Decoding Pathway Selection for Resource-Constrained Brain-Computer Interfaces: A Systematic Trade-off between Conventional Feature Engineering and Lightweight Deep Learning

Jiayi Zheng
Maynooth International Engineering College, Fuzhou University, No. 2, Xueyuan Road, Minhou County, Fuzhou City, Fujian Province, 350108, China
jiayi.zheng.fzu@outlook.com

ABSTRACT

The evolution of Brain-Computer Interfaces (BCI) towards portable, wearable, and edge computing scenarios presents stringent challenges for decoding algorithms, demanding real-time processing under constraints of limited computational power, memory, and energy consumption. This paper provides a systematic review and comparative analysis of two core technical pathways to address this challenge: conventional feature engineering based on explicit signal processing and data-driven lightweight deep learning. Conventional methods (e.g., Common Spatial Pattern (CSP), Canonical Correlation Analysis (CCA)) rely on expert knowledge, maintaining dominance in extremely resource-constrained scenarios due to their superior computational efficiency, low power consumption, and high interpretability. Lightweight deep learning methods (e.g., EEGNet, a compact convolutional neural network designed for electroencephalogram (EEG) decoding), through end-to-end learning, demonstrate advantages in decoding accuracy, especially for handling complex tasks, though their model compression and deployment processes are more complex. This paper constructs a multi-dimensional trade-off analysis framework, conducting an in-depth comparison of the two pathways from aspects such as accuracy, computational efficiency, energy consumption, data requirements, interpretability, and deployment complexity. The analysis indicates that the selection of a solution highly depends on the priority constraints of the target application: scenarios with extreme demands for power consumption and determinism (e.g., implantable devices) lean towards traditional methods; whereas for complex tasks (e.g., cognitive state monitoring) that require higher accuracy and have relatively sufficient resources, lightweight deep learning is a better choice. Finally, this paper focuses on the future. He talked about such things as edge adaptation of EEG basic model, hybrid architecture of brain insertion and collaborative design of hardware and software. The purpose is to provide a clear framework for selecting decoders in different BCI application scenarios on the edge.

Keywords: Brain-Computer Interface; Edge Computing; Computational Efficiency; Lightweight Deep Learning; Feature Engineering; Model Deployment; Resource Constraints.

1. INTRODUCTION

The practical application of Brain-Computer Interfaces (BCIs) technology is driving its evolution towards portability, wearability, and 24/7 monitoring, encountering the core bottleneck of resource constraints on edge devices: terminal devices must meet millisecond-level low-latency requirements under storage budgets ranging from kilobytes (KB) to megabytes (MB) and power budgets in the milliwatt range [1]. Therefore, the decoding algorithm must strike a balance between accuracy, efficiency and power consumption [2].

At present, there are two major ideas for lightweight decoders that people can use: one is the traditional feature engineering based on signal processing. It uses manual design and lightweight classifier and is famous for its transparent calculation and very good energy efficiency [3]. The other is lightweight deep learning based on compact neural network. Because of its powerful machine learning representation, it shows better performance in complex tasks, even if it requires more calculation [4-6]. It is important to note that the lightweight model still has difficulties when tasks or people need to be changed. Usually, they need to retrain new users [7]. Although the new EEG basic model provides a new way to adapt to few examples [8,9], it is still a challenge to adapt to their large number of parameters and make them work on edge devices [10].

This paper aims to systematically review the core algorithm, technical characteristics and performance of these two technical approaches under multi-dimensional constraints. By constructing a systematic trade-off analysis framework, it provides theoretical and practical suggestions for selecting decoder solutions in different edge application scenarios.

2. EDGE DEVICE CONSTRAINTS

Edge BCI devices are typically based on low-power microcontrollers (MCUs) with extremely limited computational capability, memory capacity, and energy budgets. A typical edge device has Random Access Memory (RAM) capacity at the KB level and must meet extremely low power constraints in the mW range [11,12]. In the BCI field, deploying Artificial Intelligence (AI) models onto EEG acquisition devices faces a fundamental contradiction between real-time requirements (< 200ms latency) and computational complexity [1]. Specifically for decoding algorithms, even lightweight networks like EEGNet require significant compression of model size and energy consumption through techniques like 8-bit quantization to be deployed on edge devices [6]. Furthermore, for paradigms like Steady-State Visual Evoked Potential (SSVEP), algorithms based on Canonical Correlation Analysis (CCA) have become the preferred solution for embedded Microcontroller Unit (MCU) platforms owing to their extremely low memory footprint and computational overhead [3]. Emerging neuromorphic computing architectures (e.g., memristor-based decoders [13]) are exploring breakthroughs in energy efficiency from a hardware perspective.

3. CONVENTIONAL FEATURE ENGINEERING METHODS

The early development of brain-computer interface mainly depends on the traditional feature engineering method. These methods extract features with obvious physiological or statistical significance from the original EEG signals and use a series of clear and understandable signal processing steps. Then, they use a lightweight machine learning model for classification and decoding. This method has great advantages in algorithm transparency, computational efficiency and resource requirements. This makes it especially suitable for edge devices with very few resources.

The effectiveness of traditional methods largely depends on the quality of feature extraction. The design of these functions is highly focused on the physiological mechanism of different EEG paradigms. For the motor imaging task, Common Spatial Pattern (CSP) is one of the most commonly used feature extraction algorithms. Its main principle is to create a set of spatial filters to make the variance of EEG signals filtered between different classes as large as possible [14]. Later work improved and developed it [15]; For example, the later developed Filter Bank CSP enhances the robustness of decoding by combining CSP characteristics from multiple frequency bands. Canonical correlation analysis (CCA) has become a reference method for decoding the steady-state visual evolved potential (SSVEP) paradigm. CCA finds the frequency component that causes the strongest response by calculating the canonical correlation coefficient [17] between EEG signal and predefined frequency reference signal. Then, the developed filter bank CCA applies CCA to multiple sub-bands, thus effectively improving the signal-to-noise ratio and recognition rate. Its computational complexity is very low, which makes it an ideal choice for real-time embedded systems [1].

3.1 Lightweight Classifiers and Edge Inference

The second step of traditional feature engineering path is to convert the extracted feature vectors into specific control commands or state categories. This step usually uses a classifier that requires little computation and memory. Linear discriminant analysis (LDA) is one of the most commonly used linear classifiers. It only needs to calculate the average and covariance matrix of training and operation, thus producing a model with few parameters [18]. Support vector machine (SVM) classifies the hyperplane by finding the maximum margin. Even in a few examples and nonlinear problems, they can work well. This cascade of “feature extraction + lightweight classifier” forms a transparent computing pipeline, and its complete complexity is manageable. The total number of its parameters is usually in KB level, which makes it easy to install on different edge hardware [1,12].

3.2 Strengths, Limitations, and Applicable Scenarios

The traditional feature engineering method has several main advantages. First of all, they are very good at computing and energy efficiency. Their algorithms have low complexity, occupy little memory space, and allow real-time decoding with several milliwatts of power consumption in a few milliseconds. Then, they provide a high degree of algorithm transparency and interpretability, because each processing step has a clear physical or statistical significance, which makes debugging and verification easier. Finally, they are very effective for data, and usually only a small amount of calibration data is needed to create an effective custom model.

However, this method also has natural limitations. First of all, it depends on the knowledge of experts and manual design; The effectiveness of features depends largely on the prior knowledge of specific paradigms and signal features, which makes the design process long and difficult to be extended to new tasks. Secondly, its performance ability is limited. For advanced cognitive tasks with complex spatiotemporal dynamics and nonlinear characteristics, the performance may not be as good as the end-to-end deep learning method [19-21]. Thirdly, it is not robust enough between different topics or conversations, and its generalization ability is limited, so it often needs to be recalibrated [7].

3.3 Summary

The traditional feature engineering method has very high computational efficiency, transparent operation and excellent edge deployment ability, which lays a solid foundation for BCI system with less resources. However, their limitations and generalization ability in dealing with complex tasks also encourage the rise of new paradigms such as lightweight deep learning.

4. LIGHTWEIGHT DEEP LEARNING METHODS

With the maturity of deep learning technology, the decoding method based on neural network shows strong representation learning ability in BCI field. In order to adapt to the resource constraints of edge devices, researchers have designed various compact network architectures and model compression technologies, which significantly reduce the computational load while maintaining high accuracy, thus forming a new generation of decoding paradigm for BCI with limited resources.

4.1 Core Network Architectures and Design Principles

The design of lightweight deep learning model follows the principle of “parameter efficiency”. It reduces the complexity of calculation, while maintaining good representation ability. The most famous lightweight convolutional neural network (CNN) model is EEGNet. It uses separable depth convolution instead of conventional convolution. This greatly reduces the number of parameters, while maintaining very good sorting performance [4]. In addition, inspired by the mobile vision model, some studies have introduced the architecture idea of MobileNet into EEG decoding. This makes the model more compact [22].

In order to better capture the long-term dependence in EEG signals, researchers have developed different hybrid architectures. Convolutional Neural Network-Long-term and Short-term Memory (CNN-LSTM) model combines the spatial feature extraction ability of CNN with the time series modeling advantage of LSTM. Recently, lightweight hybrid architecture based on transformer has become an important trend to improve EEG decoding performance. Transformer model, because of its core self-awareness mechanism, effectively captures the long-distance dependence in the sequence [23], and has successfully adapted to the attributes of EEG decoding task. For example, LGFormer [24] combines CNN’s local feature extraction with global transformer dependency modeling. EEG-Deformer [25] designed a layered transformer block “end to end”. The time-frequency layered converter network [26] is specially optimized for the time-frequency characteristics of EEG signals. National spatial models (such as Mamba architecture [27]) have also become a new lightweight choice because of their linear complexity and long sequence modeling ability. This work shows that it is an effective way to design a new generation of lightweight and high-performance EEG decoder by deeply integrating the inductive bias and attention mechanism of the structure.

4.2 Key Technologies for Model Lightweighting

In order to deploy deep learning models on edge devices, researchers have developed different technologies to compress and accelerate the models. For the optimization of network structure, the main methods are: depthwise separable convolution, channel pruning and Knowledge Distraction [28, 29]. For digital quantization and low bit width calculation, the key technologies are: weighted quantization, quantum activation, and more extreme methods, such as binary/ternary network [30]. For hardware optimization, the strategy formulated for a specific hardware platform can further improve the deployment and operation efficiency of the model on the actual equipment [31].

4.3 Advantages, Limitations, and Challenges

The lightweight deep learning method has several important advantages. First of all, they can learn well from representations, that is, they can all learn complex abstract features from raw data. This is much better than human manual selection of features, especially for complex nonlinear patterns [5,19]. Then, end-to-end optimization makes feature creation and classifier design inseparable. Thirdly, these methods have good potential and can be used in other tasks, that is, cross-task generalization. Finally, their architectures are flexible, allowing easy integration of multimodal information or modularity to adapt to different paradigms [32].

However, this method also has limitations and challenges. First of all, it uses a lot of calculation and memory. Even after being simplified, the deep learning model still needs more calculation and memory than traditional methods [10,31]. Secondly, they are very dependent on data, and usually need a lot of labeled data for training. Thirdly, they are difficult to explain, and it is difficult for people to understand how they work. Fourthly, their adaptability to individual differences needs to be improved [7]. Fifth, their training and deployment are very complicated [30].

In terms of edge deployment, Quantized EEGNet has been successfully deployed on Raspberry Pi and Cortex-M series advanced RISC machine (ARM) microcontrollers [1,6]. However, the deployment still faces challenges, such as the trade-off between the accuracy and efficiency of the model and how to adapt to different hardware platforms. It is particularly noteworthy that although the basic models show little learning potential, the basic conflict between their large number of parameters and the limited memory of edge devices has not been solved [10].

4.4 Summary

The lightweight deep learning method represents a major transformation of BCI decoding technology from “manual design” to “machine learning”. Through complex architecture design and multidimensional model compression, they began to be used in marginal environments with limited resources. However, their limitations in efficiency, data dependence and interpretability establish a clear complementary relationship with traditional methods.

5. SYSTEMATIC TRADE-OFF ANALYSIS

When designing BCI edge system, choosing decoding algorithm is a decision-making process. We seek the best balance under many strict constraints. Traditional feature engineering method and lightweight deep learning method represent two different technical paths. This chapter aims to establish a multi-dimensional evaluation framework. He wants to systematically analyze the main trade-offs between these two paths in the edge computing environment. This provides a clear decision-making basis for developers in specific application scenarios.

5.1 Establishment of Evaluation Dimensions

In order to make a comprehensive comparison, we have established six main evaluation dimensions, which directly correspond to the key constraints of edge devices and the actual needs of BCI system, including: 1) The accuracy and performance of the model; 2) Computational efficiency and memory; 3) Energy and energy consumption; 4) Dependence on data and training needs; 5) interpretability and robustness; 6) Complexity of deployment and adaptation.

5.2 Accuracy and Model Performance

In terms of model accuracy and performance, the traditional feature engineering method relies on the domain knowledge of manually designed features, and its upper performance limit is limited by the integrity of feature representation ability. In the paradigm of clear task definition and relatively stable neuron response mode, the traditional method of careful optimization can reach a very high performance saturation point. However, for more complex non-stationary signals, the adequacy of feature extraction is challenging, which leads to a lower performance upper limit.

In contrast, lightweight deep learning method automatically extracts hierarchical features from original or rarely transformed signals through end-to-end learning. Theoretically, they can approach any complex relationship. Lightweight network architecture and its variants show better or equal performance than traditional methods on multiple open data sets [2,20]. Especially for tasks that need to mix multiple data types (multimodal fusion) or involve long sequence relationships (long-distance dependency modeling), attention-based lightweight converter architectures [24,32] show good potential.

However, to achieve these performance advantages, a large number of high-quality training data are needed. The recent system evaluation [10] shows that the advantages of lightweight deep learning model are no longer so obvious in individual cases where data is scarce. They may not even be as powerful as traditional methods.

5.3 Computational and Memory Efficiency

Computational and memory efficiency are very important for edge deployment. The calculation flow of traditional feature engineering method mainly uses linear algebraic operation. The calculation is simple and fast. For example, the classification process combining CSP and LDA only needs matrix multiplication and point product operation. There are few parameters in the model. This makes it easy to deploy on a microcontroller with very few resources and make reasoning in less than one millisecond [15].

Although the design is light, the calculation and memory requirements of deep learning model are much higher than those of traditional methods. Even if the model like EEGNet reduces the number of parameters to tens of thousands, their convolution and complete connection operations still need a lot of floating-point operations (FLOP). In order to be able to use them on edge devices, model compression technology must be used. Weight quantization can reduce the weight and activation of the model, from 32-bit floating-point numbers to 8-bit or less integers, thus reducing storage and memory bandwidth [30]. The quaternion-based EEGNet(Q-EEGNet) [6] is a good example. In addition, deleting unnecessary parameters (adjustment) or using knowledge evaporation to transfer knowledge from large model to small model [28,29] can further reduce the size of the model. Efficient architecture, such as using separable depthwise convolution and botleneck structure, can reduce the computational load of the source. Nevertheless, the computational complexity of quantitative and lightweight models can be ten times higher than that of traditional methods, which requires more computing power and CPU memory bandwidth [10,31].

5.4 Energy Consumption and Power

Power consumption is the lifeline of edge BCI systems, especially portable or implantable devices. Because of its simple calculation and regular memory access mode, the traditional method has low power consumption and extremely short reasoning time, resulting in very low total energy consumption. This makes them suitable for ultra-low power scenarios powered by button cell, which need to run continuously for weeks or even months [1]. Lightweight deep learning methods, even after optimization, have higher computational intensity leading to higher dynamic power consumption. Frequent memory access is a major source of power consumption. Although techniques like quantization can reduce power, their total energy consumption is generally still higher than that of traditional methods. Design must consider the proportion of algorithm energy consumption relative to other modules like sensing and wireless communication [3]. Emerging neuromorphic computing architectures, through event-driven, analog computing, etc., offer a potential ultra-low-power solution for running brain-inspired spiking neural networks, representing an important future direction for breaking the energy-efficiency bottleneck [13].

5.5 Data Dependency and Training Requirements

In terms of data dependency and training requirements, the two methods present a stark contrast. Traditional methods are typically based on statistical signal processing theory, requiring relatively little training data. Some methods can even achieve usable performance after a few calibration trials, making them very suitable for practical scenarios where user calibration time is limited. In contrast, lightweight deep learning, as a data-driven method, has performance positively correlated with data volume. Although the required data amount is less than that for large models, it still requires hundreds or even thousands of labeled trials for adequate training to avoid overfitting, imposing a significant burden for data collection and labeling [19].

Regarding training complexity and cross-subject generalization: training for traditional methods is usually a convex optimization problem, fast and guaranteed to find the global optimum. Deep learning training involves non-convex optimization, requires hyperparameter tuning, is more time-consuming and computationally expensive, and is typically done in the cloud or on high-performance computers. Cross-subject/session generalization is a core challenge in the BCI field. Traditional methods usually require calibration for each user, exhibiting high user dependency. Although deep learning shows potential for cross-subject transfer learning through its powerful representational learning ability, lightweight models, due to their limited capacity, still face severe challenges in this task. Therefore, it is a very important research topic to combine domain adaptation and meta-learning with lightweight model [5,33]. The new basic model paradigm tries to learn the universal representation by training a large number of data, so that it can be adjusted in a few examples. However, in order to determine whether it works well in the edge scene, we need to do more tests [10].

5.6 Interpretability and Robustness

In terms of interpretability, the traditional feature engineering method has natural advantages. For example, the spatial pattern of CSP filter can be traced directly to scalp space, which is interpreted as enhancing or inhibiting the activity of specific brain sources. The characteristics of frequency band have obvious physiological significance. This is helpful to debug the algorithm and study the neural mechanism. Lightweight deep learning method is usually regarded as a “black box”, and its internal decision logic is difficult to understand directly. Although interpretable AI technology has been applied to EEG decoding [20], it is difficult to provide a meaningful explanation as a highly compressed lightweight model.

About robustness: The preparation steps in the traditional method are clear and controllable, so it has good robustness to known noise. Deep learning models, especially end-to-end models, rely on noise types in training data to resist them. When encountering new noise, they sometimes become unstable. For the non-static characteristics of EEG signal, the traditional adaptive algorithm has a perfect theory. It is still an unsolved problem to allow deep learning model to adapt to signal changes online, which brings challenges to the memory calculation and management of edge devices. Recently, the attempt of the mixed frame near infrared functional electroencephalogram (EEG-fNIRS) to decode the BCI inserted and played by stroke patients [33] and commercial headphones [34] shows a possible way to improve the robustness of the system.

5.7 Comprehensive Comparison and Decision Guide

Based on the above analysis, Table 1 summarizes a qualitative comparison of the two technical paths across multiple dimensions.

Table 1. Systematic Trade-off between Conventional Feature Engineering and Lightweight Deep Learning for Edge BCI.

Evaluation Dimension	Conventional Feature Engineering Methods	Lightweight Deep Learning Methods
Accuracy Potential	Moderate, can achieve high saturated performance in specific paradigms	High, especially for complex tasks and large-data scenarios
Computational/Memory Efficiency	Very High, KB-level storage, extremely low FLOPs	Moderate, requires compression (quantization, pruning) to reach KB MB level; computational load still relatively high
Energy Consumption	Very Low, suitable for long-term implantable/wearable devices	Medium to High, dependent on hardware acceleration and optimization
Data Requirement	Low, works with small samples	High, requires large amounts of labeled data for training
Training Complexity	Low, fast convergence, allows online updates	High, requires hyperparameter tuning, typically trained in the cloud
Interpretability	High, clear physical meaning of features and models	Low, black-box model, poor interpretability
Robustness/Generalization	Robust to known noise, but high user dependency	May generalize better through large-data learning, but sensitive to unseen noise
Deployment Complexity	Low, easy to implement on MCUs	Medium to High, requires specialized inference engines and optimization toolchains

The decision is not a binary choice but should be based on the priority constraints of the application scenario. In making a specific decision, developers should choose based on the core constraints of the application scenario. If hardware resources are extremely limited, there are extreme demands for power consumption and battery life, available calibration data is minimal, the system requires high interpretability and determinism, and the task paradigm is mature, then conventional feature engineering methods are a more pragmatic choice. Conversely, if higher decoding accuracy is pursued, traditional methods have reached a performance bottleneck, the task is complex with features difficult to design manually, sufficient training data and cloud training resources are available, the target hardware has certain acceleration capabilities, and the system allows for regular charging or has a relatively lenient power budget, then lightweight deep learning methods are more advantageous.

Smarter edge BCIs in the future may adopt dynamic or hybrid strategies. For example, using lightweight traditional methods at system startup or when resources are tight, and switching to or fusing with lightweight deep learning models after accumulating sufficient data [35]. Another approach is to use deep learning to automatically generate or optimize the feature extractors of traditional methods, combining the strengths of both. In terms of task complexity, advanced tasks like visual reconstruction and language decoding impose higher demands on the model's representational capacity and contextual modeling ability, which is often an area where lightweight deep learning holds an advantage.

5.8 Chapter Summary

This chapter systematically compares the trade-offs between conventional feature engineering and lightweight deep learning in edge BCI decoding tasks. Traditional methods hold incomparable advantages in efficiency, power consumption, and interpretability, making them a pragmatic choice in extremely resource-constrained scenarios. Lightweight deep learning methods represent a higher performance ceiling and automation potential, but they demand more data, computing power, and energy. There is no "one-size-fits-all" optimal solution. The designer's core task is to conduct a multi-dimensional trade-off analysis under clear edge constraints and application requirements to make the most suitable technology selection.

6. CHALLENGES AND FUTURE DIRECTIONS

Based on the aforementioned systematic trade-off analysis, this chapter summarizes the core challenges currently faced in decoder design for resource-constrained edge BCIs and looks forward to potential future technological breakthroughs and development paths.

6.1 Current Core Challenges

Although both conventional feature engineering and lightweight deep learning have made significant progress, achieving high-performance, highly robust decoding while meeting the stringent constraints of edge devices still presents many challenges to be resolved.

6.1.1 The "Impossible Triangle" of Accuracy, Efficiency, and Energy Consumption

Under the strict limitation of computing power, memory and power consumption, the decoder design has long faced the dilemma of "impossible triangle": it is difficult to achieve high precision, low delay and ultra-low power consumption at the same time. Traditional methods sacrifice some accuracy potential in order to achieve high efficiency. Although the lightweight deep learning method solves this triangle by compression, its quantization, preprocessing and other operations usually lead to the loss of accuracy, and the final energy efficiency ratio is still difficult to match the traditional solution [30,31]. Finding a new computing paradigm or hardware architecture to break this triangle is a fundamental challenge.

6.1.2 The Contradiction Between Data Scarcity and Personalized Generalization

EEG signals show great differences between people and change with the passage of time. The traditional method needs to be customized for everyone, which is unrealistic for us. Although lightweight models of deep learning can better represent data, training them still needs a lot of labeled data. However, it is expensive to collect high quality EEG data. This creates a problem: there is little data, but the model needs to be popularized well. Existing methods such as transfer learning and domain adaptation usually do not perform well on very lightweight models, and they require too much calculation for edge devices [7]. Knowing how to quickly adapt to new users with little online data is a big problem for practical applications. Although the new EEG basic model has trained a large number of unlabeled data, it has gained a powerful general representation ability and provided a new way to adapt to a small amount of data. How to reduce (evaporate) or adjust these huge models so that they can keep the number of KB-MB in peripheral devices is still an unsolved challenge [10].

6.1.3 Lack of Model Interpretability and Trust in Reliability

In high-risk situations, such as medical assistance, the understandability and reliability of decoding decisions are very important. The traditional feature engineering method has natural advantages, but the "black box" nature of deep learning model makes its decision logic difficult to follow, thus reducing the confidence of clinicians and users [20]. It is very important to develop a low-complexity model interpretation tool suitable for edge devices and establish a reliability evaluation index based on interpretable features to promote the adoption of BCI.

6.1.4 Algorithm Robustness in Dynamic Environments

The real edge environment is full of ever-changing noises. At present, most algorithms run well on clean static laboratory data sets, but they are not very robust in dynamic interference. The traditional pretreatment pipeline must be adjusted for each new noise. As for deep learning models, they need training data containing various noise patterns, which is almost impossible to have completely. It is a very challenging challenge to design a reliable edge reasoning algorithm, which can perceive the signal quality online and adjust its parameters or structure adaptively.

6.1.5 Lack of Deep Hardware-Software Co-design Optimization

At present, the research usually regards algorithm design and hardware deployment as two independent steps: first, develop the model on a common computer, and then deploy it to edge devices through a compression chain. This "design first, then adjust" model can't make full use of the performance potential of the underlying hardware, so it is difficult to achieve the best energy efficiency. Real hardware and software collaborative design-creating a special hardware accelerator for a specific neural network operator or EEG feature extraction pipeline-is still very new, but it is the key to release all performance potential [1].

6.2 Future Research Directions

In order to meet the above challenges, future research can make breakthroughs in the following aspects, and help the edge BCI decoder become more efficient, smarter and more reliable.

6.2.1 EEG Foundation Models and Edge Adaptation

Learning from the success in natural language processing and computer vision, building foundation models for EEG signals and researching their lightweight adaptation to edge devices is a revolutionary path to solving data scarcity and generalization problems. Future research should focus on: 1) Synergy of General and Specialized Foundation Models: On one hand, continue exploring universal EEG representation learning, using massive multi-task, multi-subject EEG data to pre-train large-scale Transformer or State Space Models, learning universal neural representations with strong generalization capabilities [8,9]. On the other hand, developing special lightweight basic models for specific paradigms may be more effective than general models for specific edge tasks. 2) Equipment with limited resources for high-efficiency fine adjustment strategy of edges: We are looking for methods for high-efficiency fine adjustment of parameters, dynamic dispersion, knowledge evaporation [29] and test time training. The goal is to compress the knowledge of the basic model into the ultra-light student network. This realizes an effective example of “pre-training in the cloud, personalization at the edge”.

6.2.2 Brain-Inspired and Ultra-Lightweight Hybrid Architecture Design

Exploring the ultra-lightweight architecture inspired by neuroscience, beyond the current convolution and attention paradigm, is another important direction. 1) Spiking Neural Networks and Neuromorphic Hardware: SNN, with its event-driven and sparse computing characteristics, is expected to consume less energy than the traditional deep learning model on the neuromorphological chip [13]. 2) Dynamic neural networks: They allow the model to dynamically adjust its depth or width according to the complexity of the input signal or the width of the remaining battery of the equipment, so as to strike a balance between accuracy and efficiency. 3) Heterogeneous hybrid architectures: They organically combine traditional signal processing modules with lightweight neural modules. Traditional modules provide the main characteristics of stability and low power consumption, while neural modules provide advanced abstraction, which ensures the efficiency foundation and improves the maximum performance.

6.2.3 Few-Shot and Unsupervised Edge Learning

Reducing the dependence on a large number of tag data is very important to achieve inclusive BCI. This needs to be developed: 1) edge meta-learning: develop a meta-learning algorithm suitable for MCU, allowing the model to quickly adapt to new users after several tests, and the adaptation process can be completed on the device. 2) Self-monitoring and online self-learning: using unlabeled continuous EEG data to learn the representation on edge devices, combined with continuous learning technology, the model can be updated online using newly generated data streams without forgetting the old knowledge, thus gradually improving the personalized performance. 3) Co-evolution driven by joint learning: multiple edge BCI devices are allowed to cooperate to train a shared and powerful basic model, while protecting users' privacy. Each device stores a personalized layer locally, realizing the unity of “collective intelligence” and personal adaptation [12,35].

6.2.4 Resource-Aware System-Level Optimization

The future decoder design must change from passive adaptation to “resource constraints” to active management centered on “resource awareness”. This requires: 1) developing a cross-layer optimization tool chain: creating a unified compilation and deployment framework, which can automatically search model architecture, compression strategy and scheduling strategy according to the precision-delay-power multi-objective Pareto boundary specified by users, and generate the best executable file. 2) Develop scalable energy precision decoding: allow the decoder to dynamically switch operation modes according to the current battery capacity or the urgency of the task, and realize intelligent management of system running time. 3) Sensor-computing collaborative design: Optimize the source of signal acquisition, for example, combine with dynamic sampling rate analog-to-digital converter (ADC), sparse sensor network, and enable high-power computing module only when relevant neuron events are detected, thus saving system-level energy.

6.2.5 Open Benchmarks and Standardized Evaluation

In order to promote the healthy development of this field, it is urgent to establish a standardized evaluation system for marginal BCI. 1) Establish an open source hardware-in-the-loop benchmark platform: not only provide data sets, but also provide a reference platform containing typical edge hardware, requiring the algorithm to submit the final executable file, so as to evaluate the end-to-end accuracy, delay and power consumption under unified conditions. 2) Develop multi-dimensional evaluation criteria: mandatory inclusion of edge-related indicators, such as energy efficiency, memory occupancy rate and start-up calibration time, and encouragement to report the interpretable analysis results of the model, so as to establish comprehensive evaluation indicators beyond the classification accuracy.

6.3 Summary and Decision Philosophy

In the future, deciding how to choose the decoding path for BCI with less resources will be more and more scene-specific, dynamic and system-specific. The idea of “one algorithm rules all” is no longer valid. The core idea of decision-making should be changed from a fixed argument about the technical route to a dynamic and intelligent resource planning. Developers should first clarify the first principle constraint of the target application scenario, and then use it as the main driving force for choosing technology. The most successful future edge BCI system may be a “hybrid intelligent agent”: it combines several lightweight decoders and has a built-in resource-aware scheduler, which can automatically select or merge the best decoding strategy according to the real-time environment, signal quality, hardware status and user’s intention, and constantly find the best balance point in the complex multi-dimensional constraints.

REFERENCES

[1] Nguyen M-D, Do T, Tran X-T, Nguyen Q-T and Lin C-T 2025 Edge AI-brain-computer interfaces system: A survey IEEE Transactions on Neural Systems and Rehabilitation Engineering 33 4051-66.

[2] Li Y, Chen E, Xiao X, Xu M and Ming D 2025 Lightweight deep learning models for EEG decoding: A review Journal of Neural Engineering 22 061004.

[3] Nguyen M-D, Do T, Le N T T, Tran X-T, Chang F and Lin C-T 2026 EdgeSSVEP: A fully embedded SSVEP BCI platform for low-power real-time applications arXiv:2601.01772 [Preprint]. Retrieved May 19, 2026, from https://arxiv.org/abs/2601.01772

[4] 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 056013.

[5] Hallal L, Rhinelander J, Venkat R and Newman A 2026 Efficient feature extraction for EEG-based classification: A comparative review of deep learning models AI 7 50.

[6] Schneider T, Wang X, Hersche M, Cavigelli L and Benini L 2020 Q-EEGNet: An energy-efficient 8-bit quantized parallel EEGNet implementation for edge motor-imagery brain-machine interfaces Proceedings of the 2020 IEEE international conference on smart computing (SMARTCOMP) (Bologna, Italy: IEEE) pp. 284-289.

[7] Yao Y, Wang H, Chen L, Peng Y and Luo J 2025 Foundation models for EEG decoding: Current progress and prospective research Journal of Neural Engineering 22 061002.

[8] Wang J, Zhao S, Luo Z, Zhou Y, Jiang H, Li S, Li T and Pan G 2025 CBraMod: A criss-cross brain fo undation model for EEG decoding arXiv:2412.07236 [Preprint]. Retrieved May 19, 2026, From https://arxiv. org/abs/2412.07236

[9] Zhou Y, Wu J, Ren Z, Yao Z, Lu W, Peng K, Zheng Q, Song C, Ouyang W and Gou C 2025 CSBrain: A cross-scale spatiotemporal brain foundation model for EEG decoding arXiv:2506.23075 [Preprint]. Retrieved May 19, 2026, from https://arxiv.org/abs/2506.23075

[10] Precision Neuroscience 2026 BrainStorm 2026 hackathon: Edge-AI for high-density ECoG decoding.

[11] Capogrosso L, Cunico F, Cheng D S, Fummi F and Cristani M 2024 A machine learning-oriented survey on tiny machine learning IEEE Access 12 23406-26.

[12] Kallimani R, Pai K, Raghuwanshi P, Iyer S and López O L A 2024 TinyML: Tools, applications, challenges, and future research directions Multimedia Tools and Applications 83 29015-45.

[13] Liu Z, Wu H, Wang Y, Li Y, Zhang Y, Wang Z, Chen Y and Li H 2025 A memristor-based adaptive neuromorphic decoder for brain-computer interfaces Nature Electronics 8 15-28.

[14] Ramoser H, Müller-Gerking J and Pfurtscheller G 2000 Optimal spatial filtering of single trial EEG during imagined hand movement IEEE Transactions on Rehabilitation Engineering 8 441-6.

[15] Blankertz B, Tomioka R, Lemm S, Kawanabe M and Müller K-R 2008 Optimizing spatial filters for robust EEG single-trial analysis IEEE Signal Processing Magazine 25 41-56.
[16] Ang K K, Chin Z Y, Zhang H and Guan C 2008 Filter bank common spatial pattern (FBCSP) in brain-computer interface Proceedings of the 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence) (Hong Kong, China: IEEE) pp. 2390-2397.
[17] Chen X, Wang Y, Gao S, Jung T-P and Gao X 2015 Filter bank canonical correlation analysis for implementing a high-speed SSVEP-based brain-computer interface Journal of Neural Engineering 12 046008.
[18] 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 031005.
[19] Craik A, He Y and Contreras-Vidal J L 2019 Deep learning for electroencephalogram (EEG) classification tasks: A review Journal of Neural Engineering 16 031001.
[20] 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 5391-420.
[21] Dai S, Xiao H, Yu S and Ye Q 2026 Autoregressive visual decoding from EEG signals arXiv:2602.22555 [Preprint]. Retrieved May 19, 2026, from https://arxiv.org/abs/2602.22555
[22] Sandler M, Howard A, Zhu M, Zhmoginov A and Chen L-C 2018 MobileNetV2: Inverted residuals and linear bottlenecks Proceedings of the 2018 IEEE conference on computer vision and pattern recognition (CVPR) pp. 4510-4520.
[23] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L and Polosukhin I 2017 Attention is all you need Advances in Neural Information Processing Systems 30.
[24] Yang W, Wang X, Qi W and Wang W 2025 LGFormer: Integrating local and global representations for EEG decoding Journal of Neural Engineering 22 026042.
[25] Ding Y, Li Y, Sun H, Liu R, Tong C, Liu C, Zhou X and Guan C 2025 EEG-Deformer: A dense convolutional transformer for brain-computer interfaces IEEE Journal of Biomedical and Health Informatics 29 1909-18.
[26] Yue C, Hu H, Shi E and Zhang S 2024 TF-HiTNet: A temporal-frequency hierarchical transformer network for EEG motor imagery classification Proceedings of the 2024 IEEE international conference on bioinformatics and biomedicine (BIBM 2024) (Lisbon, Portugal: IEEE) pp. 3961-3965.
[27] Wang J, Zhao S, Luo Z, Zhou Y, Li S and Pan G 2025 EEGMamba: An EEG foundation model with Mamba Neural Networks 192 107816.
[28] Hinton G E, Vinyals O and Dean J 2015 Distilling the knowledge in a neural network arXiv:1503.02531 [Preprint]. Retrieved May 19, 2026, from https://arxiv.org/abs/1503.02531
[29] Ren S, Duan S, Xu X and Liu Y 2024 Scheduled knowledge acquisition on lightweight vector symbolic architectures for brain-computer interfaces Proceedings of the tinyML research symposium 2024 pp. 1-5.
[30] Gholami A, Kim S, Dong Z, Yao Z, Mahoney M W and Keutzer K 2021 A survey of quantization metho ds for efficient neural network inference arXiv:2103.13630 [Preprint]. Retrieved May 19, 2026, from https:// arxiv.org/abs/2103.13630
[31] Lin J, Chen Y, Pan R, Chen D, Li T and Wang M 2023 Deploying deep learning models on microcontrollers: A survey IEEE Transactions on Neural Networks and Learning Systems 34 4929-51.
[32] He Y, Wang X, Yang Z, Xue L, Chen Y, Ji J, Wan F, Mukhopadhyay S C, Men L and Tong M C F 2023 Classification of attention deficit/hyperactivity disorder based on EEG signals using a EEG-transformer model Journal of Neural Engineering 20 056013.
[33] Chen D, Shi J, Tao B, Zhao X, Zhao Z, Li S, Xu Y, Ding T, Zhang P, Ye Q, Chen K, Wu Z, Tang Y, Jiang W, Shu K, Huang L, You Z, Zhang P and Tang Z 2025 A novel transfer learning-based hybrid EEG-fNIRS brain-computer interface for intracerebral hemorrhage rehabilitation Advanced Science 12 e05426.
[34] Mezzina G and De Venuto D 2024 Towards plug and play and portable BCIs: Embedding artifacts rejection and machine learning on wireless EEG headset Proceedings of the 54th annual meeting of the italian electronics society Lecture notes in electrical engineering vol 1113, ed C Ciofi and E Limiti (Cham, Switzerland: Springer) pp. 173-185.
[35] Ghosh S, Bhuvanakantham R, Sindhujaa P, Harishita P B, Mohan A, Gulyás B, Máthé D and Padmanabhan P 2026 A cloud-aware scalable architecture for distributed edge-enabled BCI biosensor system Biosensors 16 157.