform. On TUAB, SAMBA uses 1.0M parameters and our model uses 110k. Other state-space encoders designed for efficient EEG processing report smallest variants of 4.6M and 7.8M parameters [22, 26].

Another consideration is how models handle differences in electrode positions and counts. REVE uses a four-dimensional Fourier positional encoding applied to learned patch embeddings [3]. Our readout instead uses electrode coordinates to combine channel representations through a learned spatial basis, so the same readout can handle different electrode layouts. For intracranial EEG, we pretrain across sessions with different electrode counts and evaluate on sessions held out from pretraining.

We also compare our models against published results for Brant, BrainBERT with a Population Transformer, and BaRISTA, each of which have 1M to 500M parameters [27–30]. These models embed signal patches or spectrogram frames before the encoder. Our models are orders of magnitude smaller and take the raw electrophysiological signals as input. The intracranial reference results in Figure 2 come from BaRISTA's chronological evaluation; the differences between those evaluation splits and ours are described in Appendix A.

3 BACKGROUND: RECURSIVE QUADRATURE FILTERS

A *Recursive Quadrature Filter* (RQF) [4] is a scalar complex-valued temporal filter derived from the energy-based theory of cortical function of Heeger [31], in which pairs of neurons with a 90° phase offset compute their priors recursively from their previous responses. It receives an input drive $z(t) \in \mathbb{C}$, and its state $y(t) \in \mathbb{C}$ evolves as

$$\tau\, \dot{y} \;=\; -y + \gamma\, z + (1-\gamma)(1+\iota\omega\tau)\, y, \tag{1}$$

where $\tau > 0$ is a time constant, fixed during pretraining (Appendix B); $\omega > 0$ is the learnable phase-prior frequency; and $\gamma \in (0, 1)$ is the learnable damping, which sets the bandwidth. With $\alpha \triangleq \gamma/\tau$, $\omega_0 \triangleq (1-\gamma)\omega$, and $\lambda \triangleq -\alpha + \iota\omega_0$, Eq. (1) reduces to $\dot{y} = \lambda y + \alpha z$, a band-pass filter with unit peak gain at ω_0 and half-power bandwidth 2α. Note that γ enters both α and ω_0, and that the effective decay time is τ/γ rather than τ. The filter is unconditionally stable for $\gamma > 0$, since $\mathrm{Re}\,\lambda = -\alpha < 0$; restricting $\gamma < 1$ keeps the peak frequency ω_0 positive.

The prior term $(1+\iota\omega\tau)\, y$ is the first-order extrapolation of the state by one time constant under a phase-rotation model, i.e., $e^{\iota\omega\tau} y \approx (1+\iota\omega\tau)\, y$. This term moves the pole off the real axis; consequently, the state leads its drive at positive frequencies below ω_0, whereas with $\omega = 0$ the RQF reduces to a leaky integrator that lags its drive at every positive frequency. Following Rawat et al. [4], we also make the drive prospective: each filter is driven by $z + \tau\dot{z}$ in place of z, a first-order look-ahead of its input. We discretize Eq. (1) by zero-order hold at the sampling interval Δt and set $\tau = 2\Delta t$. Under this hold the derivative of the input is an impulse at each sample, so the prospective drive becomes an exact two-tap update on the current and preceding input samples that introduces no trainable parameters.

An encoder stacks L layers of d such filters, with learned per-filter parameters $\boldsymbol{\omega}^{(l)}, \boldsymbol{\gamma}^{(l)} \in \mathbb{R}^d$. The first layer is driven by the raw input (e.g., electrophysiological) signals, $\mathbf{z}_n^{(1)} = \mathbf{W}^{(1)} x_n$, where $x_n \in \mathbb{R}$ is the n-th sample of one input channel at the dataset-specific sampling rate; deeper layers receive $\mathbf{z}_n^{(l)} = \mathbf{W}^{(l)} \rho\Big(\mathbf{y}_n^{(l-1)}\Big)$ for $l > 1$. Here $\mathbf{W}^{(l)}$ is a complex mixing matrix and ρ is an output nonlinearity, e.g., split-ReLU applied to the real and imaginary components. Self-supervised pretraining optimizes the encoder, including its phase-prior frequencies and damping, together with the prediction head. The objective is to predict the encoder's own future representations at horizons far longer than the one-time-constant look-ahead, from 62.5 ms to 2 s (64 to 800 time constants, depending on the sampling rate) (Section 4).

The top-layer states $\mathbf{y}_n^{(L)} \in \mathbb{C}^d$ form the representation used during pretraining and by the downstream task readout. Because the RQF is a special case of a diagonal complex state-space model, the encoder trains with the same parallel-scan algorithms as modern SSMs. The encoders that we trained for the results reported below have 30k to 140k parameters (Appendix B).

4 SELF-SUPERVISED LEARNING WITH RQF ENCODERS

The RQF encoder maps a raw sequence of T input samples to top-layer states $\mathbf{y}_1^{(L)}, \ldots, \mathbf{y}_T^{(L)}$ with $\mathbf{y}_n^{(L)} \in \mathbb{C}^d$. Self-supervised pretraining uses no task labels: the encoder is trained to predict targets derived from its own future states. Figure 1 shows the complete system in its multichannel windowed configuration, in which one encoder serves two stages. During pretraining, the encoder processes each channel independently, and the prediction objective is applied to summaries of its state sequence. For downstream tasks, a readout combines these summaries across electrodes (taking into account the electrode locations) to make a task prediction.

We consider two learning objectives. The *windowed* objective uses a window summary to predict the summary of a later window, whereas the *dense* objective uses the encoder state at an individual timestep to predict future log-power. For both objectives, a source representation $\mathbf{s}_a$ is derived from the states corresponding to an earlier time a. A learned *prediction head* f_ϕ, a multilayer perceptron (MLP), maps it to an estimate $\hat{\mathbf{t}}_a = f_\phi(\mathbf{s}_a)$ of the *positive target* $\mathbf{t}_a^+$, derived from later encoder states. We compare this estimate with the positive target and K negative targets $\mathbf{t}_{a,k}^-$ using cosine similarity; the negatives may, for example, correspond to other times. The superscripts $+$ and $-$ identify positive

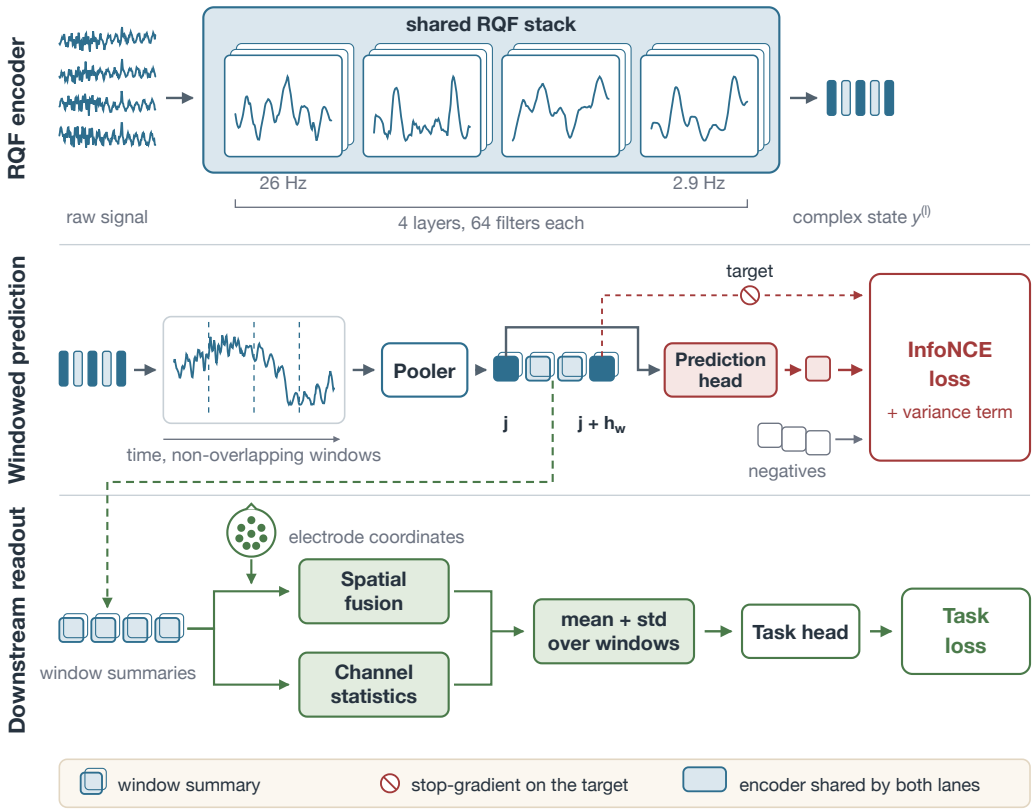


Figure 1: **Windowed self-supervised training and the multichannel task readout.** The pretraining and downstream stages share the encoder shown in the top row. **Top:** a single RQF stack processes each channel independently. The figure shows the 128-Hz EEG configuration at initialization, with four layers of 64 filters each. The front panel of each layer shows the state magnitude of one filter driven by a synthetic EEG-like probe input. The maximum initial pole frequency halves from layer to layer; the illustrated filters span 26 to 2.9 Hz. **Middle:** the state sequence is divided into non-overlapping windows. A learned pooler summarizes each window using the real and imaginary components of its mean and final state. The prediction head uses the summary of window j to estimate the summary of window $j + h_w$. One such pair is shown; training uses every window for which a target h_w windows later is available. The later summary serves as the positive target and is detached from gradient computation. Other window summaries in the batch serve as negatives. A variance penalty accompanies the InfoNCE objective. **Bottom:** the window summaries enter two branches. One projects and combines them across channels using spatial bases computed from electrode coordinates; the other computes their mean, standard deviation, and maximum across channels. The outputs are concatenated, then summarized by their mean and standard deviation over windows before entering the task head. Speech uses a single-channel readout. Dimensions, horizons, and optimization settings are in Table 10.

and negative targets, respectively. With $\mathcal{A}$ the set of sources in a batch and β an inverse temperature, the loss is the InfoNCE objective [11]:

$$\mathcal{L}_{\mathrm{NCE}} = -\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}\log\frac{\exp(\beta\,\mathrm{sim}_a^{+})}{\exp(\beta\,\mathrm{sim}_a^{+})+\sum_{k=1}^{K}\exp(\beta\,\mathrm{sim}_{a,k}^{-})}, \tag{2}$$

where sim_a^{+} is the cosine similarity between $\hat{\mathbf{t}}_a$ and $\mathbf{t}_a^{+}$; $\mathrm{sim}_{a,k}^{-}$ is the cosine similarity between $\hat{\mathbf{t}}_a$ and the kth negative. Targets are detached from gradient computation, and the prediction head is discarded after pretraining. The two objectives differ in how the source and target representations are constructed and how the prediction horizon is measured.

Table 1: Downstream classification tasks. "Example" denotes the unit assigned a label; each TUAB segment inherits its recording's label.

Benchmark	Signal	Example	Classes	Metric
TUAB	Scalp EEG	100 s segment	2: normal, abnormal	AUROC
BrainTreebank	Intracranial EEG	3 s segment	2: sentence onset, no speech	AUROC
ISRUC-SG1	Scalp EEG	30 s epoch	5: W, N1, N2, N3, REM	B-ACC
BCI-IV-2a	Scalp EEG	4 s trial	4: left hand, right hand, feet, tongue	B-ACC
Speech Commands	Audio	1 s clip	10: spoken command words	Accuracy

Windowed future prediction. The windowed objective predicts between summaries of the state sequence. We divide the top-layer states into fixed, non-overlapping windows and summarize each one into a vector $\mathbf{t}_j$: the speech configuration uses a parameter-free mean, whereas the multichannel configurations use a learned pooler that combines the mean real and imaginary activity in the window with its final state, followed by a linear projection and layer normalization. The source is the summary of window j and the target is the summary of window $j + h_w$, i.e., $\mathbf{s}_a = \mathbf{t}_j$ and $\mathbf{t}_a^+ = \mathbf{t}_{j+h_w}$, where the horizon h_w is an integer number of windows. Negatives are other window summaries in the batch, from the same input or from other inputs. A regression configuration instead minimizes the mean-squared error between the prediction head's estimate and the true future summary and needs no negatives; the BrainTreebank diagnostic in Appendix A uses this regression loss.

Dense per-timestep prediction. The dense objective uses the encoder state at an anchor timestep n as the source, without pooling states over a window. Splitting this state into its real and imaginary components gives $\mathbf{s}_n = [\mathrm{Re}\,\mathbf{y}_n^{(L)}\,;\,\mathrm{Im}\,\mathbf{y}_n^{(L)}]$. The target is built from the states in a future interval of $N_w \geq 1$ samples that begins $\Delta \geq 1$ samples after the anchor, so Δ is the prediction horizon in samples. Specifically, the target is *future log-power*, computed from the mean power of the encoder's states over that interval:

$$\mathbf{g}_n \;=\; \log\!\Big(\epsilon \;+\; \tfrac{1}{N_w}\textstyle\sum_{u=n+\Delta}^{n+\Delta+N_w-1} \big|\mathbf{y}_u^{(L)}\big|^2\Big), \tag{3}$$

taken elementwise over filters, with ϵ a small constant. The target $\mathbf{t}_n^+$ is $\mathbf{g}_n$ standardized per coordinate, and the negatives are the log-power targets of other anchors.

Multichannel composition. Multichannel recordings add a spatial component without altering the temporal prediction task. During pretraining, neither the encoder nor the prediction head receives electrode identity or position; spatial information enters only through the downstream readout in the bottom row of Figure 1. Appendix B gives the pooler and readout construction, anchor sampling, target calibration, the per-configuration negative exclusion policies, and the dataset-specific windows, horizons, and loss constants.

5 Evaluation design

The evaluation addresses three questions: 1) whether the complete RQF system can match the performance range of much larger encoders; 2) when pretraining improves performance with the model architecture held fixed; and 3) how the source and target representations used during pretraining affect downstream performance.

On TUAB, we evaluate abnormal-recording detection with different amounts of labeled training data, using patient-disjoint splits following BIOT [25]. Appendix A describes the test set. For sentence-onset detection on BrainTreebank (BTB) [8], we pretrain on BaRISTA's 17-session partition and evaluate using chronological splits within its seven sessions held out from pretraining [30]. We evaluate windowed pretraining on ISRUC-SG1 for five-class sleep staging [9]. For cross-corpus transfer, we pretrain the encoder on a subset of the TUH EEG Corpus (TUEG) and evaluate on the four-class motor imagery task in BCI-IV-2a [10]. Table 1 summarizes the classification tasks, including the supplementary Speech Commands benchmark reported in the appendix.

Throughout, *transfer* refers to reusing the pretrained encoder weights for a supervised downstream task. For all benchmarks except BCI-IV-2a, this transfer is from self-supervised pretraining to a

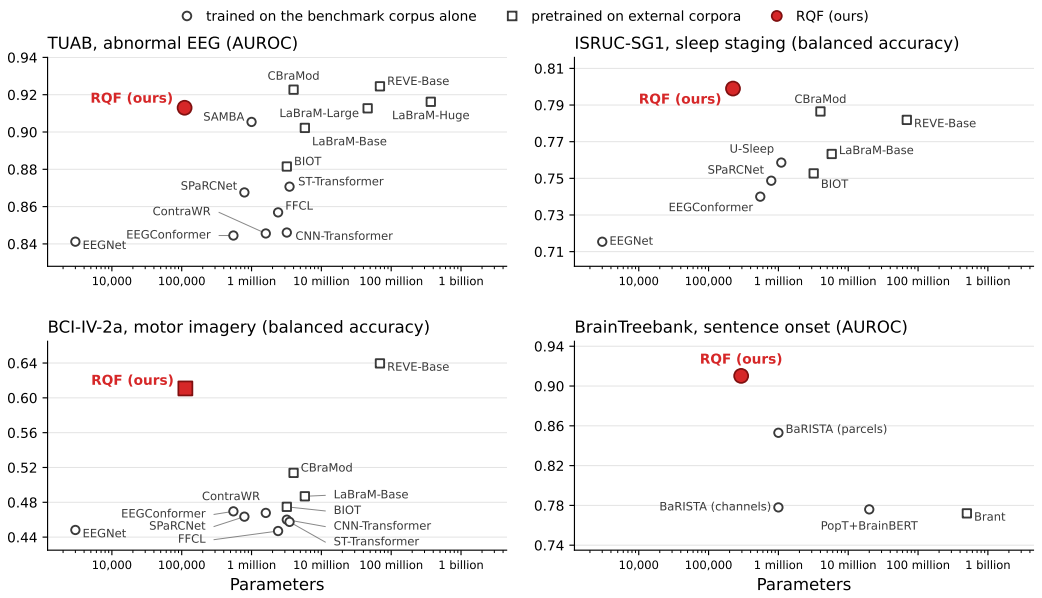


Figure 2: **Test performance versus parameter count on four benchmarks.** Each panel uses the metric named in its title. Red marks our strongest pretrained RQF configuration; circles denote training on the benchmark corpus alone, and squares denote pretraining on external corpora. Reference values come from published benchmarks and source papers. The BrainTreebank reference points use BaRISTA's chronological-split evaluation (its Appendix Table 16) [30]. For ISRUC, the RQF model and the baselines reported in CBraMod use context from 20 consecutive epochs. Appendix A provides all values and protocol details, including the RQF result when each epoch is classified independently and the protocol reported for REVE.

supervised task within the same corpus. The TUEG-to-BCI-IV-2a experiment instead tests *cross-corpus transfer*, with pretraining and task-specific data from different corpora.

We compare each pretrained encoder with a control that starts from the same initial weights but skips pretraining (*paired init.*). During supervised training, the encoder is either kept fixed (*frozen*) or updated jointly with the readout (*fully fine-tuned*, full-FT). For each benchmark, the test set is reserved for final evaluation.

Appendices A and B provide the full evaluation protocols, metric definitions, and complete results, including the supplementary Speech Commands experiments.

6 Results

Compact RQF systems perform competitively with larger models. On TUAB, windowed pretraining followed by full fine-tuning reaches 0.9130 ± 0.0036 test AUROC with 110,002 real-scalar parameters. This exceeds the reported AUROCs of SAMBA and LaBraM-Base (0.9054 and 0.9022), which use 9 and 53 times as many parameters, respectively. It is also within half an AUROC point of LaBraM-Huge (0.9162), which uses 3,355 times as many parameters. CBraMod and REVE-Base score 1.0 and 1.2 points higher while using 36 and 627 times as many parameters, respectively (Figure 2 and Table 2). In balanced accuracy, our model exceeds LaBraM-Large (0.8226) and falls below SAMBA, LaBraM-Huge, CBraMod, and REVE-Base (0.8258 to 0.8315).

On BCI-IV-2a, we use Euclidean alignment to reduce differences between recording sessions. Full fine-tuning of TUEG-pretrained weights reaches 0.6108 ± 0.0089 balanced accuracy. CBraMod scores 9.7 points lower with 35 times as many parameters, while REVE-Base scores 2.9 points higher with 612 times as many (Figure 2 and Appendix Table 8). With the same preprocessing, our RQF trained without SSL performs within 0.9 balanced-accuracy points of the versions pretrained on BCI-IV-2a or TUEG.

Table 2: Test performance on TUAB segments. Published values are from CBraMod, REVE, and SAMBA [2, 3, 21]. Published 10-s results use inputs sampled at 200 Hz; SAMBA and our models use 100-s segments sampled at 128 Hz. Our values are mean±std over three downstream seeds. The paired-initialization control starts from the corresponding pretrained model's initial weights and skips self-supervised pretraining. Parameter counts follow the real-scalar convention of Table 9. Per metric, bold marks the best reported mean; blue bold marks the best RQF mean.

Model	Params	Context	n	AUROC	B-ACC
EEGNet	3k	10 s/200 Hz	5	0.8412 ± .0031	0.7642 ± .0036
EEGConformer	550k	10 s/200 Hz	5	0.8445 ± .0038	0.7758 ± .0049
SPaRCNet	790k	10 s/200 Hz	5	0.8676 ± .0012	0.7896 ± .0018
ContraWR	1.6M	10 s/200 Hz	5	0.8456 ± .0074	0.7746 ± .0041
CNN-Transformer	3.2M	10 s/200 Hz	5	0.8461 ± .0013	0.7777 ± .0022
FFCL	2.4M	10 s/200 Hz	5	0.8569 ± .0051	0.7848 ± .0038
ST-Transformer	3.5M	10 s/200 Hz	5	0.8707 ± .0019	0.7966 ± .0023
BIOT	3.2M	10 s/200 Hz	5	0.8815 ± .0043	0.7959 ± .0057
LaBraM-Base	5.8M	10 s/200 Hz	5	0.9022 ± .0009	0.8140 ± .0019
LaBraM-Large	46M	10 s/200 Hz	5	0.9127 ± .0005	0.8226 ± .0015
LaBraM-Huge	369M	10 s/200 Hz	5	0.9162 ± .0016	0.8258 ± .0011
CBraMod	4.0M	10 s/200 Hz	5	0.9227 ± .0011	0.8289 ± .0022
REVE-Base	69M	10 s/200 Hz	n.r.	**0.9245 ± .0013**	**0.8315 ± .0014**
SAMBA-100	1.0M	100 s/128 Hz	n.r.	0.9054 ± .0044	0.8264 ± .0017
Ours — paired init., full-FT	110k	100 s/128 Hz	3	0.9092 ± .0040	0.8212 ± .0117
Ours — windowed SSL, frozen	110k	100 s/128 Hz	3	0.9017 ± .0021	0.8103 ± .0082
Ours — windowed SSL, full-FT	110k	100 s/128 Hz	3	**0.9130 ± .0036**	**0.8230 ± .0024**

Table 3: TUAB label efficiency for windowed SSL and its paired initialization control. Columns give the fraction of training labels used. Values are test AUROC, reported as mean±std over three downstream seeds from a fixed checkpoint. All configurations use the 110k-parameter model of Table 9. Differences are computed from unrounded values. Blue bold marks the best mean at each label fraction; all models are RQF.

Configuration	1%	5%	10%	25%	100%
Full-FT, windowed SSL	0.7365 ± .0148	0.8193 ± .0111	0.8439 ± .0189	**0.8856 ± .0039**	**0.9130 ± .0036**
Full-FT, paired init.	0.6404 ± .0142	0.8109 ± .0135	0.8336 ± .0073	0.8689 ± .0028	0.9092 ± .0040
Full-FT difference (pp)	+9.61	+0.84	+1.03	+1.68	+0.38
Frozen, windowed SSL	**0.7498 ± .0122**	**0.8253 ± .0109**	**0.8489 ± .0162**	0.8717 ± .0071	0.9017 ± .0021
Frozen, paired init.	0.6173 ± .0135	0.7961 ± .0051	0.8199 ± .0094	0.8332 ± .0013	0.8636 ± .0024
Frozen difference (pp)	+13.24	+2.91	+2.89	+3.86	+3.81

For ISRUC-SG1, the RQF system reaches 0.7990 ± 0.0087 balanced accuracy with 223,733 real-scalar parameters, including the context module and classifier. This score exceeds those of all comparison models, including REVE-Base, which uses 308 times as many parameters (Figure 2 and Appendix Table 5). The model uses context from 20 consecutive 30-second EEG segments to predict the sleep stage of each segment. A variant that classifies each segment independently achieves 0.7746 ± 0.0134 balanced accuracy (Appendix Table 5). The same contextual model trained without SSL reaches 0.7953 ± 0.0038, so on this benchmark the result reflects the architecture more than pretraining.

Pretraining provides its largest gains when supervision is limited. We use TUAB to measure how the benefit of pretraining changes with the amount of labeled data. We initialize two copies of the same encoder with identical random weights and pretrain one with the windowed objective. We then add the readout and train both networks on the same labeled subset, updating both the encoder and the readout. With only 1% of the training labels, the pretrained model reaches 0.7365 ± 0.0148 AUROC, compared to 0.6404 ± 0.0142 without pretraining, a gain of 9.6 points. With all labels available, the gain falls to 0.4 points. The largest benefit therefore occurs when labeled training data are scarce (Table 3).

Table 4: BrainTreebank test AUROC under our chronological-block split. For each seed, we average the results from seven sessions with two folds each, then report mean±std across three seeds. All rows use the same 293k-parameter RQF model; blue bold marks the best mean in each column.

Configuration	Frozen	Full fine-tuning
Paired init.	$0.8439 \pm .0196$	$0.8709 \pm .0094$
Dense InfoNCE SSL	$0.8958 \pm .0064$	$0.8796 \pm .0080$
Windowed InfoNCE SSL	**$0.9102 \pm .0093$**	**$0.8997 \pm .0053$**

Frozen-encoder performance favors windowed pretraining. On BrainTreebank, we compare the two pretraining objectives by freezing the pretrained encoder and training a task-specific readout. This tests whether the learned representation already contains information useful for sentence-onset detection. Windowed pretraining, which predicts between window summaries, reaches 0.9102 ± 0.0093 AUROC. Dense pretraining, which predicts future log-power from individual encoder states, reaches 0.8958 ± 0.0064. Both exceed the score of the same encoder frozen at its initial random weights (0.8439 ± 0.0196). The mean scores favor windowed prediction, although the difference is not statistically significant (Table 4).

Updating the encoder along with the readout does not necessarily improve performance. After windowed pretraining, mean AUROC is 0.9102 with the encoder frozen and 0.8997 with full fine-tuning. The decrease is reproduced across three training seeds, but is not statistically significant across sessions and folds. Full fine-tuning therefore provides no clear improvement over keeping the encoder frozen in this setting.

7 Discussion

Summary. We developed self-supervised learning for RQF encoders by training them to predict targets derived from their own future states. Each RQF learns a phase-prior frequency and a damping parameter, building oscillatory dynamics directly into the encoder. Prediction operates either between window summaries or from individual states to future log-power, without input masking, a quantizer, or a separate target encoder. The prediction horizon must be long enough to make copying the present ineffective, yet short enough that the future remains predictable. Across abnormal-recording detection, sentence-onset detection, sleep staging, and motor imagery, pretrained RQF systems with 110k to 293k parameters reach the performance range of models with 1M to 500M parameters (Figure 2). Pretraining helps most when labels are scarce or the encoder is frozen, with smaller gains under full fine-tuning. Future-state prediction thus provides a practical way to pretrain compact encoders for scalp EEG and intracranial recordings, with the same approach also applying to speech (Appendix A).

Limitations. Most experiments use corpus-specific pretrained weights, target geometry, and readouts. Cross-corpus transfer is tested only from TUEG to BCI-IV-2a, where the RQF trained without SSL performs within 0.9 balanced-accuracy points of both pretrained versions after Euclidean alignment. Comparisons with foundation models do not isolate the oscillatory bias from tokenization, masking, or scale, and several differ in input context, evaluation folds, or parameter accounting (Appendix A). We do not measure variability across pretraining runs; each pretrained configuration uses a single checkpoint. Finally, RQF poles are input-independent, and channels are encoded separately without electrode positions, leaving spatial structure to the downstream readout; the costs of these restrictions remain unmeasured.

Future work. Three direct extensions are as follows. 1) Pretrain a single RQF encoder across scalp and intracranial corpora and evaluate it with frozen readouts on every task. 2) Scaling our models beyond the 30k to 140k parameters used here to test how performance increases. 3) Comparing RQF encoders with masked prediction and patch-based Transformers with future-state prediction, with matched parameter counts, would separate the contributions of the architecture, the objective, and scale. Two further extensions remain open. 1) Test the addition of input-dependent poles, akin to the selective state transitions of Mamba and CaMBrain [32, 24]. 2) The RQF encoder is causal, but our readouts pool over the entire segment; pairing the encoder with a causal readout would allow the system to be evaluated for online decoding.

AI USE STATEMENT

Generative AI tools were used, and the authors reviewed every AI-assisted output. Among the uses for which disclosure is required, these tools helped implement the methods, helped clean and reformat the corpora, gave feedback on the design of the experiments, helped refine hypotheses, and assisted in interpreting results. They were not used to generate synthetic data sets, to develop theoretical models or conceptual frameworks, to formulate mathematical claims, to supply ingredients for proofs or to write them, or to assist with translation, and qualitative or thematic data analysis does not arise in this work.

Verification was carried out as follows. Every number reported here was read from the recorded artifacts of the run that produced it rather than from model output, each mean and standard deviation was recomputed from the individual seeds, and the claims made in the text and in the figure captions were checked line by line against the implementation they describe. The authors take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only human recordings that other groups collected, de-identified, and released, each under the terms of its own data use agreement: the Temple University Hospital EEG corpora TUAB and TUEG [7], BrainTreebank [8], ISRUC-SG1 [9], BCI Competition IV dataset 2a [10], and Speech Commands [33]. No recordings were collected for this study and no human subject interacted with it, so the approvals that govern these data are the ones documented in the publications that introduced each corpus.

Two limitations bear on how the results should be read. A model pretrained on the archive of a single hospital inherits the demographic composition of that archive, and our evaluation does not measure how performance varies across patient groups. The abnormality and sleep staging numbers are benchmark results obtained under community protocols; they are not evidence of clinical validity, and none of the models reported here is intended for diagnostic use.

REPRODUCIBILITY STATEMENT

The companion repository named at the start of Appendix B holds the shared-channel RQF encoder, the learned window pooler, both self-supervised objectives (windowed InfoNCE and dense future log-power prediction), and the coordinate-aware spatial readout. Its end-to-end examples cover TUAB and Speech Commands under both objectives, the TUAB label-budget subsets, and BCI-IV-2a with Euclidean alignment and in-domain or cross-corpus pretraining. Each covers preprocessing, pretraining, the paired-initialization control, and both frozen and full fine-tuning. The repository also provides the TUEG pretraining configuration. The BrainTreebank and ISRUC-SG1 pipelines, the TUEG preprocessing, the datasets, and the pretrained weights are not part of the current release. Appendix B gives the model configurations with their parameter counts, the frequency initialization schedule, the preprocessing applied to each corpus, the construction of both pretext tasks, and the pretraining and downstream training protocols. Appendix A defines the metrics and reports the complete set of results behind every main-text comparison, together with the differences between our evaluation procedures and those of the published models we quote. The rule that governs our own results is stated in Appendix B: the downstream epoch is selected on validation data, the test set is evaluated once at that epoch, and, unless stated otherwise, results are reported as the mean and standard deviation over three downstream seeds from one fixed pretraining checkpoint.

REFERENCES

[1] Wei-Bang Jiang, Li-Ming Zhao, and Bao-Liang Lu. Large brain model for learning generic representations with tremendous EEG data in BCI. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2405.18765.

[2] 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, 2025. URL https://arxiv.org/abs/2412.07236.

[3] 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. *Advances in Neural Information Processing Systems*, 38:22541–22577, 2025. doi: 10.52202/085713-0760. arXiv:2510.21585.

[4] Shivang Rawat, Mirko Morello, Flaviano Morone, and David J. Heeger. Prospective coding improves learning in deep continuous-time recurrent networks. *arXiv preprint arXiv:2609.04134*, 2026. URL https://arxiv.org/abs/2609.04134.

[5] David J Heeger and Klavdia O Zemlianova. A recurrent circuit implements normalization, simulating the dynamics of V1 activity. *Proceedings of the National Academy of Sciences*, 117 (36):22494–22505, 2020. doi: 10.1073/pnas.2005417117.

[6] Thomas Donoghue, Matar Haller, Erik J Peterson, Paroma Varma, Priyadarshini Sebastian, Richard Gao, Torben Noto, Antonio H Lara, Joni D Wallis, Robert T Knight, et al. Parameterizing neural power spectra into periodic and aperiodic components. *Nature neuroscience*, 23(12): 1655–1665, 2020.

[7] Neural Engineering Data Consortium. Temple University EEG Corpus: Downloads. Temple University. URL https://isip.piconepress.com/projects/nedc/html/tuh_eeg/index.shtml. Accessed September 5, 2026.

[8] Christopher Wang, Adam Yaari, Aaditya K. Singh, Vighnesh Subramaniam, Dana Rosenfarb, Jan DeWitt, Pranav Misra, Joseph R. Madsen, Scellig Stone, Gabriel Kreiman, Boris Katz, Ignacio Cases, and Andrei Barbu. Brain treebank: Large-scale intracranial recordings from naturalistic language stimuli. In *Advances in Neural Information Processing Systems*, volume 37, pages 96505–96540, 2024. doi: 10.52202/079017-3060. Datasets and Benchmarks Track; arXiv:2411.08343.

[9] 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.

[10] Michael Tangermann, Klaus-Robert Müller, Ad Aertsen, Niels Birbaumer, Christoph Braun, Clemens Brunner, Robert Leeb, Carsten Mehring, Kai J. Miller, Gernot 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.

[11] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. URL https://arxiv.org/abs/1807.03748.

[12] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html.

[13] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021. doi: 10.1109/TASLP.2021.3122291.

[14] 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*, pages 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.

[15] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024. URL https://arxiv.org/abs/2404.08471.

[16] Sarthak Yadav and Zheng-Hua Tan. Audio mamba: Selective state spaces for self-supervised audio representations. In *Interspeech 2024*, pages 552–556, 2024. doi: 10.21437/Interspeech.2024-1274. arXiv:2406.02178.

[17] Yuan Gong, Cheng-I Jeff Lai, Yu-An Chung, and James Glass. SSAST: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10699–10709, 2022. doi: 10.1609/aaai.v36i10.21315.

[18] Temesgen Mehari and Nils Strodthoff. Towards quantitative precision for ECG analysis: Leveraging state space models, self-supervision and patient metadata. *IEEE Journal of Biomedical and Health Informatics*, 27(11):5326–5334, 2023. doi: 10.1109/JBHI.2023.3310989.

[19] M. A. Al-Masud and Nils Strodthoff. Pretraining strategies and scaling for ECG foundation models: A systematic study. arXiv preprint, 2026. arXiv:2605.12241.

[20] Feng Wu, Harsh Deep, Eric Lehman, Sanyam Kapoor, Guoshuai Zhao, Rahul G. Krishnan, Gari Clifford, and Li-wei H. Lehman. SL-S4Wave: Self-supervised learning of physiological waveforms with structured state space models. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=km0xS3jZeO. arXiv:2606.19888.

[21] Jiazhen Hong, Geoffrey Mackellar, and Soheila Ghane. SAMBA: Toward a long-context EEG foundation model via spatial embedding and differential mamba. arXiv preprint, 2025. arXiv:2511.18571.

[22] Danaé Broustail, Anna Tegon, Thorir Mar Ingolfsson, Yawei Li, and Luca Benini. LuMamba: Latent unified mamba for electrode topology-invariant and efficient EEG modeling. arXiv preprint, 2026. arXiv:2603.19100.

[23] Jia Hong Puah, Sim Kuan Goh, Ziwei Zhang, Zixuan Ye, Chow Khuen Chan, Kheng Seang Lim, Si Lei Fong, Kok Sin Woon, and Cuntai Guan. EEGDM: EEG representation learning via generative diffusion model. arXiv preprint, 2025. arXiv:2508.14086.

[24] Abhilash Durgam, Nyle Siddiqui, Jeffrey A. Chan-Santiago, Qiushi Fu, Elakkat D. Gireesh, and Mubarak Shah. CaMBrain: Real-time, continuous EEG inference with causal state space models. arXiv preprint, 2026. arXiv:2605.28792.

[25] Chaoqi Yang, M Brandon Westover, and Jimeng Sun. BIOT: Biosignal transformer for cross-data learning in the wild. *Advances in Neural Information Processing Systems*, 36:78240–78260, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f6b30f3e2dd9cb53bbf2024402d02295-Abstract.html.

[26] Anna Tegon, Thorir Mar Ingolfsson, Xiaying Wang, Luca Benini, and Yawei Li. FEMBA: Efficient and scalable EEG analysis with a bidirectional mamba foundation model. In *2025 47th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1–7, 2025. doi: 10.1109/EMBC58623.2025.11252697.

[27] Daoze Zhang, Zhizhang Yuan, Yang Yang, Junru Chen, Jingjing Wang, and Yafeng Li. Brant: Foundation model for intracranial neural signal. In *Advances in Neural Information Processing Systems*, volume 36, pages 26304–26321, 2023. doi: 10.52202/075280-1144.

[28] Christopher Wang, Vighnesh Subramaniam, Adam Uri Yaari, Gabriel Kreiman, Boris Katz, Ignacio Cases, and Andrei Barbu. BrainBERT: Self-supervised representation learning for intracranial recordings. In *International Conference on Learning Representations*, 2023. arXiv:2302.14367.

[29] Geeling Chau, Christopher Wang, Sabera Talukder, Vighnesh Subramaniam, Saraswati Soedarmadji, Yisong Yue, Boris Katz, and Andrei Barbu. Population transformer: Learning population-level representations of neural activity. In *International Conference on Learning Representations*, 2025. arXiv:2406.03044.

[30] Lucine L. Oganesian, Saba Hashemi, and Maryam M. Shanechi. BaRISTA: Brain scale informed spatiotemporal representation of human intracranial neural activity. In *Advances in Neural Information Processing Systems*, volume 38, pages 185289–185322, 2025. doi: 10.52202/085713-5573. arXiv:2512.12135.

[31] David J Heeger. Theory of cortical function. *Proceedings of the National Academy of Sciences*, 114(8):1773–1782, 2017. doi: 10.1073/pnas.1619788114.

[32] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[33] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018. URL https://arxiv.org/abs/1804.03209.

[34] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33: 6696–6707, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4a5876b450b45371f6cfe5047ac8cd45-Abstract.html.

[35] David W Romero, Anna Kuzina, Erik J Bekkers, Jakub M Tomczak, and Mark Hoogendoorn. CKConv: Continuous kernel convolution for sequential data. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2102.02611.

[36] David W Romero, Robert-Jan Bruintjes, Erik J Bekkers, Jakub M Tomczak, Mark Hoogendoorn, and Jan C van Gemert. FlexConv: Continuous kernel convolutions with differentiable kernel sizes. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2110.08059.

[37] Yassine El Ouahidi et al. REVE official implementation: BCI-IV-2a preprocessing. GitHub repository, commit 06a7059, 2026. URL https://github.com/elouayas/reve_eeg/blob/06a7059a07c3dabd80aee60c3dbc1eca4bdbe1c7/preprocessing/preprocessing_bciciv2a.py.

[38] He He and Dongrui Wu. Transfer learning for brain–computer interfaces: A Euclidean space data alignment approach. *IEEE Transactions on Biomedical Engineering*, 67(2):399–410, 2020. doi: 10.1109/TBME.2019.2913914.

A COMPLETE RESULTS BY BENCHMARK

This appendix reports the complete results behind the main-text comparisons. To measure the effect of SSL, we train a second model from the same starting weights without pretraining; tables label this control *paired init.* Published values use the evaluation procedures of their source papers. Relevant differences from our procedures are described alongside the results.

Metrics. Binary tasks report AUROC, the probability that a positive example is scored higher than a negative one, and balanced accuracy (B-ACC), the mean recall across classes. ISRUC and BCI-IV-2a use B-ACC as their primary metric; Speech Commands (SC10) reports accuracy.

A.1 TUAB

TUAB contains clinical scalp EEG recordings from Temple University Hospital, labeled as normal or abnormal [7]. Our preparation contains 2,076 patients for training and validation and 253 for testing. We feed the EEG signals to the RQF encoder and compare the windowed and dense prediction objectives of Section 4, then train a readout to distinguish normal from abnormal recordings. Each 100-second segment inherits the label of its recording, and all metrics are computed over segments.

Full-data comparison. The windowed objective leads dense prediction more clearly in AUROC than in balanced accuracy. In a separate comparison with both prediction horizons fixed at two seconds and the encoders frozen, the two objectives reach 0.8978 ± 0.0089 and 0.8888 ± 0.0027 AUROC, respectively; their balanced accuracies differ by only 0.19 points, within the variation across seeds. The windowed results in Table 2 use a one-second prediction horizon. The test set matches the official evaluation partition in size and class balance. The available metadata do not allow us to verify this correspondence patient by patient.

TUAB label efficiency. Pretraining helps most when only 1% of the training labels are available. With more labels, the benefit is smaller when the encoder is fully fine-tuned; frozen pretrained encoders outperform their paired-initialization controls at every label budget. Each comparison starts from the same weights and uses the same validation and test patients.

A.2 ISRUC-SG1

ISRUC-Sleep contains overnight recordings collected at the Sleep Medicine Centre of Coimbra University Hospital; subgroup I has one recording from each of 100 adults undergoing sleep assessment [9]. We use six EEG channels from 99 participants, excluding one recording that lacks two required channels, with 79/10/10 participants for training, validation, and testing. We pretrain the RQF encoder with windowed future prediction (Section 4) and fine-tune it to assign one of five sleep stages to each 30-second epoch.

Epoch context. Adding neighboring epochs improves balanced accuracy from 0.7746 ± 0.0134 to 0.7990 ± 0.0087 (Table 5). Both RQF models first encode each 30-second epoch independently. The epoch-local model predicts from that representation alone; the contextual model passes the 20 representations through separate forward and reverse one-layer RQFs before predicting all 20 labels. The published baselines also use 20-epoch context, but CBraMod's evaluation combines the epoch representations with a shared one-layer Transformer [2]. Our contextual model uses RQFs for both epoch encoding and context, so both components differ from those baselines. Windowed pretraining adds little on this benchmark: the paired-initialization controls reach 0.7953 ± 0.0038 with context and 0.7753 ± 0.0031 without it.

Table 5: ISRUC-SG1 five-class sleep-staging test balanced accuracy. Published baselines except REVE-Base come from CBraMod Table 7 and use its shared Transformer over 20 epochs [2]. Their parameter counts omit the Transformer, whereas ours include the complete downstream model. REVE reports 20-epoch sequence prediction without specifying its context module, so the table marks its context as reported [3]. Our split excludes subject 8, whose F3 and F4 channels are missing, so we train on 79 subjects instead of 80; validation and test subjects match CBraMod's. Our results are mean±std over three seeds. Blue bold marks the best mean overall, achieved by RQF.

Model	Params	Epoch context	B-ACC
EEGNet	3k	20×30 s	0.7154 ± .0121
EEGConformer	550k	20×30 s	0.7400 ± .0133
SPaRCNet	790k	20×30 s	0.7487 ± .0075
ContraWR	1.6M	20×30 s	0.7402 ± .0126
CNN-Transformer	3.2M	20×30 s	0.7363 ± .0087
FFCL	2.4M	20×30 s	0.7277 ± .0182
ST-Transformer	3.5M	20×30 s	0.7381 ± .0205
DeepSleepNet	21M	20×30 s	0.7419 ± .0144
U-Sleep	1.1M	20×30 s	0.7586 ± .0116
BIOT	3.2M	20×30 s	0.7527 ± .0121
LaBraM-Base	5.8M	20×30 s	0.7633 ± .0102
CBraMod	4.0M	20×30 s	0.7865 ± .0110
REVE-Base	69M	reported: 20×30 s	0.7819 ± .0078
Ours — contextual RQF, full-FT	224k	20×30 s	**0.7990 ± .0087**
Ours — epoch-local RQF, full-FT	198k	one 30 s epoch	0.7746 ± .0134
Ours — contextual RQF, paired init., full-FT	224k	20×30 s	0.7953 ± .0038
Ours — epoch-local RQF, paired init., full-FT	198k	one 30 s epoch	0.7753 ± .0031

A.3 Speech Commands

Speech Commands is a collection of short spoken words recorded by volunteers, released by Google for keyword-recognition research [33]. We use the ten-command subset of 34,975 exactly one-second clips, with a class-stratified 70/15/15 training, validation, and test split. The RQF encoder receives the audio waveform and learns through the same windowed or dense prediction objective as in Section 4; the downstream task identifies which of the ten commands (yes, no, up, down, left, right, on, off, stop, go) each clip contains.

Dense and windowed prediction. On these non-neural signals, both objectives approach the strongest published compact models after full fine-tuning. With the encoder frozen, windowed prediction retains a 1.0-point advantage over dense prediction, while both remain far above their paired-initialization control (Table 6).

Table 6: SC10 ten-class test accuracy on the exact 34,975 one-second clips. Our models use raw waveforms and the same per-timestep readout. Our values are mean±std over three downstream seeds. Bold marks the best reported accuracy; blue bold marks the best RQF accuracy.

Model	Params	Input	n	Accuracy
Neural CDE [34]	89k	MFCC	5	0.898 ± .025
CKConv [35]	100k	MFCC	n.r.	0.9527
FlexConv [36]	239k	MFCC	n.r.	**0.9773**
Ours — paired init., frozen	69.3k	raw	3	0.6731 ± .0043
Ours — paired init., full-FT	69.3k	raw	3	0.9666 ± .0034
Ours — dense SSL, frozen	69.3k	raw	3	0.9408 ± .0009
Ours — windowed SSL, frozen	69.3k	raw	3	0.9503 ± .0012
Ours — dense SSL, full-FT	69.3k	raw	3	**0.9723 ± .0029**
Ours — windowed SSL, full-FT	69.3k	raw	3	0.9717 ± .0010

Phase-inclusive target diagnostic. In a separate SC10 diagnostic with an EMA target encoder, we compared power prediction alone with an additional phase-increment prediction head. Both frozen encoders were evaluated with the same diagnostic probe. The phase head reached 0.885 validation retrieval accuracy, but adding it reduced downstream validation macro recall from 0.8086 to 0.7045 with 5% of the labels and from 0.9040 to 0.8800 with all labels. This comparison covers one seed and does not test other ways of representing phase.

A.4 BRAINTREEBANK

BrainTreebank records intracranial EEG from ten patients undergoing epilepsy monitoring at Boston Children's Hospital while watching movies [8]. We use 17 sessions for SSL and seven other sessions for downstream evaluation, with seven participants shared between these sets. We apply the windowed and dense objectives of Section 4 to the intracranial signals and train the downstream readout to detect sentence onsets. This binary task uses three-second segments: positive segments are centered on a word that begins a sentence, and negative segments contain no speech at all [30]. With the encoder frozen, windowed pretraining reaches 0.9102 ± 0.0093 AUROC, compared with 0.8439 ± 0.0196 for paired initialization (Table 4).

Evaluation protocol and published reference points. We use chronological blocks with validation intervals separating training and test data, evaluating two folds per session. We therefore take published reference values from BaRISTA's chronological evaluation (Appendix Table 16), which averages five folds per session [30]. In our folds, training and test windows never overlap and are separated by at least 172 seconds. The evaluation sessions are held out from SSL, although some patients contribute recordings to both pretraining and evaluation.

Controlled comparisons. The controlled matrix separates the pretext objective from the way the encoder is used downstream. Table 4 uses the same architecture and seven sessions with two folds each while varying pretraining and whether the encoder is frozen. Table 7 tests the same transfer modes after regression pretraining; raw-voltage and STFT probes calibrate the task without a learned encoder. We retrain the controlled baselines on our folds.

Both non-neural baselines use L2-regularized logistic regression. The inputs are either standardized voltage samples or STFT magnitudes computed with a 512-sample Hann window, 384-sample overlap, and frequencies up to 150 Hz; regularization is selected on validation data.

Table 7: Controlled BrainTreebank baselines and windowed-MSE diagnostic across the same seven sessions with two folds each (test AUROC). Logistic results are deterministic fits, mean±std across these 14 results; MSE results are mean±std over three downstream seeds and were not part of the preregistered significance tests. Blue bold marks the highest mean, achieved by RQF.

Configuration	Test AUROC
Logistic regression, raw voltage	0.7672 ± .1336
Logistic regression, STFT magnitude	0.8870 ± .0678
Windowed MSE SSL, frozen	**0.9032 ± .0075**
Windowed MSE SSL, full-FT	0.9019 ± .0119

The preregistered tests use results from seven sessions with two folds each at downstream seed 0 as paired observations. Among comparisons involving a learned representation, frozen windowed SSL exceeds paired-initialization full fine-tuning at $p = 0.017$; its margin over the STFT probe is $p = 0.058$. The tests do not resolve the windowed–dense difference ($p = 0.30$), or the frozen–full-FT ordering under windowed ($p = 0.14$) or dense ($p = 1.0$) pretraining. With MSE pretraining, frozen and fully fine-tuned encoders obtain similar mean AUROC. The higher average scores of frozen encoders after InfoNCE pretraining are therefore not confirmed by these tests.

A.5 BCI-IV-2A

BCI-IV-2a contains scalp EEG recorded at Graz University of Technology from nine participants imagining movements of the left hand, right hand, feet, or tongue [10]. We use all 5,184 trials, with

five participants for training and two each for validation and testing, following REVE's released split [37]. We apply windowed future prediction (Section 4) either to the training participants' EEG or to a 3,768-hour subset of TUEG, the larger clinical EEG archive from Temple University Hospital [7], before fine-tuning for motor-imagery classification.

Cross-corpus comparison. We use Euclidean alignment to reduce differences between recording sessions, estimating a whitening matrix for each session without using its labels [38]. With this preprocessing, our RQF trained directly on labeled data performs within 0.9 balanced-accuracy points of the models pretrained on BCI-IV-2a or TUEG. Omitting alignment lowers the accuracy of both pretrained models. Published values come from REVE's benchmark. REVE applies Euclidean alignment in its own preprocessing [3], so we compare its result with our aligned results.

Table 8: BCI-IV-2a four-class motor-imagery test balanced accuracy. Published values come from REVE Table 12 [3], which does not report repeat counts. REVE applies Euclidean alignment (EA) in its own preprocessing and does not state the preprocessing behind its baseline rows. Our rows use full fine-tuning and report mean±std over three seeds. Bold marks the best reported mean; blue bold marks the best RQF mean.

Model	Params	Pretraining data	n	B-ACC
EEGNet	3k	none	n.r.	0.4482 ± .0094
EEGConformer	550k	none	n.r.	0.4696 ± .0106
SPaRCNet	790k	none	n.r.	0.4635 ± .0117
ContraWR	1.6M	none	n.r.	0.4678 ± .0125
CNN-Transformer	3.2M	none	n.r.	0.4600 ± .0108
FFCL	2.4M	none	n.r.	0.4470 ± .0143
ST-Transformer	3.5M	none	n.r.	0.4575 ± .0145
BIOT	3.2M	PREST+SHHS	n.r.	0.4748 ± .0093
LaBraM-Base	5.8M	2,535 h	n.r.	0.4869 ± .0085
CBraMod	4.0M	>9,000 h TUEG	n.r.	0.5138 ± .0066
REVE-Base	69M	61,415 h	n.r.	**0.6396 ± .0095**
Ours — paired init. (EA), full-FT	113k	none	3	0.6021 ± .0249
Ours — TUEG SSL (cross-corpus), full-FT	113k	3,768 h TUEG subset	3	0.5495 ± .0213
Ours — TUEG SSL (cross-corpus, EA), full-FT	113k	3,768 h TUEG subset	3	**0.6108 ± .0089**
Ours — BCI-IV-2a SSL, full-FT	113k	3.2 h BCI-IV-2a train	3	0.5026 ± .0148
Ours — BCI-IV-2a SSL (EA), full-FT	113k	3.2 h BCI-IV-2a train	3	0.6088 ± .0308

B Implementation and reproducibility details

The companion repository is available at https://anonymous.4open.science/r/rqf-ssl-A984.

B.1 Model configurations

Parameter accounting. RQF parameters are complex-valued, so we count each complex parameter as two real scalars. Table 9 applies this convention to the complete downstream models used in the paper. Totals include the per-filter time constants τ (256 scalars for encoders of width 64 and 512 for width 128).

Multiscale frequency initialization. Deeper RQF layers are initialized at progressively lower frequencies, so the stack begins with filters spanning several temporal scales. In the four-layer encoders used here, the upper frequency limit is multiplied by 1, $1/2$, $1/4$, and $1/8$ from the first to the fourth layer. This schedule affects only the starting values: during training, every layer can learn frequencies throughout the common range $[1/T,\ f_s/2]$.

Shared RQF configuration. All experiments use the same encoder core: a pure RQF stack with zero-order-hold discretization, split-ReLU activations, and a prospective two-tap input. The same stack is applied to every channel. It contains no skip connections, dropout, or normalization between layers. We initialize $\tau = 2\Delta t$; it stays fixed during pretraining and in the TUAB and BrainTreebank

Model	Encoder	Summary/pooler	Readout	Total
Speech classifier (d4/64)	33.7k	—	35.6k	69.3k
EEG windowed (d4/64)	33.7k	16.6k	59.8k	110k
ISRUC epoch-local windowed (d4/64)	33.7k	16.6k	147k	198k
ISRUC contextual windowed (d4/64)	33.7k	16.6k	173k	224k
BCI-IV-2a (d4/64)	33.7k	16.6k	62.6k	113k
iEEG windowed (d4/128)	133k	65.9k	94.1k	293k

Table 9: Real-scalar parameter counts for the downstream models, rounded to three significant digits (k denotes thousands). Components and totals are rounded independently. The ISRUC contextual model adds two one-layer RQFs over the epoch representations.

evaluations, while full fine-tuning on BCI-IV-2a, ISRUC, and Speech Commands also updates it. We constrain damping to $(0, 1)$ and use a mixing-weight gain of 0.5 at initialization. Table 10 gives the dataset-specific widths, prediction geometry, and training settings for TUAB and BrainTreebank. Input preprocessing is dataset-specific and described below. Layer normalization in the window pooler and downstream readout does not use dataset-wide statistics. Dense targets are standardized separately from a training-only calibration subset.

Coordinate-aware readout. Multichannel tasks use the spatial readout of Section 4, which allows one model to operate on different electrode montages. Each integer-millimetre coordinate indexes three learned 200×32 axis embeddings, covering positions from -100 to $+99$ mm. Their sum passes through a $32 \to 32 \to 16$ MLP to score 16 spatial basis functions across the observed channels. The corresponding value vectors have width 32, giving projections of $64 \to 512$ for EEG and $128 \to 512$ for iEEG. We concatenate the basis-weighted sums with the channelwise mean, standard deviation, and maximum, then summarize these features over time by their mean and standard deviation before classification. These choices produce the readout counts in Table 9.

	TUAB	BrainTreebank
Signal	scalp EEG	intracranial EEG
Encoder depth / width	4 / 64	4 / 128
Sampling rate f_s	128 Hz	2,048 Hz
Sequence context T	100 s	3 s
Window / step	1 s / 1 s (100 windows)	62.5 ms / 62.5 ms (48 windows)
Window summary	complex mean $\oplus$ final state $\to$ linear + LayerNorm	
Summary width	64	128
SSL horizon h_w	1 window (1 s); 2 windows (2 s) for the dense comparison	1 window (62.5 ms)
Inverse temperature β	10	
Sampled negatives	256, sampled uniformly from other window summaries in the batch (any channel)	
Loss weights, global / variance-hinge	1.0 / 0.04	
Pretraining optimizer	AdamW, lr 10^{-4}, weight decay 5×10^{-2}, grad-clip 1.0	
Pretraining epochs / batch	30 / 8	20 / 2
Downstream epoch cap / batch	20 / 4	20 / 2 (grad. accum. 2)
Downstream learning rates	2×10^{-4} pretrained, 2×10^{-3} new parameters; warmup-exponential; wd 10^{-2}	
Model selection	validation only; one post-selection test read per cell	

Table 10: Windowed TUAB and BrainTreebank configurations; shared settings span both columns. ISRUC and SC10 use separate schedules (Section B.4), as does BCI-IV-2a.

B.2 Datasets and preprocessing

- **Speech Commands (10-class)** [33]. We retain the 34,975 exactly one-second clips and use a fixed, class-stratified 70/15/15 split at the clip level. Waveforms are sampled at 16 kHz and standardized using training-set statistics.
- **TUAB (EEG, abnormal detection).** We derive the TCP-16 bipolar montage from the 19 referential 10–20 channels and resample it to 128 Hz. No notch filter is applied. Recordings are divided into non-overlapping 100-second segments and normalized independently as described below. Following the BIOT convention [25], we divide the non-test patients 80/20 into training and validation sets with seed 0. The held-out test cohort and its provenance are described in Appendix A.
- **ISRUC-SG1 (EEG, five-class sleep staging).** We use the NeMAR v1.0.1 release and the sleep-stage labels provided by scorer 1. We use six EEG channels (F3–A2, C3–A2, O1–A2, F4–A1, C4–A1, and O2–A1) at 200 Hz and score 30-second epochs [9]. Each recording is filtered at 0.3–35 Hz with a fifth-order Butterworth filter and at 50 Hz with a notch filter ($Q = 30$), before division into epochs. Subjects 1–80, 81–90, and 91–100 define the training, validation, and test splits; I008 is excluded because F3 and F4 are absent from the released recording. SSL uses only the training subjects. For downstream evaluation, we divide each recording into non-overlapping blocks of 20 consecutive labeled epochs. Unknown-stage epochs interrupt a block, and incomplete blocks are discarded without padding, leaving 87,100 epochs. Both RQF configurations are evaluated on these same retained epochs; only the contextual model uses information from neighboring epochs.
- **BCI-IV-2a (EEG, four-class motor imagery).** We retain all 5,184 trials and use the 22 EEG channels, excluding EOG. Signals are common-average referenced and filtered with a causal fifth-order Butterworth bandpass at 0.3–40 Hz. We extract the four-second cue interval and resample it from 250 to 200 Hz. Subjects 1–5, 6–7, and 8–9 provide 2,880 training, 1,152 validation, and 1,152 test trials, respectively. The downstream readout uses two non-overlapping two-second windows per trial. When Euclidean alignment is used, its transformation is estimated separately from the unlabeled trials of each recording session, including validation and test sessions, and applied to all trials in that session. In-domain SSL sees only training-subject trials. We transfer only the encoder; the downstream learned window pooler and spatial readout are initialized anew.
- **BrainTreebank (iEEG, sentence onset).** We use three-second segments at 2,048 Hz, retaining the channels specified by the clean-channel metadata and excluding contacts labeled GRID or VENT. Signals are Laplacian-referenced using adjacent contacts and notch-filtered at 60 Hz and its harmonics through 360 Hz ($Q = 30$). Each segment is standardized separately within each channel. Electrode coordinates are provided to the downstream spatial readout.

Segment normalization. To reduce sensitivity to isolated amplitude extremes, TUAB segments are scaled independently in each channel: $x \mapsto x/\big(q_{95}(|x|) + 10^{-8}\big)$, where the percentile is computed over time. This operation neither centers nor clips the signal.

B.3 Pretext construction

Pretext-task geometry. The two objectives supply different representations to the prediction head. Windowed prediction uses summaries of non-overlapping windows as both sources and targets. Dense prediction uses individual encoder states as sources and future log-power as targets. In the dense case, N_w in Eq. (3) controls only the interval used to measure future power; it does not pool the source representation.

For a window with sample indices $\mathcal{I}_j$, the mean summary is

$$\bar{\mathbf{t}}_j = \frac{1}{|\mathcal{I}_j|} \sum_{n \in \mathcal{I}_j} [\operatorname{Re} \mathbf{y}_n^{(L)} \,;\, \operatorname{Im} \mathbf{y}_n^{(L)}].$$

Writing $m_j = \max \mathcal{I}_j$ and $\tilde{\mathbf{t}}_j = [\operatorname{Re} \mathbf{y}_{m_j}^{(L)} \,;\, \operatorname{Im} \mathbf{y}_{m_j}^{(L)}]$ for the final state in the window, the learned pooler produces

$$\mathbf{t}_j = \mathrm{LN}\big(\mathbf{P}[\bar{\mathbf{t}}_j \,;\, \tilde{\mathbf{t}}_j]\big)\,.$$

The parameter-free deterministic configuration uses $\mathbf{t}_j = \bar{\mathbf{t}}_j$ directly.

Family	Input $[T \times C]$; rate	Encoder	Windows	Dims.	Horizon
Speech dense	$16{,}000 \times 1$; 16 kHz	d4/64	— (N_w=400)	128/64	1,600 smp = 100 ms
Speech windowed-det	$16{,}000 \times 1$; 16 kHz	d4/64	40×400	128	4 win. = 100 ms
EEG windowed+pooler	$12{,}800 \times 16$; 128 Hz	d4/64	100×128	64	2 win. = 2,000 ms
EEG dense	$12{,}800 \times 16$; 128 Hz	d4/64	— (N_w=128)	128/64	256 smp = 2,000 ms
iEEG windowed+pooler	$6{,}144 \times C$; 2,048 Hz	d4/128	48×128	128	1 win. = 62.5 ms

Table 11: Pretext geometry used for each family. Windowed horizons are in windows (h_w), dense horizons in samples (Δ). "Dims." gives the prediction-head input and target dimensions, written as input/target when they differ. The executed EEG windowed-vs-dense pair is horizon-matched at 2 s (2 windows vs 256 samples).

Contrastive objective parameters. Every contrastive configuration uses cosine similarity with inverse temperature $\beta = 10$, 256 sampled negatives, and detached targets. Windowed prediction on TUAB, BrainTreebank, ISRUC, and SC10 samples other window summaries from the batch with replacement. BCI-IV-2a and its TUEG pretraining source instead sample 256 distinct summaries from the same channel in other examples. The source and positive windows themselves never overlap. Single-channel dense prediction excludes nearby anchors whose target intervals overlap, while multichannel dense prediction draws from the same channel in other examples.

Dense targets are standardized coordinate by coordinate using 256 seeded training examples, with a standard-deviation floor of 10^{-6} and $\epsilon = 10^{-6}$ in Eq. (3). The windowed objective adds a variance-hinge penalty of weight 0.04 to discourage collapsed summaries; the dense objective uses only the contrastive loss.

Prediction-horizon selection. The horizon should be long enough that copying the present is ineffective, but short enough that the future remains predictable. For the matched TUAB windowed–dense comparison, a preregistered diagnostic selects the shortest candidate for which a persistence predictor has copy-$R^2 \leq 0.50$ while a ridge-affine predictor retains $R^2 \geq 0.20$. On the 100-second recordings, copy-R^2 falls from 0.99 at 125 ms to 0.14 at 2 s, while affine-R^2 remains 0.70 at 2 s; the rule therefore selects 2 s. The headline TUAB configuration retains the earlier one-second horizon. A validation-only comparison on ISRUC likewise selects two one-second windows over one.

The 100-ms speech horizon predates this rule. A later controlled screen found that shortening it to 12.5 ms made the target largely copyable and reduced frozen-encoder accuracy by 7–8 points, supporting the same choice qualitatively.

B.4 Training and model selection

Pretraining schedules. TUAB and BrainTreebank use AdamW with learning rate 10^{-4}, weight decay 5×10^{-2}, and gradient clipping at 1.0; SC10 uses learning rate 10^{-3} and weight decay 10^{-2}. BCI-IV-2a and TUEG pretraining use AdamW with peak learning rate 10^{-4} and weight decay 0.05. The learning rate warms up for five and two epochs, respectively, then follows a cosine schedule ending at 5% of its peak. Weight decay applies to all optimized parameters, including phase-prior frequency and damping $(\boldsymbol{\omega}, \boldsymbol{\gamma})$. Downstream evaluation always starts from the final pretraining epoch, so validation performance never selects the SSL checkpoint. ISRUC pretraining runs for 30 epochs on individual 30-second training epochs, using one-second windows and a two-second horizon. SC10 pretraining runs for 50 epochs. For BCI-IV-2a, in-domain SSL runs for 60 epochs using eight non-overlapping half-second windows per trial and a one-second prediction horizon; the cross-corpus rows use a terminal TUEG checkpoint trained for 30 epochs on a 3,768-hour subset.

Downstream training. Encoder-only transfer reuses the pretrained encoder and initializes the downstream pooler and readout anew. Encoder-and-pooler transfer also reuses the pretrained window pooler. The SSL prediction head is discarded in both cases.

The multichannel tasks use AdamW with weight decay 10^{-2} on all optimized parameters, including $(\boldsymbol{\omega}, \boldsymbol{\gamma})$ when trained. Pretrained parameters receive learning rate 2×10^{-4} and newly initialized

readout parameters receive 2×10^{-3}; both rates warm up linearly for five epochs and then decay exponentially. The frozen TUAB and BrainTreebank windowed evaluations keep both the encoder and transferred pooler fixed, optimizing only the spatial readout and classifier at the latter rate. ISRUC transfers the encoder alone and trains the pooler, spatial readout, context RQFs, and classifier for up to 50 epochs, selecting by validation Cohen's κ. BCI-IV-2a also uses a 50-epoch cap. SC10 follows its separate community recipe for up to 200 epochs: AdamW at 10^{-2} with no weight decay and $\epsilon = 10^{-4}$.

Reporting and model selection. Unless stated otherwise, results are reported as mean $\pm$ standard deviation over three downstream training runs with different random seeds, all starting from one fixed checkpoint. This measures variation in downstream training, conditional on that checkpoint. We select the downstream epoch using validation data and evaluate the test set once at that epoch.

Low-label evaluation. At label budgets of 1, 5, 10, and 25 percent, we construct nested TUAB patient subsets using sampling seed 0, retaining both classes at every budget. The 1% condition contains 16 training patients and 279 segments. Larger budgets retain the previously selected patients; validation and test patients remain unchanged. Each pretrained encoder is compared with its byte-identical initialization before SSL, under both frozen and full fine-tuning, with three downstream seeds per condition.