BrainDistill: Implantable ECoG-Based Motor Decoding with Task-Specific Knowledge Distillation

Yuhan Xie [1] **Jinhan Liu** [1] **Xiaoyong Ni** [1] **Fei Tan** [1] **Icare Sakr** [1 2] **Thibault Collin** [1 2] **Shiqi Sun** [2] **Alejandro Rodriguez Guajardo** [1 2] **Demon Fanny** [3] **Charles-francois Vincent Latchoumane** [2] **Henri Lorach** [2 4] **Gregoire Courtine** [1 2 4 5] **Mahsa Shoaran** [1]

Abstract

Transformer-based neural decoders with high parameter counts, pre-trained on large-scale datasets, have recently achieved high accuracy on complex brain-computer interface (BCI) tasks, such as motion-intention decoding. However, their large model size and high computational demands hinder their deployment in power-constrained implantable systems. To address this challenge, we introduce **BrainDistill**, a novel implantable motor decoding pipeline that integrates an implantable neural decoder (**IND**) with a task-specific knowledge distillation (**TSKD**) framework. Unlike standard feature distillation methods that attempt to preserve teacher representations in full, TSKD explicitly prioritizes features critical for decoding through supervised projection. Across multiple ECoG datasets, IND consistently outperforms prior neural decoders on motor decoding tasks, while its TSKD-distilled variant further outperforms alternative distillation methods. Finally, we present a quantization-aware training scheme that enables integer-only inference with activation clipping ranges learned during training. The quantized IND enables deployment under the strict power constraints of implantable BCIs with minimal performance loss.

[1]Neuro-X Institute, EPFL, Geneva, Switzerland [2]NeuroRestore, Defitech Center for Interventional Neurotherapies, EPFL/CHUV/UNIL, Lausanne, Switzerland [3]Department of Medicine, University of Fribourg, Fribourg, Switzerland [4]Department of Clinical Neuroscience, Lausanne University Hospital (CHUV) and University of Lausanne (UNIL), Lausanne, Switzerland [5]Department of Neurosurgery, CHUV, Lausanne, Switzerland. Correspondence to: Yuhan Xie <yuhan.xie@epfl.ch>.

Proceedings of the Workshop on Structured Data for Health at the 43rd International Conference on Machine Learning, Seoul, South Korea. Copyright 2026 by the author(s).

1. Introduction

Decoding movement intentions from neural recordings is a fundamental application of Brain-Computer Interfaces (BCIs), enabling the control of various devices such as drones (Duan et al., 2019), cursors (Bundy et al., 2016), and, most importantly, neural prostheses for patients with disabilities (Lorach et al., 2023). Multiple recording paradigms have been developed to capture neural activity from the motor cortex, including EEG (Cruz-Garza et al., 2014), intracortical spikes (Ma et al., 2017), and Electrocorticography (ECoG) (Volkova et al., 2019). Among these, ECoG offers a favorable balance between spatiotemporal resolution and practicality: it provides higher resolution than EEG while remaining more feasible for long-term use compared with invasive intracortical recordings.

Recently, large neural foundation models pre-trained on diverse neural datasets have been introduced to boost decoding performance and improve long-term stability (Zhang et al., 2023; Wang et al., 2023; 2024; Yang et al., 2023). These models yield generalizable feature representations that can be efficiently adapted to new datasets with minimal fine-tuning. However, their growing sizes pose significant challenges for the practical deployment of BCI systems. In fully implantable and long-term medical devices, Systems-on-Chips (SoCs) have been widely adopted (Shao et al., 2025). These SoCs integrate recording, decoding, and potential stimulation modules into a single cortical implant, making compact and energy-efficient decoders essential.

Knowledge distillation (KD) is a popular approach to bridge this gap by transferring knowledge from foundation models to compact student models. However, existing KD methods primarily aim to preserve teacher embeddings as completely as possible (Miles et al., 2024; Zhou et al., 2025; Guo et al., 2023), which becomes problematic when the student model lacks the capacity to mimic complex teacher features, resulting in limited performance gains. To address this challenge, we introduce **BrainDistill**, a novel implantable motor decoding pipeline that combines a lightweight neural decoder with a task-specific knowledge distillation framework. Figure 1 illustrates the BrainDistill pipeline and distillation process.

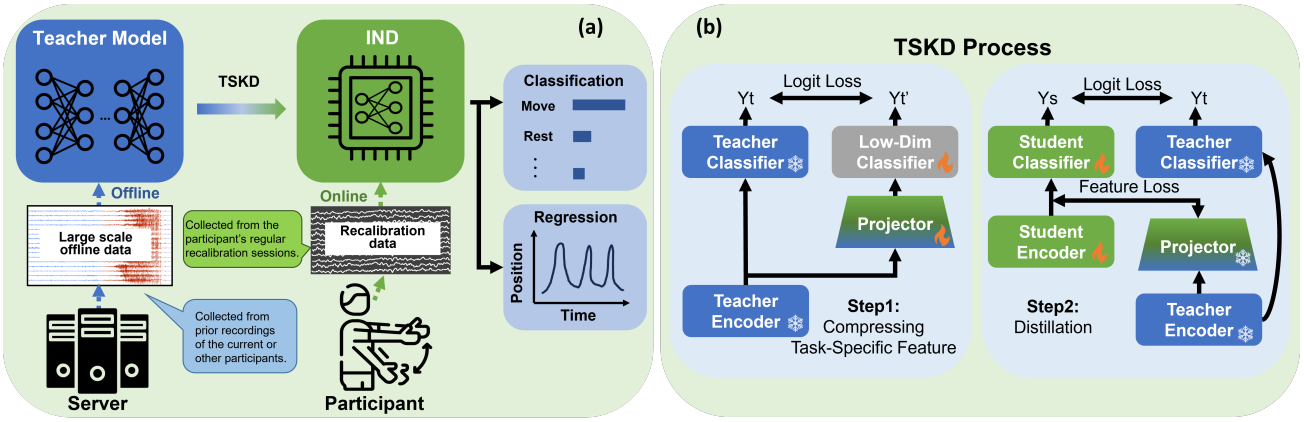

Figure 1. **BrainDistill pipeline:** (a) Training and testing paradigms of BrainDistill. IND distills from a pre-trained large neural model using small-volume recalibration data via TSKD, and then performs online motor decoding. (b) The two-step TSKD process. First, teacher embeddings are compressed in a supervised manner by a projector; second, the fixed projector is used to align student embeddings with the task space.

Our main technical contributions are:

- We propose **TSKD**, a novel distillation method that learns a teacher-to-student projection to compress task-specific information and bridge the capacity gap between teacher and student models. We further demonstrate the advantages of TSKD over prior approaches that rely on inverse student-to-teacher projections.
- In addition, we present **IND**, a neural decoder that tokenizes via the Continuous Wavelet Transform (CWT) and adopts a lightweight linear transformer architecture. To enable implantable decoding, we enhance integer-only quantization performance by a quantization-aware training (QAT) scheme using learnable clipping ranges.

We demonstrate the advantage of BrainDistill through comparisons with existing decoders and distillation methods, as well as an analysis of chip-level power consumption. Appendix A.1 reviews prior studies on motor decoding, knowledge distillation, and integer-only quantization, and highlights the novelty of our approach.

2. Method

2.1. Problem Definition

We consider a motor decoding task with neural data $\mathcal{X} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, where $x_t \in \mathbb{R}^{T \times C}$ represents a segment of brain signals with look-back length T and channel number C. At time t, y_t denotes either the category of movements or the joint position. BrainDistill is designed to support two paradigms: when a relatively large recalibration dataset $\mathcal{X}_{\text{recalib}}$ is available, IND can be trained on it directly; when it is small or the task is particularly challenging, IND can instead distill from neural foundation models pre-trained on large offline data $\mathcal{X}_{\text{offline}}$ using TSKD.

2.2. Task-Specific Knowledge Distillation

For the teacher model $F_{\mathcal{T}}$, we denote its embedding generated from $\mathcal{X}_{\text{recalib}}$ as $z_{\mathcal{T}}$, its classifier as $f_\phi^{\mathcal{T}}$, and its final prediction as $\hat{y}_{\mathcal{T}}$. We use similar notation for the student model $F_{\mathcal{S}}$. The objective of distillation is to minimize $\mathcal{L}_{\text{Distill}} = \|f_\phi^{\mathcal{T}}(z_{\mathcal{T}}) - f_\phi^{\mathcal{S}}(z_{\mathcal{S}})\|^2$ on recalibration data, where $z_{\mathcal{T}} \in \mathbb{R}^{d_t}$, $z_{\mathcal{S}} \in \mathbb{R}^{d_s}$, $d_s \ll d_t$. We can simplify $\mathcal{L}_{\text{Distill}}$ by focusing only on the weight matrix of f_ϕ, which becomes $\mathcal{L}_{\text{Distill}} = \|W_{\mathcal{T}}^\top z_{\mathcal{T}} - W_{\mathcal{S}}^\top z_{\mathcal{S}}\|^2$, $W_{\mathcal{T}} \in \mathbb{R}^{d_t \times K}$, $W_{\mathcal{S}} \in \mathbb{R}^{d_s \times K}$, and K denotes the output dimension of the decoder. For logit-based distillation, both the student encoder and classifier are randomly initialized and jointly optimized, which often leads to suboptimal convergence. Following prior work, we consider matching not only the logits but also the feature representations. We introduce a projection matrix $P \in \mathbb{R}^{d_t \times d_s}$. The objective is to train the student model and P such that $z_{\mathcal{S}} \approx P^\top z_{\mathcal{T}}$, thereby transferring task knowledge from teacher embeddings to student embeddings. We rewrite $z_{\mathcal{S}} = P^\top z_{\mathcal{T}} + \Delta$, where Δ is the approximation error of the projection, so

$$\begin{aligned}\mathcal{L}_{\text{Distill}} &= \|W_{\mathcal{T}}^\top z_{\mathcal{T}} - W_{\mathcal{S}}^\top (P^\top z_{\mathcal{T}} + \Delta)\|^2 \\ &= \|W_{\mathcal{T}}^\top z_{\mathcal{T}} - (PW_{\mathcal{S}})^\top z_{\mathcal{T}}\|^2 + \|W_{\mathcal{S}}^\top (P^\top z_{\mathcal{T}} - z_{\mathcal{S}})\|^2 \\ &\quad + \mathcal{C}(W_{\mathcal{S}}, z_{\mathcal{S}})\end{aligned} \tag{1}$$

where the cross-product term is defined as $\mathcal{C}(W_{\mathcal{S}}, z_{\mathcal{S}}) = 2\langle (PW_{\mathcal{S}})^\top z_{\mathcal{T}} - W_{\mathcal{T}}^\top z_{\mathcal{T}}, W_{\mathcal{S}}^\top (P^\top z_{\mathcal{T}} - z_{\mathcal{S}})\rangle$. This shows $\mathcal{L}_{\text{Distill}}$ jointly minimizes two objectives. The first trains a low-rank projection $PW_{\mathcal{S}}$ to replace $W_{\mathcal{T}}$, which we refer to as self-compression. The second minimizes the difference between the projected teacher embedding and $z_{\mathcal{S}}$. The cross-product term encourages both objectives simultaneously. To optimize this process, we first obtain the projection P^* by minimizing self-compression loss $\mathcal{L}_{\text{compress}} = \|W_{\mathcal{T}}^\top z_{\mathcal{T}} - (PU)^\top z_{\mathcal{T}}\|^2$, where $U \in \mathbb{R}^{d_s \times K}$ is the weight matrix of a learnable classifier with the same dimensionality as the

student classifier, as shown in step 1 of Figure 1(b). After that, we freeze P^* and optimize the student model by a joint loss with logit matching and feature matching: $\mathcal{L}_{\text{TSKD}} = \mathcal{L}_{\text{Distill}} + \lambda \|P^{*\top} z_{\mathcal{T}} - z_{\mathcal{S}}\|^2$, as shown in step 2 of Figure 1(b). Using the same expansion in Eq. 1, we can show that

$$\begin{aligned}\mathcal{L}_{\text{TSKD}} = &\|W_{\mathcal{T}}^\top z_{\mathcal{T}} - (P^* W_{\mathcal{S}})^\top z_{\mathcal{T}}\|^2 \\ &+ \|W_{\mathcal{S}}^\top W_{\mathcal{S}} + \lambda \mathbf{I}\|^2 \|P^{*\top} z_{\mathcal{T}} - z_{\mathcal{S}}\|^2 + \mathcal{C}(W_{\mathcal{S}}, z_{\mathcal{S}}).\end{aligned} \tag{2}$$

Here, P^* provides a well-initialized projection, while λ serves as a regularizer to stabilize the gradient of the second loss term, promoting the convergence of $z_{\mathcal{S}}$. Minimizing the objective drives $z_{\mathcal{S}}$ to learn the optimal compressed embedding $P^{*\top} z_{\mathcal{T}}$ for the task. The effect of varying λ is studied in Appendix A.2. We also analyze the impact of reusing U rather than randomly initializing $W_{\mathcal{S}}$ during training in Appendix A.3.

2.3. Implantable Neural Decoder

Our transformer-based decoder adopts a temporal linear-attention module with several efficiency-oriented modifications. We first compute spectral-temporal features of the neural signal through continuous wavelet transform (CWT), and tokenize the features in temporal order. These tokens then pass through a linear transformer and are averaged to generate the embedding z, which is then used for decoding through a linear layer. In the model, only the last linear layer contains a bias term, which facilitates subsequent quantization. The detailed architecture of IND is described in Appendix B.

2.4. Quantization-Aware Training and Integer-Only Inference

We develop an integer-only quantization-aware training (QAT) pipeline with learnable clipping ranges to ensure that IND can be deployed on-chip with minimal performance loss. Unlike previous integer-only quantization approaches that estimate clipping ranges statistically, such as I-ViT (Li & Gu, 2023), our QAT scheme introduces learnable clipping ranges that are jointly optimized with the network. During QAT, we update model weights as well as the clipping range α for activation values. We define a parameterized α_l for each layer l of the model, thus the scaling factor for quantization is $s_{l+1} = \frac{\alpha_l}{2^{b-1}-1}$ if the quantization bit-width is b. Inspired by PACT (Choi et al., 2018), we optimize α_l by defining gradients according to whether y_l^q falls out of the clipping range $[-\alpha_l, \alpha_l]$, which is:

$$\frac{\partial y_l^q}{\partial \alpha_l} = \frac{\partial y_l^q}{\partial y_l}\frac{\partial y_l}{\partial \alpha_l} = \begin{cases} -1, & x \in (-\infty, -\alpha_l) \\ 1, & x \in [\alpha_l, +\infty) \end{cases}. \tag{3}$$

Here, x denotes an element of y_l^q, and we set $\frac{\partial y_l^q}{\partial y_l} = 1$ by Straight-Through Estimator (STE) (Bengio et al., 2013). The implementation details are in Appendix D.1.

3. Experiments

We evaluate BrainDistill on two human ECoG datasets, **Human-C**, a clinical dataset collected for multi-class motor decoding, and **Human-D**, a dataset derived from AJILE12 dataset (Peterson et al., 2021) for naturalistic movement detection. We consider two training paradigms: directly training on $\mathcal{X}_{\text{recalib}}$ (**Scratch**) and distillation from pre-trained models (**Distillation**). In addition, we analyze the performance and energy consumption of the quantized IND, highlighting its feasibility for on-chip implementation. Detailed dataset and experimental settings are provided in A.4.

3.1. Baselines

We first compare our decoder trained from scratch with other established neural decoding models, including EEGNet (Lawhern et al., 2018), EEGConformer (Song et al., 2022), ATCNet (Altaheri et al., 2023), CTNet (Zhao et al., 2024), and LaBraM (Jiang et al., 2024) with pre-trained weights. Second, we compare our distillation method with other established logit-based, feature-based, and task-oriented knowledge distillation methods: KD (Hinton et al., 2015), SimKD (Chen et al., 2022), VkD (Miles et al., 2024), RdimKD (Guo et al., 2023), TOFD (Zhang et al., 2020) and TED (Liang et al., 2023). During distillation, we leverage two different pre-trained teacher models, LaBraM (5.8M parameters) for **Human-D** and a transformer with 100M parameters for **Human-C**.

3.2. Multi-Class Motor Decoding

We test both **Scratch** and **Distillation** on **Human-C** under intra-session and inter-session scenarios. Relative to the starting date of the first session, the six sessions were recorded on days 1, 4, 5, 6, 16, and 17, which are divided into 5 train-test splits. The notation $1-1$ represents training on the training split of session 1, and testing on the testing split of session 1. In experiments, we evaluate TSKD alone (+TSKD) and TSKD combined with the task loss, i.e., cross-entropy loss (+TSKD(CE)).

Table 1 shows the classification performance across different sessions of **Human-C**. We report both F1 score and the Average Recall. Given that different movements are highly imbalanced, Average Recall better reveals the model capability. According to the results, our decoder significantly outperforms other decoders when trained from scratch. Pre-trained LaBraM outperforms other baselines but is still worse than IND, which further validates the design of IND. With knowledge distillation, the performance continues to improve, and TSKD achieves the best performance in $4-4$, $4-5$ and $4-6$ over other KD baselines. In $1-1$ and $16-17$, TSKD(CE)

Table 1. Movement decoding performance on **Human-C**. The number of parameters is reported for each model. For each split, we report the mean and standard deviation of F1 and Avg Recall over three random initializations. Models are selected based on a validation set comprising 20% of the training data. Scores are reported as percentages. Our method and the best results are shown in **bold**, with the second-best results underlined.

SESSION	1-1		4-4		4-5		4-6		16-17	
METRICS	F1	Avg Recall	F1	Avg Recall	F1	Avg Recall	F1	Avg Recall	F1	Avg Recall
Teacher (100M)	77.6	67.4	73.3	57.2	74.5	60.8	77.3	62.5	80.6	58.9
Conformer (630K)	37.7 ±11.1	38.3 ±4.36	40.8 ±5.21	31.5 ±2.79	36.0 ±8.45	33.9 ±13.9	39.9 ±6.39	23.7 ±1.61	43.1 ±7.58	41.4 ±5.05
ATCNet (114K)	44.9 ±3.80	33.9 ±3.90	48.4 ±5.39	26.9 ±3.37	45.2 ±1.94	22.4 ±2.59	54.4 ±3.95	23.2 ±2.03	43.7 ±3.10	**53.8 ±1.29**
CTNet (152K)	46.1 ±0.40	30.5 ±1.11	49.1 ±2.25	27.2 ±1.57	42.2 ±2.25	23.5 ±1.45	54.3 ±0.77	21.6 ±2.04	44.3 ±3.20	49.7 ±3.29
EEGNet (5.5K)	29.0 ±16.8	39.2 ±4.57	42.9 ±1.89	28.8 ±3.75	41.8 ±4.76	25.1 ±7.98	42.8 ±6.38	17.5 ±1.59	54.1 ±9.01	43.7 ±6.43
LaBraM (5.8M)	56.3 ±2.44	40.3 ±3.91	59.4 ±2.68	38.7 ±1.53	58.5 ±1.26	35.1 ±1.86	66.2 ±1.04	36.0 ±2.27	**73.9 ±1.30**	46.8 ±5.29
IND (30K)	**69.1 ±1.64**	**56.6 ±4.11**	**64.9 ±3.45**	**44.9 ±3.78**	**61.5 ±2.20**	**37.1 ±5.68**	**70.6 ±0.97**	**44.2 ±5.26**	70.7 ±2.31	52.6 ±3.74
IND + KD	68.5 ±2.56	58.5 ±1.90	67.8 ±0.49	50.2 ±1.48	61.1 ±1.72	41.2 ±0.88	71.9 ±1.19	48.0 ±3.01	66.6 ±2.48	61.9 ±2.62
IND + SimKD	61.7 ±1.70	44.7 ±1.64	69.1 ±0.33	50.1 ±0.18	65.8 ±1.67	45.8 ±2.55	72.8 ±1.00	46.3 ±2.67	68.4 ±0.32	43.7 ±0.86
IND + VkD	70.9 ±1.32	60.7 ±1.49	67.6 ±0.80	50.1 ±0.54	62.2 ±2.68	41.5 ±3.00	71.7 ±1.66	49.7 ±2.10	69.9 ±2.35	63.4 ±2.49
IND + RdimKD	72.3 ±1.84	63.1 ±2.58	68.6 ±0.76	52.1 ±0.99	63.1 ±2.49	41.9 ±3.02	72.3 ±1.46	47.9 ±0.82	65.9 ±0.38	63.4 ±2.96
IND + TOFD	71.2 ±0.75	62.7 ±1.86	68.1 ±0.91	50.2 ±1.44	61.9 ±1.54	41.5 ±3.64	72.6 ±0.24	46.6 ±2.89	67.6 ±3.48	**66.9 ±1.72**
IND + TED	71.2 ±0.98	63.6 ±0.42	67.9 ±1.13	50.8 ±1.45	61.5 ±0.59	42.2 ±2.03	71.9 ±1.01	48.9 ±3.18	69.1 ±3.02	64.9 ±1.00
IND + TSKD(CE)	73.4 ±0.75	**65.2 ±5.67**	68.1 ±0.77	51.7 ±0.80	65.1 ±1.00	47.7 ±3.71	71.7 ±3.28	52.4 ±1.59	69.8 ±1.47	65.5 ±1.94
IND + TSKD	**73.9 ±1.64**	64.9 ±0.39	**73.3 ±0.63**	**58.0 ±0.98**	**75.0 ±0.20**	**60.9 ±0.61**	**77.3 ±0.34**	**59.6 ±0.70**	**71.4 ±1.59**	56.9 ±1.45

Table 2. Cross-subject decoding performance on **Human-D**. The teacher model is LaBraM fine-tuned on full training sets, and the student model is trained with 50% of the imbalanced training data. The average accuracy and AUROC over subjects 1,7, and 10 are reported. Scores are reported as percentages. Our method and the best results are shown in **bold**, and the second-best results are underlined.

METRIC	LABRAM	IND	**IND+TSKD**	IND+KD	IND+SIMKD	IND+VKD	IND+RDIMKD	IND+TOFD	IND+TED
AUROC	65.1 ± 1.80	49.2 ± 4.27	**55.8 ± 1.32**	53.2 ± 3.93	50.5 ± 7.04	48.7 ± 5.80	50.7 ± 3.85	51.7 ± 1.75	51.8 ± 2.26
Acc	57.6 ± 3.45	49.1 ± 4.41	**52.3 ± 2.08**	52.1 ± 1.58	50.0 ± 0.00	49.9 ± 3.13	49.8 ± 2.62	51.9 ± 1.75	50.4 ± 1.81

has the best performance, mainly because of class imbalance in the teacher predictions, especially in $16 - 17$.

3.3. Cross-Subject Motor Decoding

We further test how TSKD performs on cross-subject decoding using **Human-D**. The task is to detect the arm movements of each subject. For each subject, we evaluate the decoding performance on the last session using a decoder trained on recordings from all other subjects. We select subjects 1, 7, and 10, which exhibit large performance gaps between the teacher and student models, for evaluation. Table 2 shows that IND+TSKD achieves the highest cross-subject decoding performance compared to other KD baselines, demonstrating the effectiveness of the proposed method.

3.4. Feasibility of On-Chip Implementation

We quantize IND with an architecture tailored for **Human-C**. The decoder processes 320-dimensional input features, corresponding to either 5 wavelet features over 64 channels or 10 wavelet features over 32 channels, and has 2 attention layers with 32-dimensional embeddings. We compare full-precision and quantized model performances on **Human-C**. Table 3 shows that our QAT method outperforms I-ViT, and only loses $< 3\%$ of performance compared to full-precision model.

Beyond performance, we further evaluate IND on a prototype chip (Tan et al., 2026) fabricated in a 65-nm low-power process and occupying approximately 10 mm^2, featuring compute-in-memory (CIM) technology. To ensure the safety of the participant, the implanted BCI has to dissipate less than 0.4 mW / mm^2 of peak power to avoid heating surrounding brain tissue by more than 2°C, which causes cellular damage (Kim et al., 2007; Wolf, 2011; Karageorgos et al., 2020; Maguire et al., 2013). Table 4 summarizes the measured power consumption of different functional blocks. Since the model operates at an inference frequency of 10 Hz and each inference requires only 0.224 ms, we compute the runtime power (last column of Table 4) based

Table 3. Comparison of decoding performance on **Human-C** between the full precision model (FP32), I-ViT (W8A8), and our quantization method (W8A8). For each split, we report F1 and Avg Recall as mean ± std over three random seeds.

SESSION	METRIC	FP32	I-VIT	OURS
1–1	F1	73.9 ±1.64	72.4 ±0.14	**72.4 ±0.42**
	Avg Recall	64.9 ±0.39	61.1 ±1.21	**62.3 ±0.81**
4–4	F1	73.3 ±0.63	72.7 ±0.88	**73.2 ±0.56**
	Avg Recall	58.0 ±0.98	57.1 ±1.42	**57.4 ±0.67**
4–5	F1	75.0 ±0.20	71.9 ±1.49	**73.1 ±0.19**
	Avg Recall	60.9 ±0.61	55.5 ±2.47	**58.3 ±2.88**
4–6	F1	77.3 ±0.34	74.9 ±2.17	**77.5 ±0.35**
	Avg Recall	59.6 ±0.70	57.7 ±1.71	**58.1 ±0.61**
16–17	F1	71.4 ±1.59	73.8 ±0.44	**73.8 ±1.18**
	Avg Recall	56.9 ±1.45	52.2 ±5.11	**56.3 ±0.99**

on $P_{\text{runtime}} = P_{\text{leakage}} + P_{\text{dynamic}} \times 2.24 \times 10^{-4} \times 10$. The results show that IND consumes only 47 μW (4.7 μW / mm^2) on the prototype chip, which is 85 times smaller than the suggested power limit of 0.4 mW / mm^2. In conclusion, the decoding performance of quantized IND and the power measurement results on the prototype chip indicates that it is applicable to deploy the quantized IND on a cortical BCI device.

Table 4. Power breakdown by building block on the prototype chip. All values are reported in μW.

POWER GROUP	LEAKAGE (μW)	DYNAMIC (μW)	RUNTIME (μW)	%
Memory	3.058×10^0	3.124×10^3	1.006×10^1	21.56%
Logic	3.073×10^1	1.312×10^2	3.102×10^1	66.50%
Register	1.131×10^0	1.819×10^0	1.135×10^0	2.43%
Clock	8.378×10^{-2}	4.627×10^2	1.120×10^0	2.40%
CIM	3.050×10^0	1.200×10^2	3.319×10^0	7.11%
Summary	–	–	4.665×10^1	100.00%

4. Conclusion

In this paper, we introduce BrainDistill, a motor decoding pipeline that integrates a task-specific knowledge distillation (TSKD) method with a transformer-based implantable neural decoder (IND). To ensure deployability under strict hardware constraints, we introduce a quantization-aware training strategy that learns activation clipping ranges and enables efficient integer-only inference. Extensive experiments demonstrate that BrainDistill not only outperforms prior neural decoders but also surpasses alternative distillation baselines with limited recalibration data. It also generalizes better across sessions and subjects. With its compact design, strong decoding performance, and chip-ready implementation, BrainDistill takes a practical step toward clinical deployment of neural network–based BCI systems.

References

Altaheri, H., Muhammad, G., and Alsulaiman, M. Physics-informed attention temporal convolutional network for eeg-based motor imagery classification. *IEEE Transactions on Industrial Informatics*, 19(2):2249–2258, 2023. doi: 10.1109/TII.2022.3197419.

Bengio, Y., Léonard, N., and Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Bundy, D. T., Pahwa, M., Szrama, N., and Leuthardt, E. C. Decoding three-dimensional reaching movements using electrocorticographic signals in humans. *Journal of neural engineering*, 13(2):026021, 2016.

Chen, D., Mei, J.-P., Zhang, H., Wang, C., Feng, Y., and Chen, C. Knowledge distillation with the reused teacher classifier. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11933–11942, 2022.

Choi, J., Wang, Z., Venkataramani, S., Chuang, P. I.-J., Srinivasan, V., and Gopalakrishnan, K. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018.

Cruz-Garza, J. G., Hernandez, Z. R., Nepaul, S., Bradley, K. K., and Contreras-Vidal, J. L. Neural decoding of expressive human movement from scalp electroencephalography (eeg). *Frontiers in human neuroscience*, 8:188, 2014.

Duan, X., Xie, S., Xie, X., Meng, Y., and Xu, Z. Quadcopter flight control using a non-invasive multi-modal brain computer interface. *Frontiers in neurorobotics*, 13: 23, 2019.

Eliseyev, A., Auboiroux, V., Costecalde, T., Langar, L., Charvet, G., Mestais, C., Aksenova, T., and Benabid, A.-L. Recursive exponentially weighted n-way partial least squares regression with recursive-validation of hyper-parameters in brain-computer interface applications. *Scientific reports*, 7(1):16281, 2017.

Flint, R. D., Tate, M. C., Li, K., Templer, J. W., Rosenow, J. M., Pandarinath, C., and Slutzky, M. W. The representation of finger movement and force in human motor and premotor cortices. *Eneuro*, 7(4), 2020.

Guo, Y., He, Y., Li, X., Qin, H., Pham, V. T., Zhang, Y., and Liu, S. Rdimkd: Generic distillation paradigm by dimensionality reduction. *arXiv preprint arXiv:2312.08700*, 2023.

Hammer, J., Fischer, J., Ruescher, J., Schulze-Bonhage, A., Aertsen, A., and Ball, T. The role of ecog magnitude and phase in decoding position, velocity, and acceleration during continuous motor behavior. *Frontiers in neuroscience*, 7:200, 2013.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Hu, X., Cheng, Y., Yang, D., Yuan, Z., Yu, J., Xu, C., and Zhou, S. I-llm: Efficient integer-only inference for fully-quantized low-bit large language models. *arXiv preprint arXiv:2405.17849*, 2024.

Jiang, W., Zhao, L., and liang Lu, B. Large brain model for learning generic representations with tremendous EEG data in BCI. In *The Twelfth International Conference*

on Learning Representations, 2024. URL https://openreview.net/forum?id=QzTpTRVtrP.

Karageorgos, I., Sriram, K., Veselỳ, J., Wu, M., Powell, M., Borton, D., Manohar, R., and Bhattacharjee, A. Hardware-software co-design for brain-computer interfaces. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pp. 391–404. IEEE, 2020.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Kim, S., Tathireddy, P., Normann, R. A., and Solzbacher, F. Thermal impact of an active 3-d microelectrode array implanted in the brain. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(4):493–501, 2007.

Kim, S., Gholami, A., Yao, Z., Mahoney, M. W., and Keutzer, K. I-bert: Integer-only bert quantization. In *International conference on machine learning*, pp. 5506–5518. PMLR, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., and Lance, B. J. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15 (5):056013, 2018.

Lee, N., Barmpas, K., Panagakis, Y., Adamos, D., Laskaris, N., and Zafeiriou, S. Are large brainwave foundation models capable yet ? insights from fine-tuning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=J5SbLoq7Uv.

Li, Z. and Gu, Q. I-vit: Integer-only quantization for efficient vision transformer inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17065–17075, 2023.

Liang, C., Zuo, S., Zhang, Q., He, P., Chen, W., and Zhao, T. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, pp. 20852–20867. PMLR, 2023.

Lin, Y., Zhang, T., Sun, P., Li, Z., and Zhou, S. Fq-vit: Post-training quantization for fully quantized vision transformer. *arXiv preprint arXiv:2111.13824*, 2021.

Lorach, H., Galvez, A., Spagnolo, V., Martel, F., Karakas, S., Intering, N., Vat, M., Faivre, O., Harte, C., Komi, S., et al. Walking naturally after spinal cord injury using a brain–spine interface. *Nature*, 618(7963):126–133, 2023.

Ma, X., Ma, C., Huang, J., Zhang, P., Xu, J., and He, J. Decoding lower limb muscle activity and kinematics from cortical neural spike trains during monkey performing stand and squat movements. *Frontiers in Neuroscience*, 11:44, 2017.

Maguire, Y. G., Shapiro, M. G., Cybulski, T. R., Glaser, J. I., Amodei, D., Stranges, P. B., Kalhor, R., Dalrymple, D. A., Seo, D., Alon, E., et al. Physical principles for scalable neural recording. *Frontiers in computational neuroscience*, 7:61935, 2013.

Mestais, C. S., Charvet, G., Sauter-Starace, F., Foerster, M., Ratel, D., and Benabid, A. L. Wimagine: wireless 64-channel ecog recording implant for long term clinical applications. *IEEE transactions on neural systems and rehabilitation engineering*, 23(1):10–21, 2014.

Miles, R. and Mikolajczyk, K. Understanding the role of the projector in knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4233–4241, 2024.

Miles, R., Elezi, I., and Deng, J. Vkd: Improving knowledge distillation using orthogonal projections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15720–15730, 2024.

Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., Trautmann, E. M., Kaufman, M. T., Ryu, S. I., Hochberg, L. R., et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

Peterson, S. M., Steine-Hanson, Z., Davis, N., Rao, R. P., and Brunton, B. W. Generalized neural decoders for transfer learning across participants and recording modalities. *Journal of Neural Engineering*, 18(2):026014, 2021.

Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., and Ball, T. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human Brain Mapping*, aug 2017. ISSN 1097-0193. doi: 10.1002/hbm.23730. URL http://dx.doi.org/10.1002/hbm.23730.

Shao, W., Meng, W., Zuo, J., Li, X., and Ming, D. Opportunities and challenges of brain-on-a-chip interfaces. *Cyborg and Bionic Systems*, 6:0287, 2025.

Śliwowski, M., Martin, M., Souloumiac, A., Blanchart, P., and Aksenova, T. Decoding ecog signal into 3d hand translation using deep learning. *Journal of neural engineering*, 19(2):026023, 2022.

Song, Y., Zheng, Q., Liu, B., and Gao, X. Eeg conformer: Convolutional transformer for eeg decoding and visualization. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:710–719, 2022.

Tan, F., Berner, R., Zhao, Z., Xie, Y., Joudi, A., Fu, Y., Sakr, I., Collin, T., Rouanne, V., Lorach, H., Un, K.-F., Mak, P.-I., Bloch, J., Courtine, G., and Shoaran, M. A 64-channel, low-power end-to-end transformer-based bci soc for efficient motion intention decoding. In *2026 Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, pp. 1–3, 2026.

Volkova, K., Lebedev, M. A., Kaplan, A., and Ossadtchi, A. Decoding movement from electrocorticographic activity: a review. *Frontiers in neuroinformatics*, 13:74, 2019.

Wang, C., Subramaniam, V., Yaari, A. U., Kreiman, G., Katz, B., Cases, I., and Barbu, A. Brainbert: Self-supervised representation learning for intracranial recordings. *arXiv preprint arXiv:2302.14367*, 2023.

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

Wolf, P. D. Thermal considerations for the design of an implanted cortical brain–machine interface (bmi). 2011.

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

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

Zhang, L., Shi, Y., Shi, Z., Ma, K., and Bao, C. Task-oriented feature distillation. *Advances in Neural Information Processing Systems*, 33:14759–14771, 2020.

Zhao, W., Jiang, X., Zhang, B., Xiao, S., and Weng, S. Ctnet: a convolutional transformer network for eeg-based motor imagery classification. *Scientific reports*, 14(1): 20237, 2024.

Zhou, J., Zhu, K., and Wu, J. All you need in knowledge distillation is a tailored coordinate system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22946–22954, 2025.

A. Appendix

A.1. Related Work

Neural Decoders for ECoG Movement Decoding. Prior to our work, Hammer et al. (2013) showed that time–frequency features of ECoG signals can decode continuous movement trajectories, speed, and acceleration, and proposed a multiple linear regression decoder using short-time Fourier transform (STFT) features. Flint et al. (2020) demonstrated that ECoG enables precise decoding of finger movements and forces by leveraging LFADS (Pandarinath et al., 2018), a sequential deep learning model. Eliseyev et al. (2017) introduced REW-NPLS, a multilinear regression method for decoding upper-limb movements from continuous wavelet transform (CWT) features, while Śliwowski et al. (2022) replaced the multilinear model with CNN- and LSTM-based architectures, achieving improved results. These studies establish the effectiveness of time–frequency features, especially CWT features, for ECoG-based movement decoding; however, transformer-based decoders that directly tokenize CWT features have not yet been well explored.

Knowledge Distillation. To improve distillation from teacher models, Chen et al. (2022) projected student features onto the teacher space, allowing the student to reuse the teacher's classifier and achieve comparable performance. Miles & Mikolajczyk (2024) further analyzed the role of projectors in distillation and found that simpler projections improve the correlation between student and projected features, leading to better results with orthogonal projection (Miles et al., 2024). Beyond learnable projections, Zhou et al. (2025); Guo et al. (2023) directly applied PCA to teacher features and aligned student features with the projected ones. However, these approaches preserve teacher information indiscriminately, which may not be optimal for specific tasks. For task-oriented distillation, Zhang et al. (2020) proposed logit distillation with additional FC layers to implicitly align features, and Liang et al. (2023) aligned teacher and student intermediate features using task filters at each layer. Such methods, however, require architectural similarity between teacher and student models and do not guarantee the effectiveness of the implicit alignment. In contrast, our TSKD explicitly compresses task-relevant information from teacher embeddings and quantifies the projection quality using a novel metric. Besides distillation-based methods, recent work Lee et al. (2025) shows that pre-trained neural foundation models can achieve downstream performance comparable to full fine-tuning while requiring only a small number of trainable parameters through LoRA (Hu et al., 2022). However, such parameter-efficient fine-tuning (PEFT) approaches do not reduce the model size or computational footprint at inference time, making them unsuitable for implantable decoding scenarios.

Integer-Only Quantization. To enable hardware implementation of neural networks, integer-only quantization methods have been proposed. Unlike conventional quantization, these methods ensure that all operations are executed entirely in the integer domain, which requires both non-linear functions and scaling factors to be quantized. For example, I-BERT (Kim et al., 2021) and FQ-ViT (Lin et al., 2021) replace non-linear operations in transformers, while I-ViT (Li & Gu, 2023) further fully quantizes the entire computational graph of ViTs using integer arithmetic and bit-shifting. Building on this line of work, I-LLM (Hu et al., 2024) introduces DI-MatMul, which dynamically adjusts the clipping range for each token. However, none of these approaches integrate learnable quantization clipping ranges with integer-only inference in a quantization-aware training framework. This integration is particularly crucial for small models, where QAT is essential to jointly optimize model weights and clipping ranges to mitigate quantization errors. To the best of our knowledge, our approach is the first to combine learnable clipping ranges with integer-only inference in a QAT framework.

A.2. TSKD performance with varying λ

We train TSKD on IND using λ from $\{0.1, 0.5, 1, 1.5, 2\}$ and test on session 4-4, 4-5, and 4-6 of **Human-C**. The performance is illustrated in Figure 2. Results show that the decoder performance reaches a peak between $\lambda = 0.5$ and $\lambda = 1$, which also supports that λ works as a regularization term in the TSKD loss.

A.3. Reusing the student classifier U in TSKD

We tested the use of the same classifier U during phase 2 of TSKD distillation on 4-4, 4-5, and 4-6 of **Human-C**, and the results in Table 5 show that the randomly initialized classifier performs better. One hypothesis is that a well-optimized projection P already provides a good initialization for TSKD. In contrast, a trained classifier increases the norm of the feature distillation loss term. This harms the stability of the joint objective of $\mathcal{L}_{\text{TSKD}}$ according to the analysis from Appendix A.2. To validate this hypothesis, we compare the Frobenius norms of a trained classifier and a randomly initialized classifier on session 4 of Human-C. The trained classifier exhibits a Frobenius norm of 2.02, whereas the random classifier has a significantly smaller norm of 1.45. This empirical evidence supports our hypothesis.

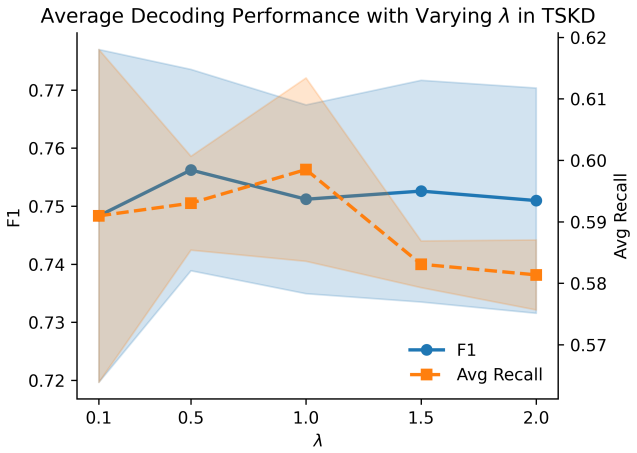

Figure 2. F1 and Average Recall with different λ. The blue and orange lines show the mean values of F1 and Average Recall over 3 sessions respectively, and the shaded area represents the standard deviation.

Table 5. Decoding performance on **Human-C** when using the same classifier and the randomly initialized classifier during TSKD training.

Classifier	4-4		4-5		4-6	
	F1	Avg Recall	F1	Avg Recall	F1	Avg Recall
U (Trained)	72.5	55.6	73.8	57.8	76.3	54.9
W_S (Randomly Initialized)	**73.2**	**57.8**	**75.2**	**61.4**	**77.2**	**60.3**

A.4. Experimental Settings

Our experiments are conducted on a workstation with an NVIDIA RTX4090 GPU (24GB), Intel i9-10900KF CPU, and 32GB RAM, using PyTorch 2.4.1 with CUDA 12.4 and Python 3.10. We adopt the Adam optimizer (Kingma & Ba, 2014). IND is configured with an embedding dimension of 32, an FFN dimension of 128, and 2 attention layers. For EEGConformer, ATCNet, CTNet, and EEGNet, we follow the Braindecode library implementation (Schirrmeister et al., 2017). For KD, VkD, and TSKD (CE), we set the weights of distillation loss and task loss both to 0.5. The λ of TSKD is by default 1.0.

B. IND architecture

B.1. Tokenization

For a given sample $x \in \mathbb{R}^{T \times C}$, we first apply z-score normalization over each channel using the pre-computed statistics. Then, we compute CWT using complex Morlet wavelets with N different center frequencies, and obtain the spectral-temporal map $x_f \in \mathbb{R}^{N \times T \times C}$. By applying average pooling along the temporal dimension, we obtain a new $x_f \in \mathbb{R}^{N \times L \times C}$, where $L \ll T$. The resulting feature is then tokenized into $x_{\text{input}} \in \mathbb{R}^{L \times (C*N)}$, where each token contains both channel and frequency information at each time step. The token is projected to embedding space through $x_{\text{emb}} = x_{\text{input}} W + \text{Pos}$, where $x_{\text{emb}} \in \mathbb{R}^{L \times d}$, $W \in \mathbb{R}^{(C*N) \times d}$ is a linear transform, and $\text{Pos} \in \mathbb{R}^{L \times d}$ is a positional embedding. Using CWT features as tokens offers clear advantages: each token encodes explicit time–frequency information within a specific temporal segment and frequency band, making the features directly interpretable. This is particularly valuable for ECoG, as CWT precisely reveals the frequency bands of neural activity. In contrast, convolutional tokenization provides only implicit interpretability, as its time–frequency representation depends on kernel size and learned weights.

B.2. Linear Attention

We adapt the linear attention method from (Katharopoulos et al., 2020) for neural data and the need for quantization. In the original linear attention formulation, K and V are first multiplied because L can be much larger than d, but for neural data, L is short as only recent neural data is meaningful for predicting the movement. Therefore, we compute linear attention with the same order as softmax attention: $V_i' = \frac{\sum_{j=1}^{N} \phi(Q_i)^\top \phi(K_j) V_j}{\sum_{j=1}^{N} \phi(Q_i)^\top \phi(K_j)}$. V_i' is the i-th output token, and $\phi(\cdot)$ is the activation

function, where we use ReLU for its quantization-friendly properties.

C. Datasets

C.1. Human-C

Human-C is a dataset for multi-class upper-limb movement classification collected in the scope of a clinical trial aiming to restore upperlimb motor function after spinal cord injury. The participant was implanted with ECoG (Mestais et al., 2014) and performed movement attempts involving hands, wrists, elbows and shoulders. Raw ECoG signals were initially sampled at 586Hz over 32 channels simultaneously. Signals were resampled at 500Hz and preprocessed as 1.5s windows with 100ms strides. The label was created based on the movement type of the last 0.1 second within each window. Each window is converted to 10 tokens after CWT tokenization.

We present the categories and number of samples in each session in **Human-C** in Table 6. In total, we cover 6 different movement states, including rest, shoulder flexion, elbow extension, forearm pronation, hand opening, and hand closing, but they differ across sessions. It is clear that the rest state always has the largest sample size, and different movements are also not balanced. This implies the difficulty of directly training from scratch on this dataset. We extract 10 different wavelet features, based on the following center frequencies: $\{10\text{Hz}, 30\text{Hz}, 50\text{Hz}, 60\text{Hz}, 70\text{Hz}, 80\text{Hz}, 90\text{Hz}, 120\text{Hz}, 150\text{Hz}, 200\text{Hz}\}$. All models are trained with $\text{lr} = 3e-4$ and $\text{weight_decay} = 1e-4$.

Table 6. Sample numbers of different categories in **Human-C**, each sample represents a 1.5-second neural recording.

SETTING	SESSION	REST	SHOULDER FLEXION	ELBOW EXTENSION	WRIST PRONATION	HAND OPEN	HAND CLOSE
Train	1	2808		762	598	388	
	4	5487	388	927	796	705	921
	16	14130		1105		877	788
Test	1	776		193	178	197	
	4	1380	211	214	226	186	195
	5	1260	158	244	214	200	193
	6	1812	144	176	173	226	151
	17	1292		125		188	95

C.2. Human-D

Each sample from **Human-D** is a segment of 64 channels and 2-second recording of 500Hz. Therefore, we create 10 tokens, each representing 0.2 seconds, and extract 7 wavelet features: $\{10\text{Hz}, 20\text{Hz}, 40\text{Hz}, 60\text{Hz}, 80\text{Hz}, 100\text{Hz}, 125\text{Hz}\}$. All of the models are trained for 30 epochs with $\text{lr} = 3e-3$. The sample numbers of the training set and test set of **Human-D** are listed in Table 7.

Table 7. Sample numbers of different categories in training and testing set of **Human-D**, each sample represents a 2-second neural recording.

SPLITS	CLASS	1	2	3	4	5	6	7	8	9	10	11	12
Train	Arm Rest	386	128	392	124	62	551	399	535	101	366	564	594
	Arm Move	370	134	379	120	56	551	404	540	107	379	575	615
Test	Arm Rest	186	45	102	115	129	180	173	178	21	154	184	191
	Arm Move	186	45	102	115	129	180	173	178	21	154	184	191

D. Quantization and Power Estimation

D.1. Quantization Aware Training

First, the input neural signal is quantized through a lookup table that maps x_t to unsigned 10-bit data. This simulates the function of a 10-bit analog-digital converter (ADC). The Wavelet kernels use 16-bit symmetric uniform quantization; wavelet features are 8-bit. In the decoder, all weights and activations are 8-bit, and biases are 32-bit. Given the activation y_l and its scale s_l, the quantized activation will be computed through:

$$y_l^q = \left\lfloor y_l * \frac{s_l}{s_{l+1}} \right\rceil * s_{l+1}, \tag{4}$$

where s_{l+1} is computed by α_l. We implement LayerNorm using the same approach as I-ViT (Li & Gu, 2023), and perform an approximation for the division operation in linear attention. In order to avoid any floating-point computation at inference time, $\frac{s_l}{s_{l+1}}$ will be converted to a dyadic number $\frac{s_l}{s_{l+1}} \approx \frac{m}{2^e}$, where m and e are 16-bit integers. Although α can be updated during training, its initialization remains important. We estimate it from the activation distribution of the training set on the full-precision model. During QAT, the quantized decoder is initialized from the full-precision model and then further optimized using either the task loss or the distillation loss. For all weights quantization in the decoder, we apply channel-wise scaling by estimating the range of each weight matrix, as in I-ViT (Li & Gu, 2023). For activations, we estimate a fixed range for non-essential operations, and apply learnable clipping on value-sensitive operations. Table 8 illustrates the details of the clipping range in each operation for **Human-C**.

Table 8. Clipping ranges of different activations in the quantized decoder. $max(|\text{min}|, |\text{max}|)$ means the matrix range is estimated to be the clipping range, and other values mean that α is initialized and it is updated during QAT.

OPERATION	CLIPPING RANGE	OPERATION	CLIPPING RANGE
Wavelet	1.0	Layer 2 Q projection	7.0
Input Layer	$max(\|\text{min}\|, \|\text{max}\|)$	Layer 2 K projection	5.0
Positional Encoding	$max(\|\text{min}\|, \|\text{max}\|)$	Layer 2 V projection	5.0
Layer 1 Q projection	1.5	Layer 2 QK multiplication	20.0
Layer 1 K projection	2.0	Layer 2 QK normalizer	150.0
Layer 1 V projection	1.5	Layer 2 Attention output	100.0
Layer 1 QK multiplication	3.0	Layer 2 Out Projection	1.5
Layer 1 QK normalizer	25.0	Layer 2 Residual & LayerNorm	$max(\|\text{min}\|, \|\text{max}\|)$
Layer 1 Attention output	10.0	Layer 2 Feedforward Layer	$max(\|\text{min}\|, \|\text{max}\|)$
Layer 1 Out Projection	1.0	Classifier	$max(\|\text{min}\|, \|\text{max}\|)$
Layer 1 Residual & LayerNorm	$max(\|\text{min}\|, \|\text{max}\|)$		
Layer 1 Feedforward Layer	$max(\|\text{min}\|, \|\text{max}\|)$		

In order to compute linear attention, we perform a quantized division with integer-only operations. For two quantized matrix A and B and their scales s_a and s_b, the quantized division generates $O * s_o \approx \frac{A * s_a}{B * s_b}$, where $O * s_o$ is an approximation of the floating-point result of the division. We first obtain s_o from the output range of the full-precision model, and then O only needs to be computed from $\frac{A * s_a}{B * s_b * s_o}$. Dividing two positive integers without remainder will leave quantization error up to 1, and when the numerator is smaller, the result is 0 which is meaningless. In order to shrink this error, we perform the following division:

$$O \approx \frac{A * s_a * 2^e}{B * s_b * s_o} \gg e. \tag{5}$$

Scaling numerator with 2^e ensures it is larger than the denominator, and also reduces the quantization error of each element to lower than 2^{-e}. In practice, we select $e = 12$.