wHuBERT: Layer Selection for Iterative Masked-Prediction Pre-training on Low-Resolution Wearable Biosignals

Anonymous authors
Paper under double-blind review

Abstract

Applications for wearable biosignals increasingly use the foundation model paradigm, where models are pre-trained without labels. HuBERT, originally developed for speech, requires two pre-training phases, the first one with coarse clustered pseudo-labels, and the second one with clusters derived from the representations of an intermediate layer from the first phase. The selection of the correct layer for the second iteration is crucial for a well-performing model. However, accurate performance metrics can only be obtained after the two pre-training phases and one fine-tuning phase for a downstream task. This represents a large commitment of compute before any concrete metric is available. One clustering layer also forces a trade-off between tasks and undercuts the "train once, apply anywhere" premise. We measure the layer selection effect on wHuBERT, a Conformer encoder pre-trained on photoplethysmography and accelerometry sampled at 25 Hz, the rate consumer wearables use. We propose a cheap probe-based protocol, applied to every layer at regular checkpoints, that selects the best-scoring layer for the given task. Across 2 independent runs, the final-checkpoint best-scoring layer for sleep staging lies just below the top, rarely at the final layer. We also computed 7 label-free statistics, including estimates of effective rank, but show they are not reliable for layer selection. We verify these results on two public sleep-stages corpora. We also fine-tune and distill an iteration-2 encoder and both the teacher and student score above the baseline from a consumer device. The protocol thus offers low-cost layer selection for foundation models on low-resolution biosignals.

1 Introduction

Consumer wearables record photoplethysmography (PPG) and accelerometry (ACC) continuously, while expert labels for the same recordings, such as polysomnography-scored sleep stages, are scarce and expensive. Self-supervised pre-training learns an encoder from the unlabeled recordings, and each downstream task then adapts it with few labels. Foundation models for wearable biosignals built this way now exist (Abbaspourazad et al., 2024; Pillai et al., 2025).

Many of their training recipes come from speech, also a long, mostly unlabeled time series. One widely used recipe is masked prediction: the model hides spans of its input and learns to predict what was hidden (Baevski et al., 2020). HuBERT (Hsu et al., 2021) predicts discrete targets obtained by offline clustering, and refines those targets in two iterations. Iteration 1 trains an encoder to predict clusters of simple signal features. Iteration 2 clusters the hidden features of the iteration-1 encoder at one intermediate layer, the *clustering layer*, and trains a new encoder to predict those assignments. Iteration 2's targets can therefore encode only what the clustering layer holds. We adapt HuBERT because it refines its own targets without labels.

Which layer to cluster is a free parameter. HuBERT clusters layer 6 of 12 (Hsu et al., 2021), a choice not tested across tasks (Chen et al., 2025). The layers of a speech encoder hold acoustic, phonetic and word content at different depths (Pasad et al., 2021). Targets drawn from one layer therefore favor one kind of content, yet one pre-trained encoder is meant to serve every downstream task. A direct

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.

comparison of layers requires pre-training a further iteration from each, as Chen et al. (2025) do. A new modality lacks even a conventional value. The final layer is a natural default, but the final layers of masked-prediction speech encoders specialize toward the pretext task (Pasad et al., 2021). We measure which layer to cluster for wearable biosignals by probing the iteration-1 encoder, without training iteration-2 encoders from other layers.

Our model, wHuBERT ("w" for wearable), applies HuBERT's two iterations to PPG and ACC recorded at 25 Hz, a rate that consumer wearables use to save battery (Figure 1). To choose its clustering layer, we freeze the encoder at every saved checkpoint of pre-training and fit a linear sleep-stage probe on every layer. The probe uses a proprietary labeled corpus, the *internal corpus*. Because labels are scarce, we also ask whether label-free statistics computed on the same features agree with the probe. The protocol needs only saved checkpoints and one labeled task, so it can be applied to any iterative masked-prediction recipe, though our results cover only 25 Hz PPG and ACC. Our contributions are:

- **A probe-based protocol for choosing the clustering layer** (Section 3.2): the argmax, the layer with the best probe score, becomes the *selected layer* once it stops moving.
- **wHuBERT** (Section 3): HuBERT's iterative masked prediction adapted to 25 Hz PPG and ACC.
- **An evaluation of fine-tuning and of distillation into a deployed architecture** (Section 7): the fine-tuned iteration-2 encoder and its 692,251-parameter distilled student both score above the deployed baseline in mean macro F1.

Applying the protocol, we find that the argmax of one run moves down from the final layer across its saved checkpoints and settles (Section 5). In each of 2 independent pre-training runs, the linear probe on the internal corpus puts the final checkpoint's argmax in the band of layers 10 and 11 of 12, below the final layer. The band contains wHuBERT's configured clustering layer. RankMe (Garrido et al., 2023), a label-free estimate of effective rank, tracks the probe across checkpoints of one layer but does not recover the selected layer on the internal corpus, the pattern Aldeneh et al. (2025) report for speech. Combining the two sensors helps the probe most at the selected layer of the iteration-1 encoder. Section 6 tests whether the argmax stays near the top of the stack on public sleep corpora.

2 Related work

We relate the clustering layer to prior work on iterative targets, layer-wise analysis, label-free assessment, sleep staging and wearable foundation models.

Where iterative targets come from. Contrastive prediction of future content learns representations without labels (van den Oord et al., 2018), and wav2vec 2.0 established masked prediction over a learned discrete inventory for speech (Baevski et al., 2020). HuBERT replaced the learned inventory with offline clustering and made the procedure iterative (Hsu et al., 2021). In speech, Chen et al. (2025) compare clustering layers by pre-training a further iteration from each. Our encoder replaces the transformer blocks with Conformer blocks (Gulati et al., 2020) and adds a distance-correlation penalty (Székely et al., 2007) to a contrastive masked-prediction objective. We propose no new objective, and choose among layers by probing the iteration-1 encoder.

What encoder layers contain. Layer-wise probing of masked-prediction speech models shows that information rises through the stack and then falls near the top (Pasad et al., 2021). The shape of that profile differs across model families (Pasad et al., 2022). Frozen-representation benchmarks learn a weighting over layers rather than committing to one (Yang et al., 2021). Our offline clustering instead consumes one layer. Combining layers before clustering, or predicting clusters of multiple layers as Chen et al. (2025) do, lies outside our scope. We measure which layer to cluster, and when in training the choice settles.

Assessing a representation without labels. Effective rank measures dimensionality from the entropy of a matrix's normalized singular values (Roy & Vetterli, 2007), which for a covariance matrix are its eigenvalues. RankMe applies that idea to select self-supervised models without a downstream label (Garrido et al., 2023). The power-law decay exponent of the eigenspectrum has

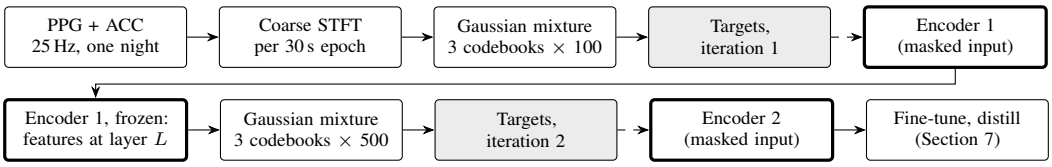

Figure 1: The two iterations of wHuBERT. Solid arrows produce targets; dashed arrows supervise an encoder, which predicts the targets at masked positions of its own input. Iteration 2 clusters the frozen iteration-1 encoder's features at one layer L to build its targets. This paper measures which L to use.

been proposed for the same purpose (Agrawal et al., 2022). The failure mode these statistics are built to detect is dimensional collapse, where embeddings occupy a smaller subspace than the one available (Jing et al., 2021). In speech, the rank of HuBERT embeddings correlates with downstream performance within a layer, but does not reliably predict the best-performing layer for a task (Aldeneh et al., 2025). Section 5.4 tests RankMe against the probe on the layers and checkpoints of a wearable encoder.

Sleep staging from wearable signals. Sleep staging is our downstream task, served well by supervised architectures over single-channel electroencephalography (Supratak et al., 2017; Phan et al., 2018; Perslev et al., 2019; Phan et al., 2021) and by models reading the wearable signals directly (Kotzen et al., 2022; Attia et al., 2024; Carter & Tarassenko, 2024; Topalidis et al., 2023). We use sleep staging only to select the clustering layer and to evaluate the resulting encoder.

Foundation models for wearable biosignals. Self-supervised pre-training has been applied to wearable photoplethysmography and electrocardiography at scale (Abbaspourazad et al., 2024), open photoplethysmography foundation models now exist (Pillai et al., 2025), and deployment work compresses such models for inference (Ni et al., 2025). We share their aim of one encoder, learned from unlabeled wearable data, for several downstream tasks. Our recipe differs in training iteratively against clustered targets, and so requires choosing a layer to cluster.

3 wHuBERT

wHuBERT applies HuBERT's iterative masked prediction to PPG and ACC. This section defines its loss and targets, the rule that selects its clustering layer, and its configuration.

3.1 Iterative masked prediction and the clustering layer

An encoder of 12 layers reads a sequence of frames, masked at a set M of positions, and computes hidden features $h_t^{(l)}$ at position t and layer l, with h_t the final layer's output. Each position has a target z_t, a cluster index with embedding e_{z_t}. At a masked position, the projected output Ph_t is compared by cosine similarity at temperature τ with its own target's embedding and with those of a set N_t of other masked positions in the same sequence. For one codebook, the loss at $t \in M$ is

$$\ell_t = -\log \frac{\exp\big(\cos(Ph_t, e_{z_t})/\tau\big)}{\sum_{u \in \{t\} \cup N_t} \exp\big(\cos(Ph_t, e_{z_u})/\tau\big)}, \tag{1}$$

and training minimizes its mean over the masked positions, summed over the three codebooks. Following w2v-BERT (Chung et al., 2021), the loss applies InfoNCE contrastive prediction (van den Oord et al., 2018) to masked codebook targets with sampled negatives.

The iterations differ in where the targets z_t come from. Iteration 1 clusters signal features. Iteration 2 freezes the converged iteration-1 encoder, clusters its features $h_t^{(L)}$ at one layer L, the clustering layer, and trains a fresh encoder against those assignments (Figure 1). The iteration-1 encoder has learned to predict its targets from context, so its features are expected to be richer than the signal features and to give better targets.

Both iterations need a codebook size and a set of frames to fit the mixture on, but only iteration 2 needs a clustering layer. wHuBERT was configured to cluster layer 10.

3.2 THE SELECTION RULE

We choose the clustering layer with a probe, a linear sleep-stage classifier fitted on the frozen iteration-1 encoder's features (Section 4). The *candidates* are the 12 encoder layers and the *pre-encoder space* that feeds the first of them, 13 in all. Iteration 2 would cluster the pre-encoder space if it used no encoder layer. At each saved checkpoint, the *argmax* is the candidate with the highest probe macro F1. The *selection rule* takes the argmax once it has stopped moving, and we call that candidate the *selected layer*. The stopping point is found retrospectively here, over all saved checkpoints, and the rule is not tested as an online stopping criterion. The rule waits because early in training the argmax is the final layer (Section 5).

3.3 IMPLEMENTATION

Front end and encoder. A recording is one night of PPG and three-axis ACC at 25 Hz, scored in 30 second windows called scoring epochs. Each channel becomes a log-power short-time Fourier transform (STFT), and both streams pass through convolutional feature encoders whose strides give one encoder frame per scoring epoch. A low-rank quadratic fusion layer combines the two encoded streams into the encoder width of 768, and its output is the pre-encoder space ("fused" in the tables). Figure 5 in Appendix I draws the whole model, and the appendix gives the dimensions and the fusion's rationale.

The encoder is a stack of 12 Conformer blocks (Gulati et al., 2020) with 12 attention heads, feed-forward width 3072, depthwise convolution kernel 31, and dropout 0.1. The Conformer blocks are the one change to HuBERT's encoder stack, and the front end, the regularizers and the Gaussian-mixture targets also differ from HuBERT. Every layer-wise result below is conditional on this configuration.

Objective. Masking follows HuBERT: spans of 10 frames start at about 8% of the frames, a masking probability of 0.8 divided by the span length, at least two per sequence. Equalizing the mask count across the batch leaves about 11% of frames masked (Appendix J). In Equation 1, N_t holds 100 negatives, P projects to 256 dimensions and $\tau = 0.1$. The loss adds a penalty on the squared distance correlation (Székely et al., 2007) between pairs of feature dimensions, weighted 10 in each sensor's feature-encoder output, 20 in the pre-encoder space and 0.01 in the first encoder layer. A second penalty acts on their squared Pearson correlation (Appendix M).

Targets and training. Both iterations pre-train without labels on a pool of internal recordings: the internal corpus of Section 7 without its test-split recordings, and further collections that outnumber it (Appendix L). Iteration 1's signal features are a coarser STFT, a 2048-point transform over a 60 second window at a 30 second hop. A diagonal-covariance Gaussian mixture (Dempster et al., 1977), fitted on the first half of the pool, clusters them into three codebooks of 100 components each, differing only in initialization. The original recipe uses k-means (Sculley, 2010). Iteration 2's mixture, three codebooks of 500 components, is fitted on the whole pool once, before the first step.

Both iterations are configured for 400,000 steps at 32 sequences on each of 4 accelerators, 128 per step. Each of lineage A's encoders (Section 5.3) has 330.8M parameters, front end included. Appendix M gives the optimizer, schedule and hardware.

4 EXPERIMENTAL SETUP: PROBES AND STATISTICS

The protocol measures what the selection rule of Section 3.2 needs, a probe score for every candidate at every saved checkpoint.

We freeze an encoder at a saved checkpoint and stream the internal corpus (Section 7), labeled with 4 sleep stages, through it once. A *sweep* repeats this pass at every saved checkpoint of one encoder. Layer indices are one-based, and the zero-based *clustering parameter* that selects layer n is $n - 1$.

Supervised readout. The probe is a closed-form ridge classifier on standardized frozen features (Appendix E), fitted per candidate, with the regularization strength chosen on the validation split from 8 values between 10^{-6} and 10. The probe is deliberately weak, a linear map that measures what is linearly decodable from each candidate under one shared protocol. Its scores therefore compare

the candidates as read by this probe, not their quality under every readout. Selection is on macro F1, since the stage distribution is skewed.

What the probe sees. Readouts are fitted on the training split and scored on the validation split, never on the test split, reserved for Section 7. Both splits are used in full, so every checkpoint sees the same 2,654,281 training and 356,877 validation frames in one fixed-seed order.

Because the probe is linear, a second, non-parametric *nearest-neighbor readout* checks whether the selection depends on the readout (Appendix F). It votes among the nearest of a uniform sample of 100,000 training frames, with the neighbor count tuned over ten values.

Label-free statistics. On the same frames we accumulate each candidate's mean, second moment and covariance in double precision. From them we compute the 7 spectral and dimension statistics of Table 4, defined with their variants in Appendix G.

Effective rank admits two conventions the literature does not always distinguish. RankMe (Garrido et al., 2023) is the exponentiated entropy of the *singular* values of the *uncentered* embedding matrix. Covariance effective rank (Roy & Vetterli, 2007) is the exponentiated entropy of the *eigenvalues* of the *centered* covariance. The two conventions differ in two ways at once: RankMe takes the square root of the spectrum and skips the centering, so a disagreement between them cannot be attributed to either change alone. Crossing the two changes gives four conventions, the corners of the effective-rank square (Appendix G).

Modality ablation. We zero one sensor at the input and *refit* the probe, so fitting and evaluation both see the ablated features. The measurement is how much label information remains decodable without that sensor, not how robust a clean-fitted probe is. Zeroing acts after the checkpoint's input scalers, so the zeroed sensor carries their per-frequency mean log-power (Appendix K).

5 Results

On the iteration-1 sweep, the protocol of Section 4 selects layer 10. Across 4 encoders of 2 lineages (Section 5.3), the final-checkpoint argmax lies in one two-layer band, and neither RankMe nor an untrained encoder recovers the selected layer.

5.1 The argmax across checkpoints

The iteration-1 sweep covers 39 checkpoints from step 20,000 to 400,000, 13 candidates each. Figure 2 and Table 2 in the appendix give the argmax at every checkpoint, its *trajectory*. The argmax starts at the final layer, migrates downward and settles on layer 10, where it stays from step 90,000, 22.5% of the way into the run. Layer 10 is the configured clustering layer, so here the probe recovers a choice already made. Iteration 2 reproduces the trajectory over 33 checkpoints to step 330,000, holding at layer 10 from step 70,000, and reaches 0.6231 macro F1 against iteration 1's 0.5683. Iteration 2 was trained to step 400,000, so a separate sweep, with an earlier probe harness and fewer label-free statistics, covers 40 checkpoints to that step. Its argmax holds at layer 10 from step 70,000 and ends at 0.6246 macro F1.

A sweep stopped at a tenth of the training budget would have named the final layer, whose effective rank is lowest at convergence (Section 5.2). The argmax settles before the probe score does. The score at the selected layer is within one percent of its peak only by step 240,000, 60% of the budget, and ends 0.0013 below its peak.

5.2 The final layer at convergence

At the final iteration-1 checkpoint the probe score rises monotonically with depth, peaks at layer 10, and then falls (Table 3 in the appendix). Layer 12 scores 0.5277 against 0.5683. Between layers 10 and 12 the covariance effective rank falls from 228.1 to 123.2, the participation ratio (Appendix G) from 97.2 to 29.1, and the mean squared off-diagonal correlation rises from 0.0084 to 0.0271. The final layer occupies a lower-dimensional, more redundant subspace than layer 10. The rank loss accompanies the lower probe score but is not shown to cause it.

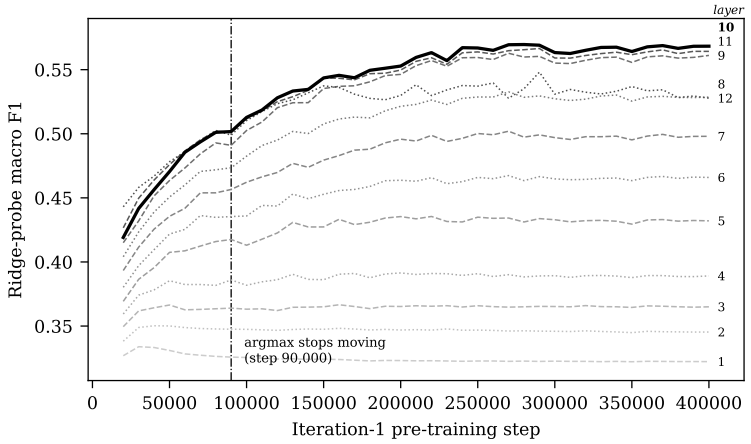

Figure 2: Ridge-probe macro F1 for every encoder layer at every iteration-1 checkpoint. The selected layer is drawn heavy, and the labels at the right edge give each curve's layer number. The argmax starts at the top of the stack, migrates down, and from the marked step does not move again within the sweep.

The same fall near the top appears in speech encoders (Pasad et al., 2021). The rank loss matters more here than for a task head: our clustering uses one layer and cannot down-weight it later, so clustering the final layer would draw the iteration-2 targets from its lower-rank subspace. Whether a lower-rank clustering layer degrades iteration 2 is not tested (Section 8).

5.3 What replicates across pre-training lineages

We call an independent iteration-1 pre-training run, together with everything trained on its targets, a *lineage*. Every sweep above comes from one lineage, A. We therefore probed a second lineage, B, at the final checkpoint of both iterations, with lineage A's probe settings on the same validation split. Lineage B shares lineage A's data and seed but not its configuration (Appendix L). Neither lineage-B run saved intermediate checkpoints.

Four findings hold on all 4 encoders (Table 6 in the appendix). First, the argmax lies at layer 10 or 11 of 12. Second, the final layer's covariance effective rank is below the argmax layer's in 4 of 4. Third, the layer at which zeroing a sensor costs most is the argmax layer or its neighbor. Fourth, RankMe selects a candidate the probe ranks 12th or worse of 13 on both lineage-B encoders, as in Section 5.4.

The exact layer does not replicate. Both lineage-B encoders put the argmax at layer 11, one above lineage A and one above lineage B's clustering layer, 10. Their argmax leads the final layer by as little as 0.0009 macro F1. The selection rule therefore locates a band, not a single layer.

5.4 Could the layer have been chosen without a probe?

A label-free statistic could replace the probe only if its argmax matched the probe's. At the final iteration-1 checkpoint, 2 of 7 do.

What each statistic would have chosen. The comparison is descriptive. Table 4 in the appendix gives the argmax of each statistic, oriented as in Appendix G, and the probe's rank of that choice. RankMe (Garrido et al., 2023) selects the pre-encoder space, which the probe ranks 13 of 13. Covariance effective rank (Roy & Vetterli, 2007) and participation ratio select the layer the probe selects. Covariance effective rank's agreement does not transfer. It misses the probe's layer on both lineage-B encoders, and on every public corpus of Section 6, all ACC-only, it selects the candidate the probe ranks last (Table 7 in the appendix). RankMe's disagreement does not transfer to BIDSleep either, where it selects the candidate the probe ranks 1 of 13.

Correlations with the probe. Table 5 in the appendix gives rank correlations against the probe in two regimes. Across checkpoints at the selected layer, the regime RankMe was proposed for, RankMe tracks the probe at Spearman 0.923. Across the layers of one converged checkpoint it reverses, to -0.533 over all candidates and -0.406 over the encoder layers alone. The permutation p-value is 0.064 for the reversal over all candidates, and 0.004 for covariance effective rank and participation ratio over the encoder layers. Neither the layers of one encoder nor the checkpoints of one run are independent samples, so these values are descriptive. The correlations corroborate Table 4, the primary comparison.

Two cautions. The across-layer correlation is confounded with depth. Given the probe-score profile of Section 5.2, any statistic peaking near layer 10 inherits a high rank correlation. The statistic with the strongest across-encoder-layer correlation still selects the pre-encoder space (Appendix C). Even across checkpoints, where label-free statistics track the probe, they do not resolve a fine-grained choice. Over the 21 checkpoints at or beyond step 200,000, every correlation falls to between 0.43 and 0.60 in absolute value. The training step, a baseline that always prefers the later checkpoint, scores 0.923 over the full range. A label-free statistic informs checkpoint choice only if it exceeds that baseline.

5.5 Control: an untrained encoder

A probe score that rises with depth could reflect pre-training or the architecture alone. An untrained encoder separates the two. It keeps the architecture, the fitted scalers and the STFT, with weights from one random initialization. Its probe score rises with depth, by +0.0086 macro F1 across the 12 encoder layers. The rise is small against the +0.2778 that pre-training adds to the iteration-2 encoder at the selected layer (0.3453 to 0.6231), and the gain is largest there, at layer 10. The untrained profile climbs monotonically to layer 12, so it lacks the interior peak on which the selection rule depends.

5.6 Modality ablation

The *fusion gain* is the probe's margin with both sensors over the better one alone. On lineage A's final iteration-1 checkpoint, at the pre-encoder space, removing ACC costs 0.0218 macro F1 and removing PPG 0.0036, so the fusion gain is only 0.0036. At layer 10 the same ablations cost 0.1602 and 0.1792, and the fusion gain is 0.1602, so there the probe score depends on both sensors under this ablation. On this encoder the fusion gain peaks at layer 10, the selected layer, and is lower at both layers above it (Figure 7 in the appendix). Table 6 in the appendix gives the layer of largest fusion gain on every encoder.

6 Transfer to public corpora, and the sampling rate

The selected layer comes from the internal corpus only. On two public corpora probed with ACC alone, the argmax stays near the top of the stack on the larger corpus and not on the smaller.

Setup. The probed encoder is lineage A's iteration-2 encoder at step 180,000, whose argmax on the internal corpus is layer 10. It is frozen, with its input scalers fitted on the pre-training pool (Appendix L). Only the probe is fitted, with the settings of Section 4 and the same 4-class mapping, on each corpus's participant-disjoint training split. As on the internal corpus, scores are on the validation split that also chooses the ridge strength and the argmax (Appendix E). The corpora differ in cohort and in sensors, so their absolute scores are not comparable, and the test is whether the *choice of layer* transfers.

Eligible corpora. We use two public corpora with wrist ACC and expert stage labels. PPG-DaLiA (Reiss et al., 2019) has no sleep labels. BIDSleep (Song, 2026) and sleep-accel (Walch, 2019) publish a derived heart rate at roughly 0.2 Hz rather than a cardiac waveform, so both enter with an all-zero PPG waveform before the checkpoint's STFT and scalers (Appendix H). BIDSleep is the larger, with 47 participants over 253 nights. Its training split holds 38 of them and 167,024 frames.

On the internal corpus, the ACC-only probe of the iteration-1 encoder selects the same layer as the full probe (Appendix K), while on this iteration-2 encoder it selects the final layer, at 0.4128 against

Table 1: Four-class sleep staging on the held-out test split (575 recordings). Metrics are computed per recording and reported as mean ± standard deviation across recordings, so the spread is between-recording variation rather than uncertainty in the mean. The baseline and the student share the deployed architecture (1.1 MB).

Model	Parameters	Macro F1 (%)	Cohen's κ	Balanced acc. (%)
Deployed baseline	692,251	70.04 ± 12.66	0.63 ± 0.14	78.48 ± 10.54
Fine-tuned encoder	186.1M	72.99 ± 11.87	0.67 ± 0.13	80.11 ± 9.19
Distilled student	692,251	72.11 ± 12.58	0.66 ± 0.14	78.63 ± 10.51

0.4038 at layer 10. The public corpora can therefore test whether the argmax stays near the top of the stack, but not the exact layer or the band.

The top of the stack holds on the larger corpus only. On BIDSleep the argmax is layer 11, at 0.5474 against 0.5423 at layer 10, while on sleep-accel it is layer 6, at 0.4646 against 0.3994 (Table 8). Of sleep-accel's validation frames, 16% carry only interpolated motion (Appendix H).

Cut to sleep-accel's size, BIDSleep keeps the first night of each of 31 participants, for 21,538 training frames against sleep-accel's 21,542. The argmax moves from layer 11 to layer 12, at 0.5392 against 0.5198 at layer 10. The internal ACC-only probe of this encoder also selects layer 12. At sleep-accel's size the argmax thus stays at the top of the stack, so size alone does not reproduce the sleep-accel result. The one cut changes the frame count, the participants and the nights per participant at once, so it does not isolate size from cohort composition. Without resampling intervals, the public results do not establish whether the band carries over to other corpora.

The sampling rate. Testing whether 25 Hz suffices needs a second encoder pre-trained at a higher rate on the same corpus, and the corpus exists only at 25 Hz. The frozen encoder cannot replace it: its front end expects 25 Hz, and resampling a faster signal to 25 Hz removes what the faster rate adds.

7 Downstream evaluation

For sleep staging, the downstream evaluation fine-tunes an iteration-2 encoder of a separate lineage, with k-means targets from layer 10 and a penalty other than lineage A's (Appendix L), and distills it for the device. The fine-tuned encoder and the distilled student score above the deployed baseline in mean macro F1. The comparison does not isolate the layer choice (Section 8).

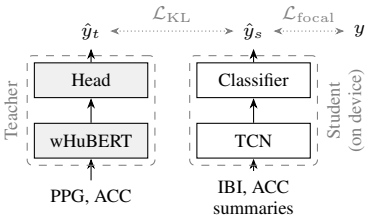

Figure 3: The teacher is the fine-tuned encoder with its classification head. The student is the deployed temporal convolutional network (TCN) on inputs the device already computes, trained by a KL term to the teacher's posterior and a focal loss to the labels.

Corpus. The internal corpus holds overnight recordings from 19 sleep studies across 9 wearable models, scored by experts into 4 stages: wake, light (N1 and N2), deep (N3) and rapid-eye-movement (REM). The split is by participant, 2,764 recordings for training, 339 for validation and 575 for test. Test recordings appear only here, but 60 of them share study and participant numbers with pre-training ring recordings (Appendix L).

Results. The fine-tuned encoder reaches 72.99 macro F1 and $\kappa = 0.67$ on the test split, against 70.04 and $\kappa = 0.63$ for the deployed baseline, a supervised classifier trained only on the internal corpus (Table 1, per-stage scores in Table 9 in the appendix).

The deployment target has the baseline's footprint, so we distill (Hinton et al., 2015) the fine-tuned encoder into the deployed architecture (Figure 3). We claim no novelty for the distillation recipe (Appendix N). The student reads inter-beat intervals (IBI), ACC summaries and timestamp features,

never the teacher's STFT. It scores 72.11 macro F1 and $\kappa = 0.66$. Its on-device latency and memory use were not measured. Without a paired interval, the differences in Table 1 are descriptive.

8 LIMITATIONS

The results rest on few lineages, one probe label and one sampling rate.

Few lineages, and a band rather than an index. Only lineage A contributes a *trajectory*, and its two sweeps are not independent, since iteration 2 trains on targets clustered from iteration 1. So the downward migration of the argmax and its settling rest on one independent run. Lineage B contributes only final-checkpoint argmaxes at its two iterations, one layer above lineage A's (Section 5.3).

Scope of the label-free result. With 13 candidates, the across-layer correlations are underpowered and confounded with depth (Section 5.4). Table 4 in the appendix covers one encoder. On every lineage, each effective-rank convention that takes the square root of the spectrum, as RankMe does, selects a candidate the probe ranks near last (Table 7 in the appendix).

One run per configuration. Lineages A and B share a seed but differ in configuration (Appendix L), so their spread mixes configuration with run-to-run variation. The downstream results are single runs without a paired interval, so their differences are descriptive. Probe scores carry no resampling intervals, so a small gap between adjacent layers does not establish which of them is better.

One probe label and one rate. The probe uses one label, sleep stages, on a corpus that exists only at 25 Hz. Whether the band holds for other labels or rates is not tested (Section 6).

No counterfactual iteration 2. No iteration-2 encoder is trained on targets clustered from another layer, so the evidence that the choice matters downstream rests on the probe and on the spectral gap between layers, not on a paired comparison of iteration-2 encoders.

9 CONCLUSION

We introduced wHuBERT, HuBERT adapted to 25 Hz PPG and ACC, and a protocol that chooses its clustering layer. The protocol probes every candidate of the iteration-1 encoder at every saved checkpoint and, retrospectively, takes the argmax once it stops moving, without pre-training an iteration-2 encoder from each candidate. At the final checkpoint of every lineage, the argmax lies below the final layer, in a band that contains the configured clustering layer. On the internal corpus, RankMe does not recover the selected layer.

The student distilled from the fine-tuned iteration-2 encoder scores above the deployed baseline in mean macro F1 (Table 1). Without a paired interval, the difference is descriptive. The student keeps the baseline's architecture, size and inputs, so deploying it changes only the weights. Distillation could thus bring foundation models to devices with tight computational constraints.

Future work can train an iteration-2 encoder from another layer to test whether the selected layer yields the best one, seek a label-free estimate of the fusion gain, and add age, sex and body mass index as downstream tasks.

REPRODUCIBILITY STATEMENT

Reproduction is limited because the pre-training pool and the internal corpus are proprietary and cannot be released. Section 3 and Appendix M specify the architecture, objective, masking, clustering and optimization. Section 4 specifies the probe protocol, including the training and validation splits it is fitted and scored on. Both splits are used in full, in one fixed-seed order. Section 6 probes one frozen encoder on public corpora. Checking those results needs that encoder's weights as well as the public data. The adapters that convert each of those corpora into the probe's input format are in the supplementary material, with their probe configurations and a self-check. Each adapter records the on-disk layout it was written against and states which sensor the corpus does not carry and is

therefore zeroed. It also records the corpus's published stage coding, its mapping to the 4 classes, and the handling of unscored epochs.

Three scripts turn the runs' own outputs into the paper's values. One reduces the probe sweeps, a second emits the tables and the prose macros, and a third emits the figures. A build gate re-runs all three and fails if any output changes. The gate catches a hand edit to a generated value but not an error in the analysis. Values that no run output in the repository holds, such as configuration constants and corpus counts, are recorded by hand, each with its source. The deployed baseline of Table 1 and the deployed architecture's size are transcribed from internal documentation.

ETHICS STATEMENT

This work analyzes overnight physiological recordings from human participants, collected under protocols approved by the responsible ethics committee, with participant consent. Recordings reach the modeling pipeline under pseudonymous study identifiers, and only aggregate results are reported.

Two risks remain. First, the cohorts are convenience samples from sleep studies and are not demographically representative, so neither the probe results nor the downstream scores should be read as evidence of uniform performance across populations. We report between-recording spread because it is large. Second, the models in this paper are consumer sleep-tracking components, not clinical instruments. The comparisons with expert-scored references are measurements against a reference standard and do not establish diagnostic validity.

AI USE STATEMENT

We used generative AI tools substantially in preparing this work. We separate the uses here because they carry different risks.

For *analysis and tooling*: the scripts that reduce the probe sweeps, compute the reported correlations, and emit every table, figure and prose number were written with AI assistance. Each output regenerates from the run artifacts, and a build gate compares checksums. The gate ties each value to its source and does not show that the analysis is correct.

For *writing*: the manuscript was drafted and edited with AI assistance, AI-generated reviews of the draft informed its revision, and the authors reviewed the text line by line. The AI-generated reviews commented on the methodology and on the interpretation of the results. The authors decided every resulting change. Generative AI was not used to design the experiments, to formulate the hypotheses or to generate data.

For *literature*: candidate related work was gathered with AI assistance, the step at which fabricated or misattributed citations can arise. Every reference is therefore resolved through a bibliographic API, each citing sentence's distinctive terms are checked against the cited work's text, and any sentence asserting another work's *result* carries a quotation located verbatim in that work. Some references are exempt from the automated text check, each for a recorded reason, such as being a dataset or an article that cannot be redistributed. The authors read each exempt work.

No AI tool generated the experimental results. The pre-training, the probe evaluations and the downstream training are real runs on real data. The authors take responsibility for all content, including anything produced with AI assistance.

REFERENCES

Salar Abbaspourazad, Oussama Elachqar, Andrew Miller, Saba Emrani, Udhyakumar Nallasamy, and Ian Shapiro. Large-scale Training of Foundation Models for Wearable Biosignals. In *International Conference on Learning Representations*, pp. 3158–3181, 2024.

Kumar K Agrawal, Arnab Kumar Mondal, Arna Ghosh, and Blake Richards. α-ReQ: Assessing Representation Quality in Self-Supervised Learning by Measuring Eigenspectrum Decay. *Advances in Neural Information Processing Systems 35*, pp. 17626–17638, 2022. doi:10.52202/068431-1281.

Zakaria Aldeneh, Vimal Thilak, Takuya Higuchi, Barry-John Theobald, and Tatiana Likhomanenko. Towards Automatic Assessment of Self-Supervised Speech Models using Rank. In *ICASSP 2025 -*

2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5, 2025. doi:10.1109/ICASSP49660.2025.10889651.

Shirel Attia, Revital Shani Hershkovich, Alissa Tabakhov, Angeleene Ang, Sharon Haimov, Riva Tauman, and Joachim A. Behar. SleepPPG-Net2: Deep learning generalization for sleep staging from photoplethysmography. *arXiv preprint arXiv:2404.06869*, 2024.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *arXiv preprint arXiv:2006.11477*, 2020. NeurIPS 2020.

Jonathan F. Carter and Lionel Tarassenko. wav2sleep: A Unified Multi-Modal Approach to Sleep Stage Classification from Physiological Signals. *arXiv preprint arXiv:2411.04644*, 2024. Machine Learning for Health (ML4H) 2024.

Li-Wei Chen, Takuya Higuchi, He Bai, Ahmed Hussen Abdelaziz, Shinji Watanabe, Alexander Rudnicky, Tatiana Likhomanenko, Barry-John Theobald, and Zakaria Aldeneh. Exploring Prediction Targets in Masked Pre-Training for Speech Foundation Models. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025. doi:10.1109/ICASSP49660.2025.10889924.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training. *arXiv preprint arXiv:2108.06209*, 2021.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 39(1): 1–22, 1977. doi:10.1111/j.2517-6161.1977.tb01600.x.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the Downstream Performance of Pretrained Self-Supervised Representations by Their Rank. In *International Conference on Machine Learning*, pp. 10929–10974, 2023.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented Transformer for Speech Recognition. *arXiv preprint arXiv:2005.08100*, 2020. Interspeech 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015.

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. *arXiv preprint arXiv:2106.07447*, 2021. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding Dimensional Collapse in Contrastive Self-supervised Learning. *arXiv preprint arXiv:2110.09348*, 2021. ICLR 2022.

Sunho Kim, Sungbin Im, and Taehyung Park. Characterization of Quadratic Nonlinearity between Motion Artifact and Acceleration Data and its Application to Heartbeat Rate Estimation. *Sensors*, 17(8):1872, 2017. doi:10.3390/s17081872.

Kevin Kotzen, Peter H. Charlton, Sharon Salabi, Lea Amar, Amir Landesberg, and Joachim A. Behar. SleepPPG-Net: a deep learning algorithm for robust sleep staging from continuous photoplethysmography. *arXiv preprint arXiv:2202.05735*, 2022. IEEE Journal of Biomedical and Health Informatics, 2023.

Juntong Ni, Saurabh Kataria, Shengpu Tang, Carl Yang, Xiao Hu, and Wei Jin. PPG-Distill: Efficient Photoplethysmography Signals Analysis via Foundation Model Distillation. *arXiv preprint arXiv:2509.19215*, 2025. NeurIPS 2025 TS4H.

T.M. Panicker, V.J. Mathews, and G.L. Sicuranza. Adaptive parallel-cascade truncated Volterra filters. *IEEE Transactions on Signal Processing*, 46(10):2664–2673, 1998. doi:10.1109/78.720369.

Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. Layer-wise Analysis of a Self-supervised Speech Representation Model. *arXiv preprint arXiv:2107.04734*, 2021. ASRU 2021.

Ankita Pasad, Bowen Shi, and Karen Livescu. Comparative layer-wise analysis of self-supervised speech models. *arXiv preprint arXiv:2211.03929*, 2022. ICASSP 2023.

Mathias Perslev, Michael Hejselbak Jensen, Sune Darkner, Poul Jørgen Jennum, and Christian Igel. U-Time: A Fully Convolutional Network for Time Series Segmentation Applied to Sleep Staging. *arXiv preprint arXiv:1910.11162*, 2019. NeurIPS 2019.

Huy Phan, Fernando Andreotti, Navin Cooray, Oliver Y. Chén, and Maarten De Vos. SeqSleepNet: End-to-End Hierarchical Recurrent Neural Network for Sequence-to-Sequence Automatic Sleep Staging. *arXiv preprint arXiv:1809.10932*, 2018.

Huy Phan, Kaare Mikkelsen, Oliver Y. Chén, Philipp Koch, Alfred Mertins, and Maarten De Vos. SleepTransformer: Automatic Sleep Staging with Interpretability and Uncertainty Quantification. *arXiv preprint arXiv:2105.11043*, 2021. IEEE Transactions on Biomedical Engineering.

Arvind Pillai, Dimitris Spathis, Fahim Kawsar, and Mohammad Malekzadeh. PaPaGei: Open Foundation Models for Optical Physiological Signals. In *International Conference on Learning Representations*, pp. 48230–48261, 2025.

Felipe Chaud Pinheiro and Cassio Guimaraes Lopes. A Low-Complexity Nonlinear Least Mean Squares Filter Based on a Decomposable Volterra Model. *IEEE Transactions on Signal Processing*, 67(21):5463–5478, 2019. doi:10.1109/TSP.2019.2932880.

Attila Reiss, Ina Indlekofer, Philip Schmidt, and Kristof Van Laerhoven. Deep PPG: Large-Scale Heart Rate Estimation with Convolutional Neural Networks. *Sensors*, 19(14):3079, 2019. doi:10.3390/s19143079.

Olivier Roy and M. Vetterli. The Effective Rank: A Measure Of Effective Dimensionality. Zenodo, 2007. doi:10.5281/zenodo.40328.

D. Sculley. Web-scale k-means clustering. *Proceedings of the 19th international conference on World wide web*, pp. 1177–1178, 2010. doi:10.1145/1772690.1772862.

Tzu-An Song. A Multi-Night Instantaneous Heart Rate and Accelerometry Dataset with EEG Sleep Stage Labels. PhysioNet, version 1.0.0, 2026. doi:10.13026/a0sy-7t69.

Akara Supratak, Hao Dong, Chao Wu, and Yike Guo. DeepSleepNet: a Model for Automatic Sleep Stage Scoring based on Raw Single-Channel EEG. *arXiv preprint arXiv:1703.04046*, 2017.

Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6), 2007. doi:10.1214/009053607000000505.

Pavlos Topalidis, Dominik P. J. Heib, Sebastian Baron, Esther-Sevil Eigl, Alexandra Hinterberger, and Manuel Schabus. The Virtual Sleep Lab—A Novel Method for Accurate Four-Class Sleep Staging Using Heart-Rate Variability from Low-Cost Wearables. *Sensors*, 23(5):2390, 2023. doi:10.3390/s23052390.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018.

Olivia Walch. Motion and heart rate from a wrist-worn wearable and labeled sleep from polysomnography. physionet.org, 2019. doi:10.13026/hmhs-py35.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko-tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. SUPERB: Speech processing Universal PERformance Benchmark. *arXiv preprint arXiv:2105.01051*, 2021. Interspeech 2021.

A THE ARGMAX AT EVERY CHECKPOINT

Table 2, the tabular form of Figure 2, lists the argmax at each of the 39 iteration-1 checkpoints for the probe and the nearest-neighbor readout. The two readouts last disagree at step 80,000, before the argmax settles at step 90,000.

Table 2: The argmax and its macro F1 at every iteration-1 checkpoint, for the ridge probe and for the nearest-neighbor (kNN) readout. Within the sweep, the ridge argmax stays at layer 10 from step 90,000 on, and the kNN argmax at layer 10 from step 70,000 on.

Step	Ridge argmax	Macro F1	kNN argmax	Macro F1
20,000	12	0.4430	12	0.4287
30,000	12	0.4582	12	0.4388
40,000	12	0.4670	12	0.4439
50,000	12	0.4782	12	0.4540
60,000	11	0.4866	12	0.4549
70,000	11	0.4952	10	0.4615
80,000	12	0.5026	10	0.4743
90,000	10	0.5017	10	0.4673
100,000	10	0.5128	10	0.4824
110,000	10	0.5185	10	0.4761
120,000	10	0.5281	10	0.4850
130,000	10	0.5333	10	0.4884
140,000	10	0.5346	10	0.4869
150,000	10	0.5436	10	0.4991
160,000	10	0.5456	10	0.4955
170,000	10	0.5437	10	0.4980
180,000	10	0.5495	10	0.5012
190,000	10	0.5511	10	0.5066
200,000	10	0.5528	10	0.5076
210,000	10	0.5596	10	0.5064
220,000	10	0.5633	10	0.5133
230,000	10	0.5570	10	0.5043
240,000	10	0.5672	10	0.5091
250,000	10	0.5669	10	0.5140
260,000	10	0.5651	10	0.5130
270,000	10	0.5694	10	0.5169
280,000	10	0.5696	10	0.5133
290,000	10	0.5691	10	0.5156
300,000	10	0.5633	10	0.5131
310,000	10	0.5626	10	0.5144
320,000	10	0.5651	10	0.5140
330,000	10	0.5673	10	0.5158
340,000	10	0.5675	10	0.5170
350,000	10	0.5642	10	0.5139
360,000	10	0.5678	10	0.5167
370,000	10	0.5688	10	0.5193
380,000	10	0.5666	10	0.5153
390,000	10	0.5683	10	0.5173
400,000	10	0.5683	10	0.5171

B EVERY CANDIDATE AT CONVERGENCE

Table 3 profiles all 13 candidates at the final iteration-1 checkpoint: the probe score and the fusion gain over the better sensor alone.

Table 3: Every probed representation at iteration-1 step 400,000. Fusion gain is macro F1 with both sensors minus the better of the two single-sensor ablations. ⋆ marks the selected layer.

Representation	Macro F1	Fusion gain
fused	0.3128	0.0036
1	0.3223	-0.0041
2	0.3453	0.0003
3	0.3650	0.0163
4	0.3891	0.0374
5	0.4321	0.0710
6	0.4659	0.0966
7	0.4980	0.1172
8	0.5286	0.1354
9	0.5611	0.1574
10 ⋆	0.5683	0.1602
11	0.5643	0.1556
12	0.5277	0.1187

Table 4: Which representation each label-free statistic would choose at the final iteration-1 checkpoint, and where the probe ranks that choice among the 13 candidates. The probe's own choice is layer 10. Of the 7 statistics, 2 select it and the other 5 select the candidate the probe ranks last. The gap is the probe's macro F1 on its own choice minus its macro F1 on the statistic's choice, at the same checkpoint.

Label-free statistic	Selects	Probe rank	Macro F1 gap
Covariance effective rank	10	1 of 13	0.0000
Participation ratio	10	1 of 13	0.0000
RankMe (uncentered, sqrt)	fused	13 of 13	0.2555
Spectral decay α	fused	13 of 13	0.2555
Minimum per-dimension std	fused	13 of 13	0.2555
Mean per-dimension std	fused	13 of 13	0.2555
Mean sq. off-diagonal corr.	fused	13 of 13	0.2555

C What each label-free statistic selects

The minimum per-dimension standard deviation is the cautionary case of Section 5.4. Over the encoder layers, it has the strongest correlation with the probe of any label-free statistic, yet it selects the pre-encoder space when that candidate is included (Table 4). Its high correlation holds only because the encoder-layer comparison excludes the one candidate it ranks wrongly. Table 7 restricts the comparison to the effective-rank conventions and adds the other sweeps and corpora.

D Label-free statistics against the probe, as correlations

Table 5 correlates each statistic against the probe in the two regimes of Section 5, and Figure 4 plots the per-candidate shapes the correlations summarize. The across-layer column has only 12 points, so it is reported with a permutation p-value.

E The linear probe

The probe regresses one-hot stage targets on standardized features by ridge regression, solved in closed form from sufficient statistics accumulated over the training split. Each feature dimension is centered on its training-split mean and divided by its training-split standard deviation, with 10^{-8} added to the variance. The validation features are standardized with the same training statistics. The 4 one-hot target columns are centered on the training-split class frequencies, which gives each class

Table 5: Spearman correlation of each label-free statistic against ridge-probe macro F1, in the two regimes. Across layers: the 12 encoder layers of the final iteration-1 checkpoint. Across checkpoints: 39 checkpoints of one pre-training run, at the selected layer. Spectral decay α and the mean squared off-diagonal correlation are negated, as in Appendix G, so a positive value means the statistic agrees with the probe. * marks a two-sided permutation $p < 0.05$ across layers. The across-checkpoint values are descriptive and carry no mark: successive checkpoints of one run are not independent samples.

Label-free statistic	Across layers	Across checkpoints
RankMe (sqrt, uncentered)	-0.406	+0.923
Cov. effective rank (no sqrt, centered)	+0.783*	+0.921
Participation ratio	+0.783*	+0.911
Spectral decay α	-0.014	+0.922
Minimum per-dimension std	+0.958*	+0.927
Mean per-dimension std	+0.559	+0.933
Mean squared off-diagonal corr.	-0.140	+0.914
Training step (baseline)	–	+0.923

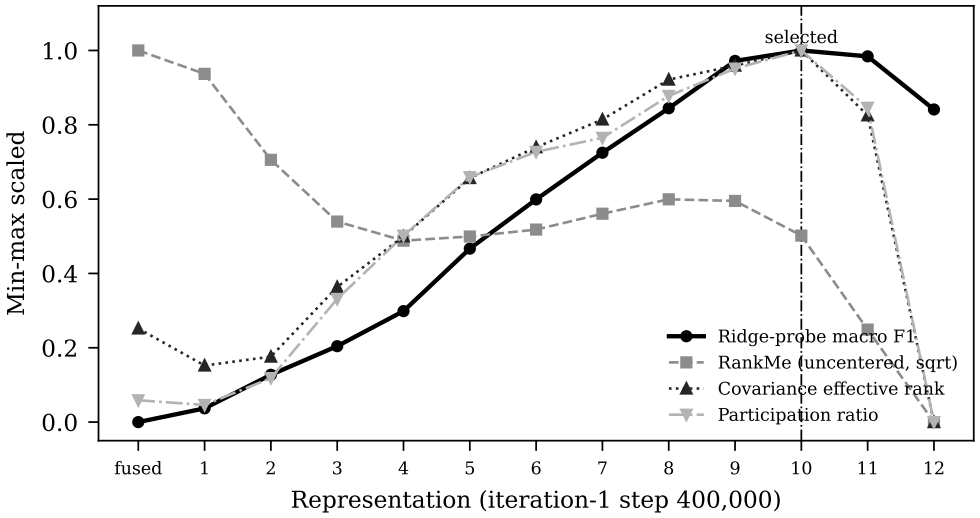

Figure 4: The probe and three label-free statistics across the thirteen representations of the final iteration-1 checkpoint, each scaled to its own range. Covariance effective rank and participation ratio peak where the probe peaks. RankMe is maximized on the pre-encoder space, which the probe ranks last.

an unpenalized intercept. A frame is assigned to the class with the largest regression score. Every frame has weight one: there is no class weighting and no resampling.

The regularization strength is added to the correlation matrix of the standardized features, so one grid serves every candidate whatever its activation scale. The grid is the 8 powers of ten from 10^{-6} to 10. Each candidate takes the strength with the highest validation macro F1. Macro F1 is the unweighted mean of the per-class F1 over all 4 classes, and a class with no predicted or no labeled frames scores zero. When two strengths tie, the smaller is taken, and when two candidates tie, the shallower is taken. Strengths tie in 134 of the 507 fits of the iteration-1 sweep, and a tie changes only the strength reported, never a score or an argmax. The smallest strength, at the edge of the grid, is selected in 322 of those fits, but in no fit do the two smallest strengths differ by more than 0.00004 in macro F1.

Every readout and statistic reads features that the encoder computes in evaluation mode with masking switched off. No frame is replaced by the mask embedding, dropout is inactive, and the batch normalization in the fusion layer of lineage A's iteration-1 encoder uses its running statistics. The pre-encoder candidate is the unmasked output of the fusion layer.

Recordings in a batch are padded to the longest one. The probe fit, the nearest-neighbor reservoir, the label-free statistics and the scoring use only the unpadded frames that carry a stage label. Padded frames are dropped only after the forward pass. Inside the encoder, self-attention ignores them, but the positional and depthwise convolutions read them, so the last frames of a padded recording can depend on the padding. The training split is batched in one fixed-seed order, so at a fixed batch size each training recording receives the same padding at every checkpoint. How much the padding changes a probe score was not measured.

The validation split chooses each candidate's ridge strength and the argmax, and it also gives the reported macro F1. Probe scores are therefore validation scores, not held-out test scores. The same holds on the public corpora (Appendix H).

F THE NEAREST-NEIGHBOR READOUT

If the selected layer were a property of ridge regression rather than of the encoder, a different readout would choose differently. The nearest-neighbor readout of Section 4 agrees with the ridge argmax at 36 of 39 iteration-1 checkpoints. On iteration 2 it settles one layer below, on 9, and stays there. On lineage B's iteration 1 it lands a layer below, and on its iteration 2 it takes the final layer, a layer above. On the size-matched BIDSleep probe it selects layer 10 where ridge selects the final layer. On the internal corpus the readout moves the argmax by about a layer, as much as the lineages disagree. The statements that the argmax is never the final layer refer to the linear probe, because the nearest-neighbor readout selects the final layer on lineage B's iteration 2.

For each candidate the readout keeps a reservoir of at most 100,000 training frames, a uniform sample of the training stream in the probe's fixed-seed order. The reservoir is drawn with seed 12345, is not stratified by participant or stage, and holds the same frames for every candidate. The reservoir is the whole training split where that split is smaller than the cap, as on sleep-accel and the size-matched BIDSleep cut. Each dimension is standardized with the reservoir's own mean and standard deviation, the latter floored at 10^{-8}, and neighbors are ranked by Euclidean distance in that space. A validation frame takes the majority stage among its k nearest reservoir frames. A tied vote goes to the lowest class index, in the order light, deep, REM, wake, as in scikit-learn's nearest-neighbor classifier. The neighbor count is chosen per candidate on validation macro F1 from $k \in \{1, 2, 3, 4, 5, 10, 20, 50, 100, 200\}$, the smaller count winning a tie, and no two counts tied in any probe.

The reservoir is drawn batch by batch, so at a fixed batch size it is the same at every checkpoint of a sweep, but a different batch size draws a different sample. Two separate probe runs over the same 33 iteration-2 checkpoints of lineage A agree on the readout's argmax at 32 of them. A candidate's readout macro F1 differs between the two runs by up to 0.011, against 0.0013 for the ridge probe. The readout does not separate layers whose scores differ by less than that.

Table 6 is the cross-lineage comparison of Section 5.3.

Table 7 reports, for each corner of the square described in Section 4, where the probe ranks the candidate that corner selects. Read down a column for one sweep or corpus and across a row for one convention. On the internal sweeps, neither square-root corner, RankMe's included, agrees with the probe. Without the square root, both corners rank better on every internal sweep, but only covariance effective rank agrees with the probe, and only on lineage A. Standardized RankMe's choice ranks last on every sweep and corpus.

G HOW THE LABEL-FREE STATISTICS ARE COMPUTED

All statistics are accumulated in double precision from a running mean, second moment and covariance. They use the same streamed frames the probe is fitted on, the labeled frames of the training split (Appendix E).

Let $\lambda_1 \geq \cdots \geq \lambda_D$ be the eigenvalues of the centered feature covariance and $\sigma_1 \geq \cdots \geq \sigma_D$ the singular values of the uncentered feature matrix. Write $H(\cdot)$ for the entropy of a spectrum normalized to sum to one. Tables 4 and 5 report 7 statistics: RankMe, covariance effective rank, participation ratio, spectral decay, the mean and minimum per-dimension standard deviation, and the mean squared

Table 6: Each column is one encoder at its last probed step, and all are probed with the same settings on the internal corpus. Column B, iter. 2 is trained on targets clustered from the encoder in column B, iter. 1. Column A, iter. 1 is a second, separate probe of the step-400,000 checkpoint in Tables 2 and 3, and its macro F1 at the ridge argmax is 0.5681 against their 0.5683. The ridge argmax is layer 10 or 11 and never the final layer; the argmax of the nearest-neighbor (kNN) readout ranges from layer 9 to the final layer. In every column, the final layer's covariance effective rank is below the argmax layer's.

	A, iter. 1	A, iter. 2	B, iter. 1	B, iter. 2
Last probed step	400,000	330,000	361,284	400,000
Ridge-probe argmax layer	10	10	11	11
kNN-readout argmax layer	10	9	10	12
Macro F1 at the ridge argmax	0.5681	0.6231	0.4045	0.4657
Margin over the final layer	0.0402	0.0105	0.0029	0.0009
Cov. effective rank, argmax	228.1	225.9	178.4	181.7
Cov. effective rank, final layer	123.1	180.5	154.2	146.1
Layer where removing a sensor costs most	10	9	11	10
Layer of largest fusion gain	10	9	11	9

off-diagonal correlation. The list below also defines four variants. Only Table 7 reports three of them, the two remaining corners and standardized RankMe, because the iteration-1 sweep behind the other two tables does not record them. The fourth, the fraction of dead dimensions, enters no table.

- **RankMe** (Garrido et al., 2023): $\exp H(\sigma)$, the exponentiated entropy of the *uncentered singular* values.
- **Covariance effective rank** (Roy & Vetterli, 2007): $\exp H(\lambda)$, over the *centered eigen* values.
- The two remaining corners of the effective-rank square, $\exp H(\sqrt{\lambda})$ and $\exp H(\sigma^2)$, so that a disagreement between the first two can be attributed to the square root or to the centering.
- **Standardized RankMe**: RankMe recomputed after each feature dimension is first rescaled to unit variance, so that a disagreement with RankMe can be attributed to scale rather than to the square root or centering choices the square above isolates.
- **Participation ratio**: $(\sum_i \lambda_i)^2 / \sum_i \lambda_i^2$, which is 1 at complete collapse.
- **Spectral decay** α: α-ReQ (Agrawal et al., 2022) models the covariance spectrum as a power law, $\lambda_k \propto k^{-\alpha}$, and estimates α by a linear fit to the whole spectrum on log-log axes. Here α is the least-squares slope of $\log \lambda_k$ against $-\log k$, fitted over the first half of the eigenvalues above 10^{-12} times λ_1, so that $k \leq 384$ for 768 dimensions. The tail of a finite-sample covariance spectrum is noise crushed toward zero and has large leverage on a log scale, so a full-spectrum fit biases α upward unless the frame count far exceeds the dimension.
- **Dimension statistics**: the mean and minimum per-dimension standard deviation, and the fraction of dimensions whose standard deviation falls below 10^{-4} times the mean. These dimensions are dead, and the rest are live.
- **Mean squared off-diagonal correlation**: the squared correlation summed over pairs of live dimensions and divided by $D(D-1)$. Dead dimensions are left out of the sum, since their correlation is cancellation noise divided by a vanishing standard deviation.

Eigenvalues are computed in double precision, and negative ones, which arise only from round-off, are clipped to zero. Before the entropy, each spectrum is divided by its sum, with 10^{-12} added to the sum and to every entry, so a zero eigenvalue contributes a finite term. No probed space has a dead dimension. In every candidate, internal and public, the smallest per-dimension standard deviation is at least 0.17 of the mean, so the off-diagonal average runs over all pairs.

Each statistic is oriented before use, in the selections of Table 4 and the correlations of Table 5. The off-diagonal correlation is negated, because a smaller value is the better-conditioned one. The decay

exponent is negated as well, so a slower decay ranks higher. A positive correlation therefore always means the statistic agrees with the probe. The monotone ordering of α is a convention of this paper, not a rule of α-ReQ. In α-ReQ, a good representation has an α close to one, and an α either too high or too low marks poor generalization (Agrawal et al., 2022).

H PUBLIC-CORPUS TRANSFER

Table 8 gives the probe results on the public corpora. All of them probe lineage A's iteration-2 encoder at step 180,000. Scores are on each corpus's validation split, which also chooses the ridge strength and the argmax, so they are validation scores, not held-out test scores.

Each public corpus is split once, by participant, into a training and a validation split, with no test split. The adapter assigns every fifth participant in identifier order to validation, so no participant contributes nights to both splits. BIDSleep's training split holds 38 participants over 206 nights and 167,024 frames, and its validation split 9 participants over 47 nights and 37,130 frames. The cut keeps the first night of each of the first 31 participants in identifier order, and each night stays in the split it already had. It trains on 25 participants with 21,538 frames and validates on 6 with 5,139. On sleep-accel each participant contributes one night, 25 participants with 21,542 frames for training and 6 with 5,231 for validation. Frames count labeled epochs only.

On BIDSleep and sleep-accel the PPG waveform is written as zeros when each corpus is converted, before the STFT. A zero waveform has zero power, which the log transform floors at -100 dB in every frequency bin, and the checkpoint's PPG scaler then standardizes that floor. On these corpora the PPG input is therefore the standardized log floor, constant over time, rather than the scalers' mean, and its green wavelength channel is kept. The zero-PPG ablation of Section 4 instead sets the standardized PPG spectrogram and its wavelength channels to zero.

Stage coding. Both corpora publish their stage coding with the pinned version. BIDSleep 1.0.0 codes each expert-scored 30-second epoch as 0 Wake, 1 N1, 2 N2, 3 N3, 4 REM or 5 Unknown (Song, 2026). The adapter maps N1 and N2 to light sleep and N3 to deep sleep, and writes the 2,363 Unknown epochs as unscored. The sleep-accel 1.0.0 record codes 0 Wake, 1 N1, 2 N2, 3 N3 and 5 REM (Walch, 2019). Its label files also carry -1 on 438 epochs and 4 on 356, which the reference code linked from the dataset's page names unscored and N4. The adapter writes -1 as unscored and merges N4 into deep sleep with N3. Unscored epochs enter neither fitting nor scoring. The BIDSleep adapter also writes as unscored every epoch less than half covered by motion samples, 11,760 further epochs, so 14,123 of its 218,277 epochs are unscored. The sleep-accel adapter has no such rule and fills missing motion samples from the recorded ones by interpolation. On sleep-accel, 1,218 scored epochs have no motion sample, 839 of them in the validation split, 16% of its frames. Of those, 826 come from one participant whose motion record ends early in the labeled night.

Resampling. Both adapters resample ACC in two steps. Each axis is first interpolated linearly from its recorded timestamps onto a uniform grid that starts at the first labeled epoch. The sleep-accel grid runs at 50 Hz. Each BIDSleep night's grid runs at the rate of that night: the inverse of the median interval between its motion samples in the labeled span, rounded to a whole hertz. Of the 253 BIDSleep nights, 166 run at 50 Hz and the rest between 33 and 66 Hz. The grid is then resampled to 25 Hz by polyphase filtering. The anti-aliasing filter is a Kaiser-windowed low-pass filter, with $\beta = 5$ and its cutoff at 12.5 Hz, the Nyquist frequency of the output. Every resampled epoch holds 30 s of samples.

BIDSleep's labels start at the recording's start time, a local clock time. The adapter aligns that time to the motion clock by a whole-hour offset, rounded from the gap to the first motion sample. It drops motion samples with a non-finite value and sorts the rest by time.

The sleep-accel adapter neither sorts nor filters the motion samples, and it reads their timestamps in file order. In 11 of the 31 motion files, some timestamps are earlier than the one before them, although linear interpolation requires increasing times. Within the labeled nights, 81 such backward steps occur, in the files of 7 participants, and the largest goes back 160 s. Sorting the samples by time would change the converted ACC in 308 scored epochs, 20 of them in the validation split. Which order is correct is not known.

I FRONT END

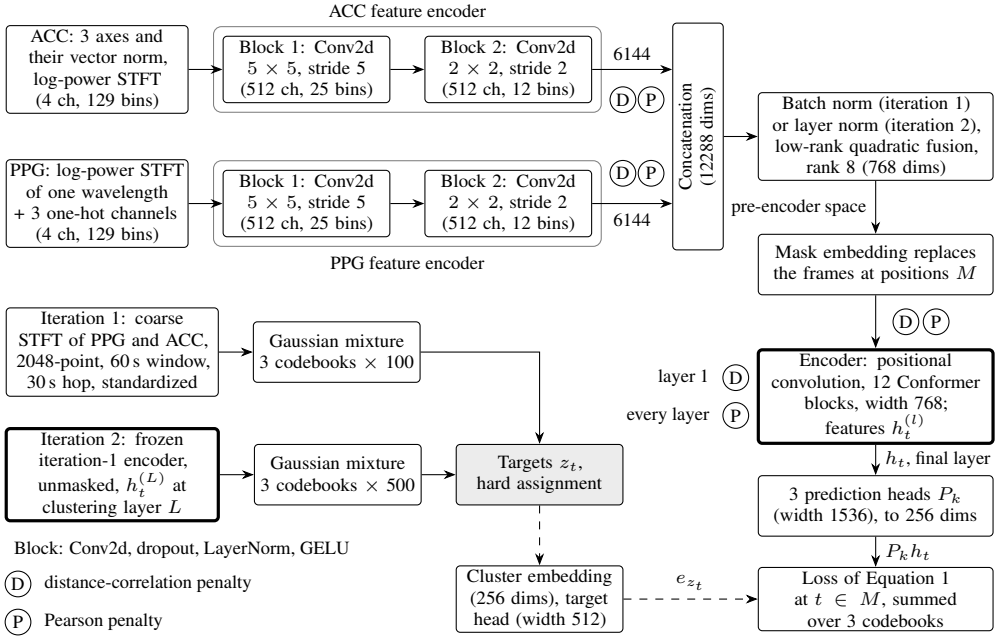

Figure 5: The wHuBERT model as trained in lineage A. The four input channels of each sensor pass through its own feature encoder. A low-rank quadratic fusion maps the concatenation to the pre-encoder space. After masking, a Conformer encoder computes the features $h_t^{(l)}$. Three prediction heads project its final layer, and Equation 1 scores them against the target embeddings. Dashed arrows carry the targets to the loss. The iterations differ in the targets (bottom left) and in the fusion's normalization. D and P mark where each penalty acts. On the pre-encoder space, both penalties see a masked frame as the mask embedding.

The STFT is a 256-point transform over a 6 second window at a 3 second hop. It gives 129 frequency bins covering 0 to 12.5 Hz, at 50% overlap. The window is a periodic Hann window, zero-padded to 256 points. STFT frames are centered on multiples of the hop, and the signal is reflected at each end of the recording. ACC carries its three axes plus their vector norm, computed from the 25 Hz samples before the transform. PPG carries the log-power STFT of one wavelength and three constant channels that one-hot encode which wavelength it is: green, red or infrared. A recording with several wavelengths yields one training sequence per wavelength. Of the 277,891 pre-training files, 939 carry all three wavelengths and the rest carry green only. ACC's four channels and PPG's signal channel are standardized, and the wavelength channels are not.

Each sensor has its own feature encoder of two blocks. Each block is a Conv2d, dropout of 0.1, a LayerNorm over channels and frequency at each frame, and a Gaussian error linear unit (GELU). The first Conv2d maps 4 channels to 512 with a 5×5 kernel and stride 5 on both the frequency and the time axis. The second maps 512 channels to 512 with a 2×2 kernel and stride 2 on both axes. Neither pads. The frequency axis shrinks from 129 to 25 to 12 bins, which drops the bins from 11.7 Hz up. Every ten STFT frames become one encoder frame, one per 30 seconds. The windows of those ten STFT frames span 33 s: the encoder frame's own 30 seconds and the last 3 s of the preceding 30. Neighboring encoder frames therefore overlap by 3 s, and no encoder frame reads past the end of its own 30 seconds. The 512 channels and 12 bins are flattened into 6144 dimensions per sensor, and the two sensors are concatenated into one 12288-dimensional vector. Each feature encoder has 1,138,688 parameters.

The fusion layer maps the concatenation to the encoder width of 768. We use a quadratic projection instead of a simple linear one to account for the non-linear nature of movement noise in PPG signals (Kim et al., 2017). Since a full quadratic combination of 12288 channels is impractical and computationally wasteful, we employ a low-rank implementation of the quadratic-form matrices

(Panicker et al., 1998), of rank 8. Such low-rank implementations reduce the number of parameters and the computational complexity (Pinheiro & Lopes, 2019). With x the normalized concatenation, output o of the layer is

$$y_o = w_o^\top x + \sum_{r=1}^{8} (u_{ro}^\top x)(v_{ro}^\top x) + b_o. \tag{2}$$

The factors u_{ro} and v_{ro} belong to one output and are not shared across outputs. One bias-free linear map from 12288 to $768 \cdot (1 + 2 \cdot 8)$ dimensions computes the linear terms and both banks of factors, and a single bias per output takes the place of factor biases. In lineage A's encoders the layer has 160,457,472 parameters, 48.5% of the 330.8M parameters of the whole encoder. The normalization of x is batch normalization in lineage A's iteration-1 encoder and layer normalization in its iteration-2 encoder. Both normalize over valid frames only, and padded frames leave the layer as zeros. Rank 8 also restricts which quadratic interactions the projection can represent. This paper reports no ablation of the rank. Figure 6 illustrates this fusion architecture.

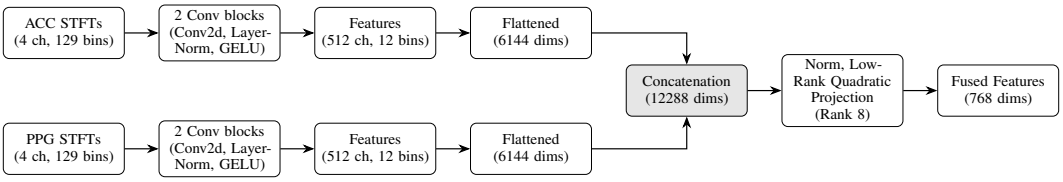

Figure 6: Architecture of the front-end fusion layer combining ACC and PPG STFTs into a joint feature space.

J THE MASKED-PREDICTION OBJECTIVE

The details below are those of lineage A's runs. Each of the three codebooks has its own prediction head, cluster embeddings and target head, and Equation 1 is evaluated once per codebook. For codebook k, P_k maps the final encoder layer's 768-dimensional output to 256 dimensions through four blocks of a linear layer, batch normalization and GELU, of width 1536, and a final linear layer. The target embedding e_z is a learned 256-dimensional embedding of cluster z, passed through a second head of the same form, of width 512. The loss is the cross-entropy of Equation 1, averaged over the masked frames of the batch on each accelerator and summed over the three codebooks.

The three mixtures of an iteration are fitted on the same features, with the same number of components, from the same batches. They differ only in their k-means++ initialization, drawn in turn from one seeded random stream. They are therefore three restarts of one fit.

Each mixture has diagonal covariances and is fitted by gradient descent on the mean negative log-likelihood, not by expectation maximization. Its means start from k-means++ seeding on the first batch of the fit, with no k-means iterations after it. Its variances start at the per-dimension variance of that batch, and its weights start uniform. The fit makes one pass over its batches with no convergence test, one Adam step per batch: 4,371 steps in iteration 1 and 8,743 in iteration 2. The learning rate follows a cosine from 10^{-2} to 10^{-4}, and the gradient norm is clipped at 1. Every variance is floored at 0.1, in the fit and in the assignment, and a floored variance receives no gradient. Each frame's target is the component with the highest posterior, a hard assignment. Lineage B's mixtures follow the same recipe.

Iteration 1's mixtures cluster standardized features, so the floor is a tenth of their unit variance. Between 81% and 84% of the variances of each fitted iteration-1 codebook sit at the floor, and none of iteration 2's do. Iteration 2's mixtures read layer 10 of the iteration-1 encoder in evaluation mode, without masking. The perplexity of the fitted mixture weights is 56 to 61 in iteration 1's codebooks and 54 to 62 in iteration 2's. How often frames take each component was not counted.

The negatives N_t are 100 draws, with replacement, from the other masked frames of the same sequence. A negative in the positive's cluster has the positive's embedding and is removed from the denominator. Padded frames are never masked, so they are never positives or negatives, and they are left out of the mixture fit. They do pass through P_k, and its batch normalization therefore includes them.

Masking uses the span sampler of the training code with probability 0.8, span length 10 and at least two spans. For a sequence of n valid frames it draws $\lfloor 0.8n/10 + u \rfloor$ span starts, with u uniform on $[0, 1)$, and masks 10 frames from each. It then equalizes the batch: each sequence keeps a random subset of its masked frames, as many as the least-masked sequence in the batch has. Training sequences are whole files, padded rather than cropped, and their lengths are bimodal: 50% of the 279,769 training sequences have 400 encoder frames and 42% have 80. A batch of 32 on one accelerator therefore almost always contains a short sequence. On average a sequence keeps 27 masked frames, 7% of a 400-frame sequence, and its negatives are drawn from that small set.

We measured the masked share by running the sampler of the training code, with lineage A's settings and padding mask, on 1,000 batches of 32 sequences drawn from the training-sequence lengths. On average 11% of valid frames are masked (20% averaged per sequence), against 57% without the equalization. The subset is drawn frame by frame, so masked runs average 1.5 frames, and only 8% of masked frames lie in runs of 10 or more.

K Modality ablation on every candidate

A sensor is zeroed where it enters the encoder, after its log-power STFT has been standardized by the checkpoint's input scalers. The scalers subtract a per-channel, per-frequency mean and divide by a standard deviation, both stored in the checkpoint. The zeroed sensor therefore carries the scalers' mean log-power spectrum, constant over time, in each of its channels: ACC's three axes and vector norm, and PPG's spectrogram. PPG's three wavelength channels, which encode green in every probed recording, are zeroed with it. The public corpora of Section 6 are ablated the same way. Only the ridge probe is refitted on ablated inputs. The nearest-neighbor readout and the label-free statistics are computed on unablated inputs only.

Figure 7 shows the modality ablation on each candidate at the final iteration-1 checkpoint, where the ACC-only probe selects layer 10. On lineage A's iteration-2 encoder at step 180,000, the encoder probed in Section 6, the ACC-only probe instead selects the final layer, at 0.4128 against 0.4038 at layer 10.

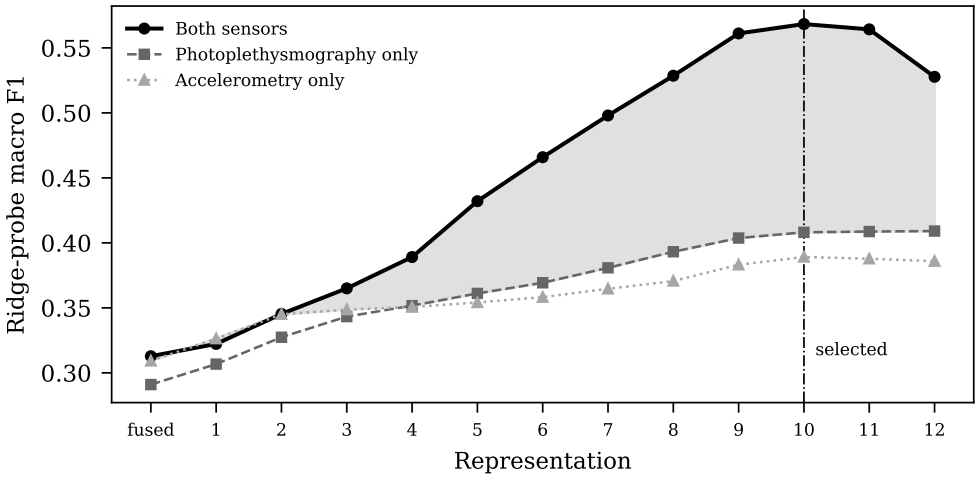

Figure 7: Ridge-probe macro F1 of each candidate at the final iteration-1 checkpoint, with both sensors and with each sensor alone, the probe refitted in each case. The shaded band is the fusion gain, the margin of both sensors over the better one alone.

L Pre-training data

Every pre-training run of Sections 5 and 6, both iterations of lineages A and B, uses one pool and one participant filter. The pool holds 40 collections of internal recordings: every study-device collection of the internal corpus (Section 7), further internal sleep studies, and three internal collections that the

probe and the downstream evaluation do not use. The last three supply 92.9% of the pool's 277,891 files. No labels enter pre-training.

The internal corpus is split by participant: 2,098 participants for training, 246 for validation and 399 for test, 2,743 in all. A participant is counted here as a study and a participant number in the recordings' file names, and no such pair appears in two splits. Whether one person appears under numbers in two studies cannot be read from the files. Of the training participants, 172 have recordings from both a ring and a watch.

Before pre-training, every file of a test-split participant is removed from each collection whose split has a test part. Validation participants are not removed. Pre-training, the input scalers, fitted on the whole pool, and iteration 2's mixture therefore saw the validation participants' signals without labels, and the probe is scored on those participants' labeled recordings. Iteration 1's mixture is fitted on the first half of the pool in its stored order, 139,872 of 279,769 sequences. All of them come from the three collections outside the sleep studies, so iteration 1's mixture saw no recording of the internal corpus.

Two collections of ring recordings have no filter entry and enter the pool in full. Of the 575 test recordings, 60 share a study and participant number with ring recordings in the pool. For 45 of them, a ring file starts at the same millisecond as a watch file with that number. No test-split recording itself enters the pool.

Lineages A and B share the pool, the seed, the schedule, the per-accelerator batch, the masking, the codebook sizes and the encoder's depth and width. Lineage B has the Pearson penalties but no distance-correlation penalty (Appendix M). Its fusion layer uses layer normalization, as lineage A's iteration 2 does, where lineage A's iteration 1 uses batch normalization (Appendix I). Lineage A's iteration 1 also standardizes the signal features before clustering them, a setting absent from lineage B's configuration. In lineage B's iteration 2, the Pearson penalty on the encoder layers is weighted 0.01 where the other three runs use 0.1. Lineage B ran on a different revision of the training code, and its iteration 1 stopped at step 361,284 of a 400,000-step schedule. Its iteration 2 clusters layer 10 of that final checkpoint.

The encoder fine-tuned in Section 7 comes from a separate lineage, not probed in Sections 5 and 6. Its iteration 2 clusters layer 10 of its iteration-1 encoder, as lineage A does, into codebooks of the same sizes. Both of its iterations fit their codebooks by k-means instead of a Gaussian mixture. Their configurations set neither the Pearson nor the distance-correlation penalties of lineages A and B. Instead, each adds one other penalty at weight 0.1. In iteration 2's code, that penalty is the sum of the squared correlations between every dimension of the ACC feature-encoder output and every dimension of the PPG one. Each dimension is scaled to unit norm over the frames of the batch, padded frames included, and is not centered. Whether iteration 1's code computes the same penalty is not established. The lineage trains on one accelerator, with 32 sequences per step for 200,000 steps in iteration 1 and 48 per step for 400,000 steps in iteration 2. Its pool omits some of lineage A's collections and uses the same test lists, so no test-split recording enters it, while the validation participants and the ring recordings above do. Whether its iteration-1 codebooks saw recordings of the internal corpus is not known.

M Optimization

Both iterations of lineages A and B use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ at a peak learning rate of 10^{-4}. Weight decay is 10^{-2} on the encoder's weight matrices, 0.1 on those of the prediction heads, target heads and cluster embeddings (Appendix J), and zero on biases and normalization parameters. The learning rate rises linearly from zero over the first 32,000 steps, then decays exponentially to 5% of its peak, 5×10^{-6}, at step 400,000. Training is data-parallel on 4 accelerators, each drawing 32 sequences per step, so an optimizer step sees 128 sequences. There is no gradient accumulation and no gradient clipping, and training runs in 16-bit mixed precision.

Pearson penalties. Lineages A and B add a Pearson penalty in three spaces, the ACC and PPG feature-encoder outputs and the pre-encoder space, weighted 10, 10 and 20. On the unpadded frames of the batch, the penalty is the mean squared correlation over pairs of distinct dimensions, plus a hinge $\frac{1}{D}\sum_j \max(0, 1 - s_j)$ on the standard deviation s_j of each dimension. The same penalty is

applied to the output of every encoder layer. The per-layer penalties are averaged with weight 1.2^{l-1} for layer l, so the final layer carries 19% of the average and the first layer 2.5%. The average is weighted 0.1 in every run except lineage B's iteration 2 (Appendix L). Lineage A also carries the distance-correlation penalty of Section 3. That penalty has no hinge, and each step estimates it on 128 frames and 64 dimensions drawn at random. In training, both penalties on the pre-encoder space see each masked frame as the mask embedding that replaces it.

N Downstream results

Table 1 in Section 7 gives the sleep-staging metrics aggregated over all classes. Across recordings, the standard deviation of macro F1 lies between 11.9 and 12.7 points for all three models. The differences in mean are 2.07 points between the student and the deployed baseline and 2.95 points between the fine-tuned encoder and the baseline. The baseline's scores are transcribed from documentation, without per-recording values, so no paired interval can be computed for either difference.

Fine-tuning. The fine-tuned encoder starts from the final iteration-2 checkpoint of its lineage (Appendix L). It trains on the 2,764 training recordings of the internal corpus, 12 whole recordings per step, on one accelerator in 16-bit mixed precision, without gradient accumulation or clipping. At each frame, the output of the final encoder layer passes through a perceptron with three hidden layers of width 1,536. Each hidden layer is a linear map, layer normalization, GELU and dropout 0.3. The perceptron's input has dropout 0.01, and its output layer maps to 256 dimensions. A last linear layer maps those to the 4 stages. The loss is the cross-entropy summed over the labeled frames of the batch, without class weights.

AdamW updates the encoder at a peak learning rate of 5×10^{-5} and the last linear layer at 2×10^{-4}, both with weight decay 10^{-3}. The perceptron is not trained: it keeps its random initialization and acts as a fixed nonlinear map between the two. The encoder is frozen for the first 5,000 steps, while the learning rate rises linearly from zero, and the rate is then held at its peak. The exponential decay of the schedule starts at step 20,000, after training stopped. During training, frames are masked with lineage A's pre-training span settings (Appendix J). A channel mask, with probability 0.5 and span length 64, also zeroes spans of the 768 pre-encoder channels, the same channels at every frame of a recording.

After each epoch of 231 steps, macro F1 is computed on each validation recording and averaged. Training stops once that mean has not improved for 50 epochs, at step 19,635. The reported encoder is the checkpoint with the best validation macro F1, 0.7582 at step 8,085. No random seed is set.

Parameter counts. Lineage A's encoder has 330.8M parameters, front end included (Section 3), and 160,457,472 of them are in its low-rank quadratic fusion (Appendix I). The fine-tuned encoder of Table 1 comes from the separate lineage of Section 7. Its checkpoint holds 179.8M parameters in the encoder and 186.1M with the classification head, 6,305,536 of them in the untrained perceptron. The Conformer stack, positional convolution and mask embedding hold 168,073,856 parameters in both encoders, so the gap between them lies in the front end. The fusion layer of the fine-tuned encoder is a layer normalization that both sensors share, followed by one linear map to the encoder width: 9,450,240 parameters, against lineage A's 160,457,472. Its PPG feature encoder reads one input channel where lineage A's reads four, and so has 38,400 fewer parameters. These two differences account for the whole gap.

Distillation. The student is trained on the 2,764 training recordings of the internal corpus, with the split configuration the encoder was fine-tuned with. The 339 validation recordings select the checkpoint and stop training early. Each training recording is read twice: by the teacher from its PPG and ACC signals, and by the student as inter-beat intervals, 8 ACC summaries and 4 timestamp features. The loss is $0.75\,\mathcal{L}_{\text{focal}} + 0.25\,\mathrm{KL}(p_{\text{teacher}} \,\|\, p_{\text{student}})$. The focal loss, with $\gamma = 2$, compares the student's logits to the expert labels, and both posteriors are plain softmaxes, at temperature 1. The teacher is frozen for the first 20,000 steps and trainable afterward, through the KL term only, at a peak learning rate of 5×10^{-5}. Training is capped at 50,000 steps and stopped early at step 28,717. The reported student is the checkpoint with the best validation F1 score, at step 20,067, so its teacher had been trainable for 67 steps.

O PER-STAGE DOWNSTREAM RESULTS

Table 9 gives per-stage F1 for the fine-tuned encoder and the distilled student. For both, deep sleep has the lowest mean F1, roughly three quarters of the light-sleep figure. Per-stage scores for the deployed baseline are not reported, so the table cannot show whether either model improves deep-sleep classification over it.

Tables 1 and 9 are computed per recording by the same evaluator, and they differ in which recordings enter each mean. For one test recording, macro F1 is the unweighted mean of the per-stage F1 over the stages that occur in that recording's expert labels or in the model's predictions. A stage absent from both is left out, and a stage the model predicts but the labels do not contain scores zero. Table 1 averages this value over all test recordings. Each row of Table 9 averages instead over the recordings whose expert labels contain that stage. The two therefore weight the stages differently: the per-stage means average to 72.39 for the fine-tuned encoder and 71.52 for the student, not to the macro F1 of Table 1. The standard deviations of both models are population standard deviations across recordings.

The evaluator pairs a recording's per-stage scores with the stages in its expert labels by position. When the model predicts a stage that the labels lack, the stages after it, in the order light, deep, REM, wake, receive the wrong scores. The light-sleep entries are therefore never affected, and the deep-sleep entries only in a recording whose labels lack light sleep. Macro F1 is unaffected. The number of affected recordings is not known.

Table 7: Where the probe ranks the representation each corner of the effective-rank square selects. Lower is better; 1 is agreement. Every corner taking the square root, which is RankMe's convention, ranks tenth or worse, except RankMe's own corner on BIDSleep and BIDSleep, cut.

Effective rank	A, iter. 1	A, iter. 2	B, iter. 1	B, iter. 2	BIDSleep	BIDSleep, cut	sleep-accel
$\exp H(\sigma)$, uncentered (RankMe)	13 of 13	13 of 13	12 of 13	13 of 13	1 of 13	2 of 13	10 of 13
$\exp H(\sqrt{\lambda})$, centered	13 of 13	13 of 13	12 of 13	13 of 13	13 of 13	13 of 13	13 of 13
$\exp H(\sigma^2)$, uncentered	3 of 13	6 of 13	5 of 13	5 of 13	2 of 13	1 of 13	11 of 13
$\exp H(\lambda)$, centered (cov. effective rank)	1 of 13	1 of 13	4 of 13	3 of 13	13 of 13	13 of 13	13 of 13
RankMe on standardized features	13 of 13	13 of 13	13 of 13	13 of 13	13 of 13	13 of 13	13 of 13

Table 8: Frozen-encoder transfer at 25 Hz. Every corpus probes lineage A's iteration-2 encoder at step 180,000, with the scalers of that checkpoint; only the readouts are fitted, on each corpus's own participant-disjoint split. Frames are the probe's training split, and scores are on the validation split, which also chooses the ridge strength. "Selects" is the argmax representation on that corpus. † marks a corpus where it differs from layer 10, this encoder's internal argmax with both sensors; with PPG zeroed, the internal probe of the same encoder selects the final layer. Absolute scores are not comparable across corpora, which differ in cohort and in which sensors they carry.

Corpus	Sensors	Subj.	Frames	Macro F1	Selects
BIDSleep	ACC only	47	167,024	0.5474	layer 11 †
BIDSleep, cut	ACC only	31	21,538	0.5392	layer 12 †
sleep-accel	ACC only	31	21,542	0.4646	layer 6 †

Table 9: Per-stage F1 (%) on the test split, per recording (mean ± standard deviation across recordings). The Deep row has the lowest mean F1 for both models.

Stage	Fine-tuned encoder	Distilled student
Wake	78.06 ± 16.86	75.38 ± 18.04
Light	80.85 ± 8.26	81.16 ± 9.83
Deep	59.14 ± 25.13	58.16 ± 26.49
REM	71.51 ± 31.31	71.39 ± 29.87