Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Early Exit as an Energy Strategy for Motor Decoding on Edge Hardware

McKenzie Izev

mckenzieizev@gmail.com

AIT Austrian Institute of Technology

Research Article

Keywords:

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11708453/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Early Exit as an Energy Strategy for Motor Decoding on Edge Hardware

Mckenzie Izev

AIT Austrian Institute of Technology (AIT)

Abstract

Motor decoding on edge hardware enables real-time, closed-loop neuroprosthetic and human–machine interface applications, but the energy cost of running deep decoding models continuously on battery-constrained devices remains a critical barrier. Conventional approaches compute a full model inference for every neural window, even though neural activity during movement is highly non-stationary and many windows are trivially separable. This work investigates early exit—dynamic, input-adaptive computation in which inference terminates at an intermediate layer when a confidence criterion is met—as an energy strategy for motor decoding on edge hardware. We formulate early exit for both regression-based kinematic decoding and classification-based movement-state decoding, and introduce energy-aware exit criteria that trade decoding accuracy against the estimated cost of continued computation. Using representative motor decoding models (MLP, CNN, and recurrent architectures) evaluated on standard neural datasets, we characterize the accuracy–energy Pareto frontier across exit thresholds, layer depths, and hardware configurations. Results show that early exit reduces inference energy substantially while preserving decoding performance, with the largest savings on windows that are easy to decode, and that energy savings scale with the skewness of the per-window difficulty distribution. We further show that naive confidence thresholds are suboptimal on energy grounds and that calibration-aware exit policies improve the energy–accuracy trade-off. Finally, we discuss the implications of dynamic computation for latency, real-time constraints, and on-device deployment, and identify open challenges including distribution shift, temporal correlation of exit decisions, and hardware support for branching inference. Our findings position early exit as a practical, model-agnostic lever for energy-efficient motor decoding at the edge.

Keywords: early exit; dynamic neural networks; motor decoding; brain–computer interfaces; energy-efficient inference; edge computing; adaptive computation; neural signal processing; embedded machine learning; real-time decoding

1. Introduction

The ambition to restore movement and communication for individuals with severe motor impairment has driven decades of research in motor decoding, the process by which neural signals are translated into control commands for external devices (Wolpaw et al., 2002). Brain-computer interfaces (BCIs) for motor restoration span invasive and non-invasive modalities, from intracortical recordings that drive robotic arms (Hochberg et al., 2012) to electroencephalography (EEG)-based systems that allow users to spell or navigate virtual environments through imagined movement (Pfurtscheller and Neuper, 2001). Across this diverse landscape, a common computational challenge has emerged: the decoding model must operate under severe resource constraints while maintaining the reliability and latency characteristics that closed-loop interaction demands (Lebedev and Nicolelis, 2017). This tension between computational sophistication and energy efficiency has become particularly acute as the field moves from laboratory demonstrations toward wearable and implantable systems intended for continuous, everyday use (He et al., 2018).

The conventional response to this tension has been to design increasingly compact models—lightweight convolutional networks (Lawhern et al., 2018), spiking neural networks (Yin et al., 2021), and quantized architectures (Wu et al., 2018)—that fit within the memory and power budgets of embedded processors. These efforts have yielded impressive results, with modern lightweight decoders achieving classification performance competitive with larger models while reducing computational cost by factors of forty to eighty (Yin et al., 2021). Yet a fundamental inefficiency persists in virtually all deployed motor decoding pipelines: every neural window, regardless of its complexity or informativeness, is subjected to the same full-model inference. A window of neural activity recorded during a period of quiet rest receives identical computational treatment to a window captured during a challenging transition between movement states (Sussillo et al., 2015). This uniform allocation of computation ignores the highly non-stationary nature of neural activity and the substantial variability in decoding difficulty across windows (Shenoy et al., 2013).

Early exit, a form of dynamic, input-adaptive computation, offers a direct response to this inefficiency (Teerapittayanon et al., 2016). In an early-exit architecture, intermediate classifiers are attached to multiple depths of a neural network, and inference terminates at an earlier exit when a confidence criterion is satisfied, bypassing the remaining layers (Matsubara et al., 2022). The approach was formalized in the mid-2010s through architectures such as BranchyNet (Teerapittayanon et al., 2016) and has since been explored extensively in computer vision and natural language processing, where it has been shown to reduce inference energy and latency while preserving accuracy on the majority of inputs (Han et al., 2022). The central premise is simple yet powerful: many inputs are "easy" and can be correctly classified by a shallower sub-network, and the computational resources that would have been spent processing them through the full model can be conserved (Laskaridis et al., 2021). For energy-constrained edge hardware, this conservation translates directly into extended battery life and reduced thermal load (Dimitriou et al., 2023).

The application of early exit to motor decoding is not merely an extension of existing work in other domains. Motor decoding from neural signals presents distinctive characteristics that shape both the opportunity and the design space for dynamic computation (Suppiah et al., 2026). First, neural activity during movement is temporally correlated and exhibits structured non-stationarity: periods of stable motor output alternate with transitions, and decoding difficulty varies systematically with the phase of movement and the cognitive state of the user (Churchland et al., 2012). This structure suggests that a well-designed exit policy could exploit temporal context to anticipate when early termination is likely to be safe. Second, motor decoding encompasses both classification tasks (e.g., distinguishing imagined movement classes) and regression tasks (e.g., predicting continuous kinematics), and the early-exit framework must be adapted to the confidence estimation challenges inherent in each (Inoue et al., 2017). Third, the consequences of an erroneous early exit differ from those in typical computer vision applications: a premature termination that yields an incorrect control signal in a closed-loop BCI may have safety implications or disrupt the user's sense of agency, elevating the importance of calibrated confidence estimation (Guo et al., 2017).

This paper investigates early exit as an energy strategy for motor decoding on edge hardware. Our investigation is guided by three research questions. First, what is the achievable energy–accuracy trade-off when early-exit mechanisms are integrated into representative motor decoding architectures? Second, how do the characteristics of neural signals—their noise, non-stationarity, and temporal structure—affect the design and performance of exit criteria? Third, what are the practical considerations for deploying early-exit decoders on resource-constrained hardware, including the interplay between energy savings, latency constraints, and the calibration of confidence estimates?

To address these questions, we conduct a systematic study spanning model architectures, exit criteria, and evaluation metrics. We consider three representative model families that span the design space of modern motor decoders: multilayer perceptrons (MLPs), which remain common in real-time BCI systems due to their low overhead (Schirrmeister et al., 2017); convolutional neural networks (CNNs), which have become the standard for EEG-based decoding (Lawhern et al., 2018); and recurrent architectures, which capture temporal dependencies in neural sequences (Sussillo et al., 2015). For each architecture, we introduce intermediate exit points and evaluate a range of exit criteria, from simple confidence thresholds to entropy-based measures and energy-aware policies that explicitly account for the cost of continued computation (Wang et al., 2019; Xin et al., 2020). We evaluate performance on publicly available neural datasets that encompass both classification-based and regression-based motor decoding tasks, and we characterize the accuracy–energy Pareto frontier across exit thresholds and hardware configurations.

Our contributions are as follows. We provide the first systematic evaluation of early exit for motor decoding, establishing empirical baselines for the energy savings achievable across model families and task types. We introduce energy-aware exit criteria that improve upon naive confidence thresholding by incorporating the estimated cost of continued inference into the exit decision (Laskaridis et al., 2021). We analyze the relationship between per-window decoding difficulty and exit behavior, showing that the energy savings from early exit scale with the

skewness of the difficulty distribution and that temporal correlation in neural signals creates opportunities for anticipatory exit policies. We examine the calibration of confidence estimates in the motor decoding context (Guo et al., 2017) and demonstrate that uncalibrated confidence leads to suboptimal energy–accuracy trade-offs. Finally, we discuss the implications of our findings for the design of practical early-exit decoders and identify open challenges, including distribution shift across sessions (Shenoy et al., 2013), the temporal correlation of exit decisions, and the need for hardware support for branching inference (Putic et al., 2020).

The remainder of this paper is organized as follows. Section 2 reviews the literature on motor decoding, energy-efficient neural inference, and early-exit architectures, positioning our work within the broader research landscape. Section 3 describes our methodology, including the model architectures, datasets, exit criteria, and evaluation protocol. Section 4 presents the results of our experiments, characterizing the accuracy–energy trade-offs and the factors that influence them. Section 5 discusses the implications of these findings, their limitations, and directions for future work. Section 6 concludes.

2. Literature Review

2.1 Motor Decoding on Edge Hardware

The computational demands of motor decoding have grown substantially as the field has transitioned from linear classifiers and hand-engineered features toward deep learning approaches (Lotte et al., 2018). Early BCI systems relied on methods such as linear discriminant analysis, common spatial patterns, and support vector machines (Blankertz et al., 2008), which imposed modest computational burdens and could be implemented on low-power microcontrollers. The introduction of deep neural networks brought improvements in decoding accuracy, particularly for challenging tasks such as cross-session generalization and decoding from low-quality consumer-grade EEG (Schirrmeister et al., 2017), but at the cost of increased computational complexity (Lawhern et al., 2018). A typical CNN-based EEG decoder may involve hundreds of thousands of parameters and millions of multiply-accumulate operations per inference, a workload that strains the capabilities of wearable and implantable hardware (Wu et al., 2018).

The response of the research community has been multifaceted. At the model level, lightweight architectures have been designed specifically for EEG processing. EEGNet, a compact convolutional network, remains a widely used benchmark due to its favorable accuracy-to-parameter ratio (Lawhern et al., 2018). More recent work has explored spiking neural networks (SNNs) that exploit event-driven computation to reduce energy consumption, with convolutional SNN architectures achieving performance within a few percent of EEGNet while reducing computational cost by a factor of forty to eighty (Yin et al., 2021). At the hardware level, field-programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs) have been deployed to accelerate motor imagery classification with minimal power consumption and reduced latency (Wu et al., 2018). These hardware solutions often incorporate aggressive quantization and parallelization to meet the real-time constraints of closed-loop BCI (Biggs et al., 2023).

A parallel line of work has focused on reducing the input data volume rather than the model complexity. Electrode selection algorithms dynamically identify a subset of channels that carry the most discriminative information for a given task, allowing the decoder to operate on a fraction of the available signal and thereby reducing both acquisition and computation costs (Lotte et al., 2018). Feature selection and reduction techniques similarly eliminate or compress the feature representation before it reaches the decoder (Blankertz et al., 2008). These approaches are complementary to model-level optimizations and have been shown to achieve substantial energy savings with minimal accuracy loss. However, they share a common limitation: the selection is typically static or slowly adaptive, determined offline or updated on a timescale of minutes, and does not respond to the moment-to-moment variability in decoding difficulty that characterizes natural motor behavior (Shenoy et al., 2013).

2.2 Energy-Efficient Inference and Dynamic Computation

The broader machine learning community has developed a rich toolkit for energy-efficient inference, much of which is relevant to motor decoding. Pruning removes redundant weights or entire filters from trained networks, reducing both parameter count and computational cost (Han et al., 2015). Quantization reduces the numerical precision of weights and activations, enabling efficient integer arithmetic and reducing memory bandwidth requirements (Jacob et al., 2018). Knowledge distillation trains a compact student model to mimic the behavior of a larger teacher, transferring accuracy while reducing inference cost (Hinton et al., 2015). These techniques are largely static: once applied, the resulting model has a fixed computational graph and processes every input with the same set of operations (Han et al., 2022).

Dynamic computation represents a departure from this static paradigm (Han et al., 2022). Instead of committing to a fixed computational path for all inputs, dynamic methods adapt the amount or nature of computation to the input at hand. This adaptation can take several forms. Input-dependent routing selects among multiple specialized sub-networks or experts based on the input characteristics (Shazeer et al., 2017). Early exit, the focus of this paper, terminates inference at an intermediate layer when a confidence criterion is met (Teerapittayanon et al., 2016). Skip connections and layer dropping bypass certain layers or blocks for inputs that do not require them (Wang et al., 2019). The common thread is a recognition that not all inputs are equally difficult, and that uniform computation wastes resources on the easy majority (Laskaridis et al., 2021).

The theoretical foundation for dynamic computation rests on the observation that deep networks often produce correct predictions before the final layer, and that the additional computation performed by later layers, while beneficial for difficult inputs, is frequently unnecessary for simpler ones (Kaya et al., 2019). This phenomenon has been termed "overthinking" in the literature (Kaya et al., 2019). Empirical studies have shown that the distribution of input difficulty is typically skewed, with a large fraction of inputs being amenable to shallow processing and a smaller fraction requiring the full depth of the model (Matsubara et al., 2022). The energy savings achievable through early exit therefore depend critically on this distribution: the more skewed the difficulty distribution, the greater the

proportion of inputs that can exit early, and the larger the aggregate energy reduction (Laskaridis et al., 2021).

2.3 Early Exit Architectures and Applications

The early-exit paradigm was formalized by Teerapittayanon et al. (2016) through BranchyNet, which augmented a standard convolutional network with side branches containing intermediate classifiers and demonstrated that a large fraction of test images could be correctly classified at early exits. Subsequent work extended the approach to recurrent networks (Kaya et al., 2019), transformers (Xin et al., 2020; Zhou et al., 2020), and other architectures, and developed more sophisticated training procedures that jointly optimize the main network and the exit classifiers (Han et al., 2022). A comprehensive survey by Matsubara et al. (2022) categorizes early-exit methods along dimensions including exit placement, exit criteria, and training objectives, and discusses applications ranging from image classification to natural language processing.

The choice of exit criterion is central to the performance of an early-exit system (Laskaridis et al., 2021). The simplest criterion is a confidence threshold: if the maximum softmax probability at an intermediate exit exceeds a threshold, inference terminates (Teerapittayanon et al., 2016). More sophisticated criteria use entropy, margin between top classes, or learned gating functions that take multiple features of the intermediate representation into account (Wang et al., 2019; Xin et al., 2020). Recent work has shown that scalar confidence alone provides limited information about the true correctness of an early prediction, and that learned controllers operating on multidimensional detection states can substantially outperform confidence thresholds at high exit rates (Dai et al., 2020). The calibration of confidence estimates is also critical: neural networks are often overconfident, particularly at intermediate layers, and uncalibrated confidence can lead to premature exits that sacrifice accuracy without commensurate energy savings (Guo et al., 2017).

Hardware considerations for early exit have received increasing attention (Biggs et al., 2023). The benefits of early exit are not automatic on all hardware platforms: if the exit branch is executed in parallel with the main network, or if the memory access patterns of branching inference are inefficient, the theoretical energy savings may not materialize (Dimitriou et al., 2023). Work on FPGA and ASIC implementations has explored reconfigurable architectures that power down unused layers after an early exit (Putic et al., 2020), and software–hardware co-design approaches that predict exit points proactively to enable more aggressive voltage and frequency scaling (Korol et al., 2023). The energy savings reported in the literature vary widely depending on the hardware platform, the model architecture, and the input distribution, underscoring the importance of empirical evaluation on target hardware rather than extrapolation from theoretical estimates (Matsubara et al., 2024).

2.4 Early Exit in Neural Signal Processing

The application of early exit to neural signal processing, and to EEG in particular, is a nascent but growing area. A recent study integrated early-exit mechanisms into a lightweight architecture for EEG-based emotion recognition, achieving a 2.5-fold improvement in inference speed and a 1.8-fold reduction in parameters while improving classification accuracy

(Thuong et al., 2026). This work established that early exit is compatible with the noisy, variable nature of EEG signals, although it focused on emotion recognition rather than motor decoding and did not examine the energy–accuracy trade-off in detail. The entropy-based exit criteria used in that study are a natural fit for EEG, where the intermediate representations at different layers capture different frequency bands and spatial patterns (Lawhern et al., 2018).

Motor decoding presents challenges that differ from emotion recognition. The temporal structure of motor-related neural activity is more pronounced, with movement-related cortical potentials, event-related desynchronization, and other phenomena that evolve over hundreds of milliseconds (Pfurtscheller and Lopes da Silva, 1999). A decoder that processes windows of neural data can exploit this structure to anticipate when the current window is likely to be easy or difficult (Churchland et al., 2012). Furthermore, motor decoding often involves regression to continuous kinematics rather than classification, requiring adaptations of the confidence estimation machinery that has been developed primarily for classification tasks (Inoue et al., 2017). The regression setting poses unique challenges: the notion of a "confident" prediction is less straightforward for continuous outputs, and the consequences of an erroneous early exit may be graded rather than catastrophic (Sussillo et al., 2015).

The literature on energy-efficient motor decoding has largely treated computation as a fixed cost to be minimized through model compression, without considering the possibility of input-adaptive computation (Wu et al., 2018; Yin et al., 2021). Conversely, the early-exit literature has focused on vision and language applications, with limited attention to the distinctive characteristics of neural time series (Matsubara et al., 2022). This paper bridges these two bodies of work by systematically evaluating early exit as an energy strategy for motor decoding, characterizing the accuracy–energy trade-off across model families and task types, and identifying the design considerations that arise when dynamic computation is applied to neural signals on edge hardware.

3. Methods

3.1 Overview of Experimental Design

Our investigation is structured around three model architectures, two decoding tasks, and a set of exit criteria that span a range of complexity. For each architecture–task combination, we train a baseline model without early exits to establish reference accuracy and energy consumption. We then introduce intermediate exit points at multiple depths, train the augmented models with a multi-exit loss (Teerapittayanon et al., 2016), and evaluate the accuracy–energy trade-off across exit thresholds and criteria. All energy measurements are performed on a representative edge hardware platform, and we report both per-inference energy and aggregate energy over a test set to capture the effect of input-dependent computation (Dimitriou et al., 2023).

3.2 Datasets and Tasks

We evaluate on two publicly available neural datasets that represent distinct motor decoding paradigms.

The first dataset is a motor imagery EEG dataset comprising recordings from multiple subjects performing imagined left- and right-hand movements (Tangermann et al., 2012). EEG was recorded from 22 channels at 250 Hz using a standard 10–20 montage. The decoding task is binary classification of the imagined movement class from 2-second windows of EEG. This dataset is representative of non-invasive BCI applications where the decoder must operate on noisy, low-amplitude signals and where inter-subject variability is substantial (Lotte et al., 2018).

The second dataset is an intracortical recording from a non-human primate performing a center-out reaching task (Shenoy et al., 2013). Neural activity was recorded from a 96-channel microelectrode array in motor cortex, and spikes were binned into 20 ms windows. The decoding task is regression of the two-dimensional hand velocity from 10 consecutive time bins (200 ms of history). This dataset is representative of invasive BCI applications where the decoder must produce continuous, low-latency control signals from high-dimensional neural activity (Sussillo et al., 2015).

For both datasets, we partition the data into training, validation, and test sets using a subject-wise or session-wise split to prevent information leakage (Schirrmeister et al., 2017). Hyperparameters, including exit thresholds where applicable, are selected on the validation set and evaluated on the held-out test set.

3.3 Model Architectures

We select three model families that span the design space of motor decoding architectures.

The MLP model consists of three fully connected layers with 256, 128, and 64 units, respectively, with ReLU activations and dropout applied between layers (Schirrmeister et al., 2017). For the classification task, the output layer is a softmax over two classes; for the regression task, the output layer is a linear projection to two velocity components. MLPs remain common in real-time BCI systems due to their low computational overhead and predictable memory access patterns (Lotte et al., 2018).

The CNN model follows an EEGNet-inspired architecture with temporal convolution, depthwise spatial convolution, and separable convolution layers (Lawhern et al., 2018). The temporal convolution captures frequency-specific patterns, the spatial convolution learns electrode weightings, and the separable convolution combines these features. For the classification task, global average pooling is followed by a softmax layer; for the regression task, the pooling layer is followed by a fully connected layer producing the velocity output. CNNs are the dominant architecture for EEG-based decoding and offer a favorable balance of expressiveness and efficiency (Schirrmeister et al., 2017).

The recurrent model is a two-layer gated recurrent unit (GRU) network with 128 hidden units per layer (Cho et al., 2014). The input at each time step is the neural feature vector (EEG channel amplitudes or spike counts), and the output is either a class probability or a velocity prediction. The GRU captures temporal dependencies that are not explicitly modeled by the

feedforward architectures and is representative of sequence models used in intracortical decoding (Sussillo et al., 2015).

3.4 Early Exit Integration

For each architecture, we introduce intermediate exit points at approximately one-third and two-thirds of the network depth (Teerapittayanon et al., 2016). Each exit point consists of a lightweight classification or regression head attached to the intermediate representation. For the MLP, the first exit follows the first hidden layer and the second exit follows the second hidden layer. For the CNN, exits are placed after the depthwise spatial convolution and after the separable convolution. For the GRU, exits are placed after the first GRU layer and after the second GRU layer. The exit heads are designed to add minimal computational overhead: for classification, a single fully connected layer with a softmax output; for regression, a single fully connected layer with a linear output (Laskaridis et al., 2021).

Training proceeds in two phases. In the first phase, the baseline model is trained to convergence using the standard task loss. In the second phase, the exit heads are initialized and the entire augmented model is fine-tuned with a weighted sum of the losses at each exit and the final output (Teerapittayanon et al., 2016). The loss weights are selected to balance the contribution of each exit and to prevent the early exits from degrading the performance of the full model. For the classification task, we use cross-entropy loss; for the regression task, we use mean squared error (Kaya et al., 2019).

3.5 Exit Criteria

We evaluate four exit criteria, ranging from simple to sophisticated.

The confidence threshold criterion terminates inference at an exit if the maximum softmax probability (for classification) or a normalized confidence score (for regression) exceeds a threshold (Teerapittayanon et al., 2016). This is the standard baseline in the early-exit literature.

The entropy criterion computes the Shannon entropy of the softmax distribution at each exit and terminates if the entropy falls below a threshold (Wang et al., 2019). Entropy captures the spread of the predicted distribution and is often a more sensitive indicator of prediction reliability than the maximum probability alone.

The margin criterion computes the difference between the top-2 class probabilities and terminates if the margin exceeds a threshold (Wang et al., 2019). This criterion focuses on the discriminability of the top classes and can be more robust when the softmax distribution is broad but the top classes are well-separated.

The energy-aware criterion incorporates the estimated cost of continued computation into the exit decision (Laskaridis et al., 2021). At each exit, we compute a score that combines the confidence measure with a penalty term proportional to the expected remaining computation. The exit is triggered when the confidence score exceeds a threshold that is adjusted upward for exits that are followed by more remaining computation, reflecting the fact that the energy

savings from exiting at an early layer are greater than those from exiting at a later layer. This criterion is designed to optimize the energy–accuracy trade-off rather than accuracy alone.

For the regression task, where softmax-based confidence measures are not directly applicable, we use the variance of the predictive distribution at each exit (estimated through Monte Carlo dropout; Gal and Ghahramani, 2016) as the confidence measure. Lower variance indicates a more confident prediction, and the exit criteria are adapted accordingly.

3.6 Energy Measurement

Energy measurements are performed on an edge hardware platform representative of wearable and embedded BCI systems. The platform comprises a low-power ARM Cortex-M7 microcontroller with 512 KB of RAM and a clock frequency of 480 MHz. The neural network inference is implemented in C using the CMSIS-NN library, which provides optimized kernels for ARM Cortex-M processors (Lai et al., 2018). Energy is measured using a precision power monitor that samples the current drawn by the microcontroller at 10 kHz, and the energy per inference is computed by integrating the power over the inference duration (Dimitriou et al., 2023).

To isolate the energy attributable to the neural network inference from other system components, we measure the baseline power consumption of the microcontroller in an idle state and subtract it from the total. We report both the mean energy per inference across the test set and the aggregate energy for the entire test set, the latter capturing the effect of input-dependent variation in computation (Matsubara et al., 2024).

For each model and exit criterion, we sweep the threshold across a range of values and record the accuracy and energy at each operating point. This produces an accuracy–energy Pareto frontier that characterizes the achievable trade-offs for that model–criterion combination (Laskaridis et al., 2021).

3.7 Evaluation Metrics

The primary metrics are decoding accuracy (for classification) or mean squared error and correlation (for regression), and energy per inference (in millijoules). We also report the exit rate, defined as the fraction of test windows that terminate at an intermediate exit rather than the final layer, and the distribution of exit depths (Teerapittayanon et al., 2016).

To characterize the energy–accuracy trade-off, we compute the energy savings relative to the baseline at matched accuracy levels where possible, and the accuracy loss at specified energy savings levels. We also compute the area under the accuracy–energy Pareto curve as a summary measure of the achievable trade-off frontier, with larger areas indicating better trade-offs (Matsubara et al., 2022).

Statistical significance is assessed using bootstrap confidence intervals with 1000 resamples (Efron and Tibshirani, 1993), and differences between criteria are evaluated using paired tests across test windows.

4. Results

4.1 Baseline Performance and Energy Consumption

We first establish the baseline performance of the three architectures without early exits. On the motor imagery classification task, the CNN achieves the highest accuracy (72.4%), followed by the MLP (68.1%) and the GRU (66.9%). The ranking reflects the CNN's ability to learn spatial filters that capture the topography of motor imagery-related activity (Lawhern et al., 2018). On the intracortical regression task, the GRU achieves the highest correlation with hand velocity (0.81), followed by the MLP (0.76) and the CNN (0.73). The GRU's advantage on this task is consistent with the temporal structure of intracortical activity during reaching, which involves sequential activation of motor cortical populations (Churchland et al., 2012).

The baseline energy consumption per inference varies substantially across architectures and tasks. For the classification task, the CNN consumes the most energy (2.84 mJ per inference), followed by the GRU (2.17 mJ) and the MLP (0.92 mJ). For the regression task, the GRU consumes the most energy (3.41 mJ), followed by the CNN (2.88 mJ) and the MLP (1.04 mJ). These differences reflect both the computational complexity of the architectures and the input dimensionality, which is higher for the intracortical dataset (96 channels versus 22 for EEG) (Wu et al., 2018).

4.2 Accuracy–Energy Trade-offs with Confidence Thresholding

The introduction of early exits with confidence thresholding produces substantial energy savings at modest accuracy costs across all architectures and tasks (Teerapittayanon et al., 2016). On the motor imagery classification task, the CNN achieves a 31% reduction in aggregate energy at a threshold that preserves accuracy within 1 percentage point of the baseline (71.5% versus 72.4%). The exit rate at this operating point is 42%, meaning that nearly half of the test windows terminate before the final layer. The MLP and GRU exhibit similar patterns, with energy savings of 28% and 24% respectively at matched accuracy thresholds, and exit rates of 38% and 33%.

On the intracortical regression task, the energy savings are somewhat smaller but still substantial. The GRU achieves a 26% reduction in aggregate energy at a threshold that preserves the correlation with hand velocity within 0.02 of the baseline (0.79 versus 0.81). The exit rate is 35%. The MLP and CNN achieve savings of 22% and 19% respectively. The smaller savings on the regression task reflect the fact that the distribution of decoding difficulty is less skewed for continuous kinematics than for binary classification: fewer windows are "easy" enough to be confidently decoded by the early exits (Laskaridis et al., 2021).

The exit depth distribution reveals that the first exit accounts for the majority of early terminations. For the CNN on the classification task, 68% of exiting windows terminate at the first exit and 32% at the second exit. This pattern is consistent across architectures and tasks and indicates that the largest energy savings come from the earliest exit opportunities (Matsubara et al., 2022). The windows that reach the second exit are those that are moderately difficult, while the windows that proceed to the final layer are the most challenging.

4.3 Comparison of Exit Criteria

The comparison of exit criteria reveals that entropy and margin criteria outperform simple confidence thresholding at high energy savings levels (Wang et al., 2019). At a 30% energy reduction target, the entropy criterion achieves 1.2 percentage points higher accuracy than the confidence threshold on the CNN classification task (70.9% versus 69.7%), and the margin criterion achieves a 1.5 percentage point advantage (71.2% versus 69.7%). At lower energy savings levels (10–15%), the differences between criteria are small and not statistically significant.

The energy-aware criterion provides the best overall trade-off when evaluated across the full Pareto frontier (Laskaridis et al., 2021). By adjusting the exit threshold to account for the remaining computation, this criterion shifts the operating point toward earlier exits when the energy penalty for continuing is high, and toward later exits when the accuracy benefit of continued computation is likely to be substantial. On the CNN classification task, the energy-aware criterion achieves an area under the Pareto curve that is 8.4% larger than the confidence threshold, 4.1% larger than entropy, and 3.7% larger than margin. The advantage is most pronounced at intermediate energy savings levels (20–35%), where the choice of exit criterion has the largest effect on the accuracy–energy trade-off.

The learned controller approach, which uses a multidimensional state to predict the correctness of early exits (Dai et al., 2020), was not implemented in our experiments due to the additional training complexity and the focus on criteria that can be implemented with minimal overhead on edge hardware. However, the results from the confidence-based criteria suggest that the multidimensional approach could yield further improvements, particularly at high exit rates where scalar confidence is least informative (Dai et al., 2020).

4.4 Effect of Input Difficulty Distribution

The energy savings from early exit are strongly dependent on the distribution of input difficulty (Laskaridis et al., 2021). We quantify this relationship by computing, for each test window, the minimum exit depth at which the prediction is correct, and we analyze the distribution of this quantity across the test set. For the motor imagery classification task, the distribution is highly skewed: 38% of windows are correctly classified at the first exit, 22% at the second exit, and 40% require the final layer. For the intracortical regression task, the distribution is less skewed: 24% of windows achieve their best prediction at the first exit, 19% at the second exit, and 57% require the final layer.

The skewness of the difficulty distribution directly predicts the achievable energy savings (Matsubara et al., 2022). A simple model that assumes each window exits at the depth at which it would be correctly classified (an oracle that knows the correct answer) yields an upper bound on savings. For the classification task, the oracle savings are 52% of baseline energy; the achieved savings of 31% at matched accuracy represent 60% of this upper bound. For the regression task, the oracle savings are 38%; the achieved savings of 26% represent 68% of the upper bound. The gap between oracle and achieved savings reflects the imperfection of the confidence measures: some windows that could be correctly classified early are not confident

enough to trigger an exit, and some windows that are incorrectly classified early are confident enough to exit prematurely (Guo et al., 2017).

We also examine the temporal structure of decoding difficulty. For the motor imagery dataset, difficulty exhibits weak but statistically significant temporal correlation: the autocorrelation of the exit depth at lag 1 is 0.31, indicating that consecutive windows tend to have similar difficulty (Pfurtscheller and Lopes da Silva, 1999). This correlation suggests that the exit behavior of previous windows could inform the exit policy for the current window, a possibility we discuss further in Section 5. For the intracortical dataset, the temporal correlation is stronger (autocorrelation of 0.47 at lag 1), reflecting the smooth evolution of neural activity during reaching (Churchland et al., 2012).

4.5 Calibration Analysis

The calibration of confidence estimates at intermediate exits has a substantial impact on the energy–accuracy trade-off (Guo et al., 2017). We measure calibration using the expected calibration error (ECE) and reliability diagrams. At the first exit of the CNN classification model, the ECE is 0.087, indicating significant overconfidence: the model's predicted probabilities are systematically higher than the observed accuracy. At the second exit, the ECE decreases to 0.052, and at the final output, it is 0.031. This pattern—increasing calibration with depth—is consistent with the literature on early-exit networks and reflects the fact that intermediate representations are less informative and the exit heads are trained with fewer effective examples (Kaya et al., 2019).

The overconfidence at early exits leads to suboptimal exit decisions: windows that are predicted with high confidence at the first exit are more likely to be incorrectly classified than the confidence value suggests, and the simple thresholding criterion exits on these windows, sacrificing accuracy (Guo et al., 2017). Temperature scaling, a post-hoc calibration method that divides the logits by a learned temperature parameter (Guo et al., 2017), reduces the ECE at the first exit from 0.087 to 0.041 and improves the accuracy at matched energy savings by 1.8 percentage points on the CNN classification task. The improvement is smaller but still positive for the MLP and GRU models. For the regression task, calibration of the predictive variance is more challenging, and we did not observe a consistent benefit from post-hoc calibration methods in our experiments (Gal and Ghahramani, 2016).

4.6 Sensitivity to Exit Placement

The placement of exit points affects both the achievable energy savings and the accuracy of early predictions (Teerapittayanon et al., 2016). We compare the two-exit configuration described above with a three-exit configuration in which an additional exit is placed immediately after the first layer of each architecture, and with a single-exit configuration in which only the middle exit is retained. The three-exit configuration achieves slightly higher energy savings at matched accuracy (an additional 3–5 percentage points of savings) but incurs a larger accuracy penalty at aggressive exit thresholds, as the very early exit is often unreliable (Laskaridis et al., 2021). The single-exit configuration achieves lower savings (a reduction of 8–12 percentage points) but is simpler to implement and imposes less training overhead. The

optimal configuration depends on the application's tolerance for accuracy loss and the available training resources.

4.7 Energy Breakdown and Hardware Considerations

The measured energy savings translate directly into extended battery life for edge deployments (Dimitriou et al., 2023). For the CNN classification model operating at a 30% energy reduction, the energy per inference decreases from 2.84 mJ to 1.99 mJ. For a device with a 100 mAh battery operating at 3.3 V and performing one inference per second, this reduction extends the operating time from approximately 32 hours to 46 hours, assuming the decoder is the dominant energy consumer. In practice, other components such as the analog front-end and wireless radio may consume comparable or greater energy, so the system-level savings will be smaller but still meaningful (Wu et al., 2018).

The energy savings from early exit are not uniformly distributed across the inference pipeline. The exit heads themselves consume energy, and the branch instructions and control flow logic add overhead that partially offsets the savings from skipped layers (Biggs et al., 2023). We estimate that the exit heads account for 8–12% of the energy consumed at the exits where they are evaluated, and the control overhead accounts for an additional 3–5%. These overheads are included in all reported energy measurements. On hardware platforms with more sophisticated power management capabilities, such as the ability to gate power to unused layers (Putic et al., 2020), the savings could be larger, but such features are not available on the microcontroller platform used in this study.

5. Discussion of Findings

5.1 Interpretation of Results

The results establish that early exit is an effective energy strategy for motor decoding on edge hardware, achieving substantial reductions in inference energy at modest accuracy costs across model architectures and task types (Teerapittayanon et al., 2016; Laskaridis et al., 2021). The magnitude of the savings—26–31% at matched accuracy in our experiments—is meaningful in the context of wearable and implantable BCI systems, where every millijoule of saved energy translates into extended operating time, reduced thermal burden, and improved user comfort (He et al., 2018). The finding that a large fraction of neural windows can be confidently decoded by shallower sub-networks aligns with the broader observation that many inputs are "easy" in the early-exit literature (Matsubara et al., 2022), but the specific rates and patterns we observe are shaped by the distinctive characteristics of neural signals (Shenoy et al., 2013).

The comparison of exit criteria reveals that the choice of criterion matters most at intermediate to high energy savings levels, where the trade-off between exiting early and preserving accuracy is most acute (Wang et al., 2019). The energy-aware criterion, which explicitly accounts for the cost of continued computation, outperforms accuracy-only criteria across the Pareto frontier (Laskaridis et al., 2021). This finding has a straightforward interpretation: the decision to exit is fundamentally an economic one, trading the expected accuracy benefit of additional computation against its energy cost, and criteria that ignore the cost side of this trade-

off will make suboptimal decisions. The improvement from energy-aware exit criteria is modest in absolute terms but comes at negligible computational overhead, making it an attractive default for practical deployments.

The calibration analysis reveals a challenge that is specific to the early-exit setting: intermediate exits are systematically overconfident, and this overconfidence leads to premature exits that sacrifice accuracy (Guo et al., 2017). The overconfidence is not surprising—it is a well-documented phenomenon in deep networks—but its consequences are more severe in the early-exit context because the exit decision directly determines whether the model's prediction is used (Kaya et al., 2019). Temperature scaling provides a simple and effective remedy, improving the accuracy–energy trade-off by nearly two percentage points at matched energy on the CNN classification task (Guo et al., 2017). The fact that calibration had a smaller effect on the regression task reflects the different nature of uncertainty in continuous prediction, where the predictive variance is a less direct measure of correctness than the softmax distribution (Gal and Ghahramani, 2016).

5.2 Implications for Motor Decoding System Design

The findings have several implications for the design of practical early-exit motor decoders. First, the exit placement should prioritize early exits, as the majority of energy savings come from the first exit opportunity (Teerapittayanon et al., 2016). A single well-placed early exit may capture most of the achievable savings with less training complexity and control overhead than multiple exits (Laskaridis et al., 2021). The marginal benefit of additional exits is positive but diminishing, and the optimal number depends on the skewness of the difficulty distribution and the tolerance for accuracy loss.

Second, the exit criterion should be calibrated and, ideally, energy-aware (Guo et al., 2017; Laskaridis et al., 2021). The calibration can be performed post hoc using a held-out validation set, adding minimal complexity to the training pipeline. The energy-aware adjustment requires knowledge of the relative costs of computation at different depths, which can be estimated offline and incorporated into the threshold selection. Neither of these refinements imposes significant runtime overhead.

Third, the temporal structure of neural signals offers an opportunity that has not been fully exploited in our experiments or in the existing literature (Churchland et al., 2012). The observation that decoding difficulty is temporally correlated suggests that the exit decisions of previous windows carry information about the current window. A sequential exit policy that conditions the threshold on recent exit behavior could potentially improve the energy–accuracy trade-off, although it would also introduce temporal dependencies that complicate the analysis of error propagation. This is a promising direction for future work.

5.3 Limitations

Several limitations of this study should be acknowledged. The experiments were conducted on two datasets, and while these datasets span the invasive–non-invasive and classification–regression dimensions, they do not represent the full diversity of motor decoding applications

(Lotte et al., 2018). The generalizability of the quantitative findings to other datasets, subjects, and recording modalities remains to be established. The edge hardware platform used for energy measurements, while representative of wearable BCI systems, is not the only relevant platform; different microcontrollers, FPGAs, or neuromorphic chips may exhibit different energy characteristics for branching inference (Dimitriou et al., 2023).

The training procedure for the early-exit models involved a fine-tuning phase that added computational overhead, and we did not systematically explore the trade-off between training cost and inference savings (Kaya et al., 2019). In applications where models are retrained frequently (e.g., to adapt to session-to-session variability), the training overhead could be a significant consideration. The regression task posed particular challenges for confidence estimation, and our approach using predictive variance, while functional, is not the only or necessarily the best method (Gal and Ghahramani, 2016). Alternative approaches, such as ensembles of exit heads or Bayesian neural network methods, might yield better calibration and improved trade-offs.

The study did not address the interaction between early exit and other energy-saving techniques such as quantization, pruning, or electrode selection (Han et al., 2015; Jacob et al., 2018; Lotte et al., 2018). In a practical system, these techniques would be combined, and the combined effect might be less than the sum of the individual effects due to overlapping mechanisms (Han et al., 2022). For example, a pruned model may have less redundant computation to skip through early exit, potentially reducing the relative benefit of dynamic computation. This interaction is an important consideration for system-level optimization and warrants further investigation.

5.4 Open Challenges and Future Directions

Several open challenges emerge from this work. The most pressing is the need for robust performance under distribution shift (Shenoy et al., 2013). Motor decoding models are known to degrade when applied to new sessions or subjects, and the confidence estimates that drive exit decisions may become miscalibrated under such shifts (Guo et al., 2017). An exit policy that is well-calibrated on the training distribution may exit prematurely or conservatively on a shifted distribution, leading to either accuracy loss or lost energy savings. Methods for online calibration and adaptation of exit thresholds are needed to maintain performance in deployment.

A second challenge concerns the temporal dynamics of exit decisions. In a closed-loop BCI, the decoder operates continuously, and the exit decisions at successive time steps are not independent (Churchland et al., 2012). A premature exit that produces an incorrect control signal may lead to corrective movements or error-related neural activity that affects subsequent decoding. The consequences of early-exit errors therefore propagate through the closed-loop system in ways that are not captured by our open-loop evaluation. Simulated closed-loop experiments or human-in-the-loop studies are needed to characterize these effects and to design exit policies that are robust to them.

A third challenge is hardware support for branching inference (Putic et al., 2020). The energy savings we measured on a general-purpose microcontroller reflect the computational savings from skipped layers but do not exploit the full potential of dynamic computation. Hardware that can power down unused layers after an early exit, or that can dynamically adjust voltage and frequency in response to the predicted computational load, could achieve larger savings (Biggs et al., 2023). The development of such hardware, and the software stacks to program it, is an active area of research but has not yet been applied to motor decoding (Dimitriou et al., 2023).

A fourth direction concerns the integration of early exit with the broader system architecture. In a wireless BCI, the decoder is one component of a pipeline that includes signal acquisition, preprocessing, and transmission (He et al., 2018). The energy savings from early exit could be leveraged to reduce transmission bandwidth (by transmitting only the decoded output rather than raw signals), or to increase the duty cycle of acquisition (by spending the saved energy on more frequent sampling). These system-level trade-offs have not been explored and could yield larger overall savings than decoder-level optimization alone.

Finally, the application of early exit to more sophisticated decoding models, such as transformers and large-scale foundation models that are beginning to be applied to neural data, presents both challenges and opportunities (Xin et al., 2020; Zhou et al., 2020). Transformers have been the primary target of early-exit research in natural language processing, and the techniques developed there could be adapted to neural decoding. However, the computational characteristics of transformers differ from the convolutional and recurrent models studied here, and the energy–accuracy trade-offs may differ accordingly (Matsubara et al., 2024).

6. Conclusion

This paper investigated early exit as an energy strategy for motor decoding on edge hardware. Through a systematic evaluation spanning three model architectures, two decoding tasks, and four exit criteria, we established that early exit can reduce inference energy by 26–31% at matched accuracy, with the largest savings coming from the earliest exit opportunities (Teerapittayanon et al., 2016; Laskaridis et al., 2021). The achievable savings depend on the skewness of the input difficulty distribution, which is more favorable for classification than for regression and varies across recording modalities (Matsubara et al., 2022). Energy-aware exit criteria that account for the cost of continued computation outperform accuracy-only criteria (Laskaridis et al., 2021), and calibration of intermediate confidence estimates is essential for realizing the full benefit of dynamic computation (Guo et al., 2017).

These findings position early exit as a practical and model-agnostic lever for energy-efficient motor decoding. Unlike techniques that require architectural changes or retraining from scratch, early exit can be retrofitted to existing models with a modest fine-tuning phase (Kaya et al., 2019). Unlike static compression methods, it adapts computation to the input at hand, exploiting the substantial variability in decoding difficulty that characterizes neural signals (Shenoy et al., 2013). The energy savings, while not sufficient on their own to enable indefinite operation on battery power, are a meaningful contribution to the system-level energy budget

and can be combined with other optimization techniques to extend the operating time of wearable and implantable BCI systems (He et al., 2018; Wu et al., 2018).

The broader significance of this work lies in its demonstration that the principles of dynamic computation, developed primarily in the context of computer vision and natural language processing, are applicable to neural signal processing (Han et al., 2022). Motor decoding is a domain where the constraints are particularly severe—low-power hardware, real-time latency requirements, and the need for reliable performance in the face of noisy and non-stationary signals—and where the benefits of input-adaptive computation are therefore particularly valuable (Lebedev and Nicolelis, 2017). As BCI systems move from laboratory demonstrations toward clinical and consumer deployment, techniques that reduce energy consumption without sacrificing reliability will become increasingly important. Early exit, and dynamic computation more broadly, offers a principled approach to achieving this goal.

Future work should address the open challenges identified in this paper: the robustness of exit policies under distribution shift (Shenoy et al., 2013), the temporal dynamics of exit decisions in closed-loop operation (Churchland et al., 2012), the development of hardware that can fully exploit the savings from branching inference (Putic et al., 2020; Biggs et al., 2023), and the extension of early-exit techniques to the next generation of neural decoding models (Matsubara et al., 2024). The integration of early exit with other energy-saving techniques, and the evaluation of the combined system on realistic deployment scenarios, will be essential for translating the findings of this study into practical impact.

REFERENCES

1. Biggs, B., Bouganis, C.-S., and Constantinides, G. (2023). ATHEENA: A toolflow for hardware early-exit network automation. In *Proceedings of the IEEE 31st Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 121–132.

2. Blankertz, B., Lemm, S., Treder, M., Haufe, S., and Müller, K.-R. (2008). Single-trial analysis and classification of ERP components—a tutorial. *NeuroImage*, 56(2), 814–825.

3. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

4. Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., and Shenoy, K. V. (2012). Neural population dynamics during reaching. *Nature*, 487(7405), 51–56.

5. Dai, X., Kong, X., and Guo, T. (2020). EPNet: Learning to exit with flexible multi-branch network. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, 235–244.

6. Dimitriou, A., Hu, M., Hare, J., and Merrett, G. V. (2023). Exploration of decision sub-network architectures for FPGA-based dynamic DNNs. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*.

7. Efron, B., and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman and Hall/CRC.

8. Gal, Y., and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 1050–1059.

9. Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 1321–1330.

10. Han, S., Pool, J., Tran, J., and Dally, W. J. (2015). Learning both weights and connections for efficient neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 1135–1143.

11. Han, Y., Huang, G., Song, S., Yang, L., Wang, H., and Wang, Y. (2022). Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11), 7436–7456.

12. He, B., Yuan, H., Meng, J., and Gao, S. (2018). Brain–computer interfaces. In *Neural Engineering*, 4th ed., Springer, 87–120.

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

14. Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., Haddadin, S., Liu, J., Cash, S. S., van der Smagt, P., and Donoghue, J. P. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398), 372–375.

15. Inoue, Y., Hirose, M., and Shinoda, K. (2017). Adaptive computation of deep neural networks for efficient inference. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2632–2636.

16. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2704–2713.

17. Kaya, Y., Hong, S., and Dumitras, T. (2019). Shallow-deep networks: Understanding and mitigating network overthinking. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 3301–3310.

18. Korol, G., Jordan, M. G., Rutzig, M. B., Castrillon, J., and Beck, A. C. S. (2023). Pruning and early-exit co-optimization for CNN acceleration on FPGAs. In *Proceedings of the Design, Automation and Test in Europe Conference (DATE)*.

19. Lai, L., Suda, N., and Chandra, V. (2018). CMSIS-NN: Efficient neural network kernels for Arm Cortex-M CPUs. *arXiv preprint arXiv:1801.06601*.

20. Laskaridis, S., Kouris, A., and Lane, N. D. (2021). Adaptive inference through early-exit networks: Design, challenges and directions. In *Proceedings of the 5th International Workshop on Embedded and Mobile Deep Learning (EMDL)*, 1–6.

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

22. Lebedev, M. A., and Nicolelis, M. A. L. (2017). Brain–machine interfaces: From basic science to neuroprostheses and neurorehabilitation. *Physiological Reviews*, 97(2), 767–837.

23. Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., and Yger, F. (2018). A review of classification algorithms for EEG-based brain–computer interfaces: A 10 year update. *Journal of Neural Engineering*, 15(3), 031005.

24. Matsubara, Y., Levorato, M., and Restuccia, F. (2022). Split computing and early exiting for deep learning applications: Survey and research challenges. *ACM Computing Surveys*, 55(5), 1–30.

25. Matsubara, Y., Levorato, M., Restuccia, F., and Sathi, V. (2024). Early-exit deep neural network: A comprehensive survey. *ACM Computing Surveys*.

26. Pfurtscheller, G., and Lopes da Silva, F. H. (1999). Event-related EEG/MEG synchronization and desynchronization: Basic principles. *Clinical Neurophysiology*, 110(11), 1842–1857.

27. Pfurtscheller, G., and Neuper, C. (2001). Motor imagery and direct brain–computer communication. *Proceedings of the IEEE*, 89(7), 1123–1134.

28. Putic, M., Buyuktosunoglu, A., Venkataramani, S., Bose, P., Eldridge, S., and Stan, M. (2020). Dyhard-DNN: Even more DNN acceleration with dynamic hardware reconfiguration. In *Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC)*.

29. Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., and Ball, T. (2017). Deep

learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, 38(11), 5391–5420.

30. Suppiah, R., Danthakanni, R., Nair, M., & Kafumbe, S. (2026, July). Novel Hierarchical Edge AI Architecture for Resource-Constrained Embedded Platforms: A Comprehensive Framework for Distributed Intelligence. In *2026 6th International Conference on Electrical, Computer and Energy Technologies (ICECET)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICECET65726.2026.11632851

31. Suppiah, R., & Kim, N. (2026, July). Neural Cognitive Fingerprint Network (NeuroPrint-Net). In *2026 6th International Conference on Electrical, Computer and Energy Technologies (ICECET)* (pp. 1-7). IEEE. https://doi.org/10.1109/ICECET65726.2026.11632430

32. Suppiah, R., & Kim, N. (2026). Single-channel gamma burst dynamics and adaptive deep learning for robust EEG motor state decoding. *Biomedical Engineering Letters*, 1-14. https://doi.org/10.1007/s13534-026-00577-w

33. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*.

34. Shenoy, K. V., Sahani, M., and Churchland, M. M. (2013). Cortical control of arm movements: A dynamical systems perspective. *Annual Review of Neuroscience*, 36, 337–359.

35. Sussillo, D., Churchland, M. M., Kaufman, M. T., and Shenoy, K. V. (2015). A neural network that finds a naturalistic solution for the production of muscle activity. *Nature Neuroscience*, 18(7), 1025–1033.

36. Tangermann, M., Müller, K.-R., Aertsen, A., Birbaumer, N., Braun, C., Brunner, C., Leeb, R., Mehring, C., Miller, K. J., Müller-Putz, G., Nolte, G., Pfurtscheller, G., Preissl, H., Schalk, G., Schlögl, A., Vidaurre, C., Waldert, S., and Blankertz, B. (2012). Review of the BCI competition IV. *Frontiers in Neuroscience*, 6, 55.

37. Teerapittayanon, S., McDanel, B., and Kung, H. T. (2016). BranchyNet: Fast inference via early exiting from deep neural networks. In *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR)*, 2464–2469.

38. Thuong, D. T. M., Huy, N. P., Phung, T.-N., Cuong, D. N., Trung, N. P., and Nguyen, T. A. (2026). Enhancing the efficiency of lightweight deep neural networks in EEG-based emotion recognition using early-exit techniques. In *Proceedings of the International Conference on Artificial Intelligence and Big Data* (Springer).

39. Wang, M., Mo, J., Lin, J., Wang, Z., and Du, L. (2019). DynExit: A dynamic early-exit strategy for deep residual networks. In *Proceedings of the IEEE International Workshop on Signal Processing Systems (SiPS)*, 178–183.

40. Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain–computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6), 767–791.

41. Wu, T., Li, Y., and Yang, Z. (2018). Efficient FPGA acceleration of convolutional neural networks for motor imagery classification. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 1234–1239.

42. Xin, J., Tang, R., Lee, J., Yu, Y., and Lin, J. (2020). DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2246–2251.

43. Yin, Y., Li, Y., and Zhang, X. (2021). An energy efficient convolutional spiking neural network for EEG analytics towards wearable BCI applications. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

44. Zhou, W., Xu, C., Ge, T., McAuley, J., Xu, K., and Wei, F. (2020). BERT loses patience: Fast and robust inference with early exit. In *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 1–12.