Biomedical Signal Processing and Control 100 (2025) 107000

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

Constructing lightweight and efficient spiking neural networks for EEG-based motor imagery classification ☆

Xiaojian Liao [a], Guang Li [b], You Wang [b], Lining Sun [a], Hongmiao Zhang [a,*]

[a] The Jiangsu Provincial Key Laboratory of Advanced Robotics, School of Mechanical and Electric Engineering, Soochow University, Suzhou 215000, Jiangsu, China
[b] The State Key Laboratory of Industrial Control Technology, Institute of Cyber Systems and Control, Zhejiang University, Hangzhou 310027, Zhejiang, China

ARTICLE INFO

Keywords:
Spiking neural network
Brain–computer interface
ANN to SNN conversion
Motor imagery

ABSTRACT

Spiking neural networks (SNNs) are gaining attention across various fields, including EEG-based brain–computer interfaces. SNNs utilize spike sequences to characterize and convey information, offering a more bio-interpretable approach and consuming less energy than artificial neural networks (ANNs). However, the binary nature of spike sequences makes it challenging for SNNs to be trained directly using the backpropagation method like ANNs. A prevalent SNN training method involves converting ANNs to SNNs, while the converted SNN often suffers severe performance degradation problems. We theoretically derive the ANN to SNN conversion conditions. Based on this, we comprehensively analyze the sources of conversion errors, and it seems that excessive ANN parameters may result in accuracy loss during conversion. Building upon this analysis, we propose a lightweight and efficient neural network (LENet) for motor imagery (MI) classification. Specifically, LENet can effectively capture spatiotemporal features while replacing the traditional fully connected layer (FCL) with the classification convolution block (CCB), which can reduce the model parameters compared to FCLs. Through experiments on two MI datasets (BCI IV-2a and IV-2b), LENet not only effectively reduces conversion errors but also adeptly balances parameters and performance. The results show that LENet outperforms state-of-the-art classification methods. Meanwhile, our proposed CCB effectively improves the performance of baseline methods compared to FCL. Furthermore, through an analysis of energy consumption, we observe a substantial reduction in power consumption for SNNs compared to ANNs. This work introduces novel perspectives for future SNN application scenarios for the non-stationary MI signals.

1. Introduction

Brain-Computer Interface (BCI), which allows the brain to generate independent information exchange and control with the outside world and enhance communication and interaction between the brain and the environment [1], has become a vital research direction. There are various methods to acquire the brain signal in the BCI system, such as functional near-infrared (fNIR), functional magnetic resonance imaging (fMRI), and electroencephalogram (EEG). EEG has the advantages of greater portability, convenience, and lower cost than other methods, so it is most commonly used in BCI systems.

Currently, motor intention decoding is a typical application scenario for EEG-based BCI systems. For example, through motor intention decoding, BCIs allow people with movement disorders to enable control between the brain and devices by imagining specific actions, which is widely used in the field of rehabilitation [2,3].

Due to raw EEG signals being characterized by low spatial resolution and low signal-to-noise ratios [4], this poses a great challenge in developing effective BCIs. Early studies mainly focused on feature extraction based on experience and then utilized machine learning to complete the EEG signal decoding [5,6]. However, these classical methods tend to lose the inherent characteristics of the EEG signal when extracting features and are highly dependent on artificial feature generation.

Neural networks can automatically capture the effective information of EEG during the training process, avoiding the labor cost consumed by manually extracting features in traditional machine learning methods. Therefore, with the rapid development of hardware technology, artificial neural networks (ANNs) have made significant progress in image recognition [7] and natural language processing [8] and pushed forward the field of EEG-based BCI [9]. For example, Liu et al. [10] proposed a convolutional neural net called FBMSNet, designed from both spatial and temporal dimensions. Multi-scale temporal features

☆ This work was partially supported by the National Key Research and Development Program of China under Grant 2023YFB4704600.
* Corresponding author.
E-mail address: zhanghongmiao@suda.edu.cn (H. Zhang).

https://doi.org/10.1016/j.bspc.2024.107000
Received 4 February 2024; Received in revised form 9 September 2024; Accepted 30 September 2024
Available online 7 October 2024

are first extracted using mixed depthwise convolution, followed by spatial filtering. Li et al. [11] proposed a multilayered spatiotemporal feature-based fusion of CNN and LSTM networks, in which CNN and LSTM extract spatial and temporal features, respectively. Finally, all the features are merged into a fully connected layer.

However, relevant biological studies have revealed that, unlike traditional ANNs, biological neural networks communicate through discrete spikes rather than specific values. The spiking neural network (SNN) [12] was proposed by researchers, which primarily utilizes discrete action potentials (spikes) to characterize and transmit information, offering a potential avenue for better understanding the workings of the brain [13]. Moreover, owing to its temporal dynamics feature, SNN facilitates the processing of spatiotemporal information through spike-based learning and memory mechanisms [14]. It proves to be efficient in processing brain signals, and the integration of SNN into BCI tasks has gained popularity in recent years. Besides, Leveraging the characteristics of SNNs, which rely on spikes and a sparse temporal communication mechanism, enables low-power inference on specific neural morphological hardware platforms such as TrueNorth [15], SpiNNaker [16], Loihi [17], and others. Currently, only some results have been achieved in simple tasks. For example, Kasabov [18] proposed a 3D evolved SNN architecture called NeuCube, where analysis of the internal structure of a trained NeuCube model can reveal meaningful spatial and temporal relationships in the data. Subsequently, researchers have proposed methods based on the NeuCube architecture for motor intention decoding [19], emotion recognition [20], etc. Liao et al. [21] proposed an SNN model called SCNet, which deeply integrates the features of CNN and SNN and utilizes adaptive coding to capture the information of EEG effectively. Gong et al. [22] proposed a novel SNN model called SGLNet, which combines spike-based adaptive graph convolution and LSTM to achieve high accuracy in motor imagery (MI) classification and emotion decoding.

Despite extensive research in the field, SNNs still face challenges in matching the performance of ANNs in common BCI tasks. This challenge arises due to the complexity of training a deep SNN compared to an ANN, as spikes are non-differentiable, posing difficulties for direct backpropagation (BP) [23]. Recent studies aim to improve SNN performance by incorporating structures and learning rules from ANNs. Some researchers [24–27] propose converting trained ANNs to SNNs, aligning their outputs and network structures as closely as possible. While this approach can lead to effective deep SNNs, the converted SNN often suffers severe performance degradation problems. Specifically, since current ANN performance is positively correlated with parameters, existing ANNs usually have many parameters. However, ANN with too many parameters may increase the loss of accuracy during the conversion process. Therefore, designing a lightweight model for efficient EEG signal decoding becomes crucial.

To address these issues, we propose a lightweight and efficient neural network (LENet) based on the ANN to SNN conversion method for MI classification. Specifically, building upon the theoretical foundation of ANN to SNN conversion, we further analyze the two conversion errors that propagate with the model weights and conclude that an excessive number of ANN parameters may increase accuracy loss during the conversion. Consequently, we propose the LENet, which replaces the traditional fully connected layer by classification convolution block, not only effectively reduces conversion errors but also adeptly balances parameters and performance. Our contributions are mainly summarized as follows:

- We theoretically derive the conditions for ANN to SNN conversion and conduct a detailed analysis of the conversion errors that propagate with the model weights. Our analysis reveals that each layer of neurons introduces two distinct types of inherent errors.
- The proposed LENet effectively balances model parameters and performance. In comparison to models with more parameters, LENet significantly reduces errors after conversion to SNN.
- Our proposed classification convolution block proves effective not only for LENet but also for other baseline models.
- Extensive experiments confirm our analysis and show the effectiveness of the proposed method, which outperforms other state-of-the-art methods on two MI datasets.

The rest of the paper is organized as follows: Section 2 describes the process of ANN to SNN conversion, analyzes the two errors inherent in the conversion process, and the architecture of LENet. The experimental setting and results are presented in Sections 3 and 4. A discussion is presented in Section 5, followed by a summary in Section 6.

2. Methods

In this section, we begin by introducing the EEG data processing procedure and its representation. Subsequently, we conduct a theoretical derivation of the conversion from ANN to SNN. Building upon this derivation, we analyze the two errors inherent in the conversion process, which propagate with the model weights. Following this, we describe the LENet architecture in detail. Finally, we elaborate on the neural network training methodology.

2.1. EEG preprocessing and representation

We employ common preprocessing methods for MI signals, which are outlined as follows: For the BCI IV-2a and IV-2b datasets, considering that the primary energy activity band of the MI signal falls within the 7 Hz to 35 Hz range, we apply a bandpass filter to isolate the MI signal within this frequency range.

The preprocessed EEG signals were input into the model. Formally, $X \in R^{C\times T}$ denotes the raw EEG signal of one trial, and $Y \in \{0, 1, \ldots, N_c - 1\}$ denotes its corresponding label where C denotes the number of channels of the EEG signal, T denotes the sampling point of the EEG signal, and N_c denotes the number of categories for that experiment.

2.2. Conversion from ANN to SNN

In this section, we initially introduce the neuron models for both ANN and SNN. Subsequently, we undertake the theoretical derivation of the ANN-SNN conversion. The conversion and inference framework is illustrated in Fig. 1. The fundamental idea is to align the output activation value of the ANN neuron with the firing rate of the SNN neuron. A prevalent strategy involves converting Rectified Linear Unit (ReLU) functions to Integrate-and-Fire (IF) neurons. Building upon this derivation, we analyze the two errors inherent in the conversion process, which propagate with the model weights.

2.2.1. Neuron model for ANNs

The computation of neuron models for artificial neural networks can be reduced to a combination of linear transformations and nonlinear mappings.

$$a_l = max\left(w_l^{ANN} a_{l-1} + b_l^{ANN}, 0\right) \tag{1}$$

where a_l is the nonlinear output value of the lth layer based on the $ReLU$ activation function, w_l^{ANN} is the weight of the (l)-th layer neuron, and b_l^{ANN} is the bias term of the lth layer.

2.2.2. Neuron model for SNNs

Similar to previous work, we use the IF neuron in the SNN as a model for converting the ReLU activation function in the ANN. We use Eqs. (2)–(4) to describe the dynamics of this IF neuron as follows:

$$m_l(t) = v_l(t-1) + w_{l-1}^{SNN} s_{l-1}(t) + b_{l-1}^{SNN} \tag{2}$$

$$s_l(t) = H\left(m_l(t) - v_{th,l}\right) = \begin{cases} 1, m_l(t) \geq v_{th,l} \\ 0, m_l(t) < v_{th,l} \end{cases} \tag{3}$$

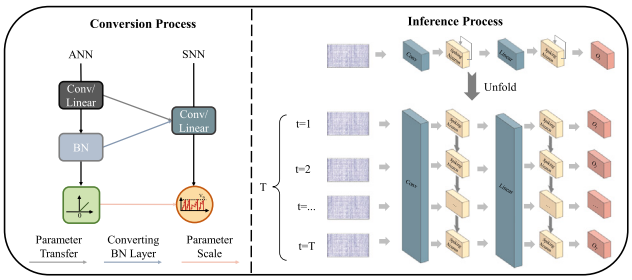


Fig. 1. Illustration of the ANN to SNN conversion and SNN inference.

$$v_l(t) = m_l(t) - v_{th,l}s_l(t) \tag{4}$$

where $m_l(t)$ and $v_l(t)$ denote the membrane potential before and after the trigger spike at time step t, respectively. $v_{th,l}$ is the firing threshold of the neuron of the lth layer. b_{l-1}^{SNN} denotes the bias current of the neuron of the $(l-1)$-th layer, and w_{l-1}^{SNN} denotes the weight between the neuron in $(l-1)$-th and lth layer. $H(\cdot)$ is the Heaviside step function, and $s_l(t)$ denotes the output value of the neuron at time step t, which is equal to 1 if there is a spike and 0 otherwise. Also, here, we use the term 'soft reset' instead of 'hard reset'. After neuron firing, the membrane potential $v_l(t)$ decreases by an amount equal to the firing threshold $v_{th,l}$ rather than returning directly to the reset value.

2.2.3. ANN to SNN conversion

The key idea of the ANN to SNN conversion is to map the activation values of the simulated neurons in the artificial neural network to the firing rates of the spiking neurons in the SNN. Specifically, the membrane potential update equation is obtained by combining Eq. (2) and Eq. (4) as follows:

$$v_l(t) - v_l(t-1) = w_l^{SNN}s_{l-1}(t) + b_l^{SNN} - v_{th,l}s_l(t) \tag{5}$$

Summing Eq. (5) from time 1 to T, divide both sides by T as follows:

$$\frac{v_l(T) - v_l(0)}{T} = \frac{w_{l-1}^{SNN}\sum_{i=1}^{T}s_{l-1}(i)}{T} + b_{l-1}^{SNN} - \frac{v_{th,l}\sum_{i=1}^{T}s_l(i)}{T} \tag{6}$$

Dividing both sides of Eq. (6) by $v_{th,l}$ and using $r_l(T) = \frac{\sum_{i=1}^{T}s_l(i)}{T}$ to represent the firing rate of lth layer neurons.

$$r_l(T) = \frac{w_l^{SNN}r_{l-1}(t) + b_l^{SNN}}{v_{th,l}} - \frac{v_l(T) - v_l(0)}{Tv_{th,l}} \tag{7}$$

Since $s_l(t)$ can only take the value of 0 or 1, the firing rate $r_l(t)$ is in the range of $[0, 1]$. When $T \to \infty$, it is easy to conclude:

$$r_l(T) = clip\left(\frac{w_l^{SNN}r_{l-1}(T) + b_l^{SNN}}{v_{th,l}}, 0, 1\right) \tag{8}$$

where $clip(x,\ 0, 1) = x$ when $x \in [0, 1]$, $clip(x,\ 0, 1) = 1$ when $x > 1$, and $clip(x,\ 0, 1) = 0$ when $x <\ 0$.

From Eq. (1), we assume that the upper bound of the output of all ReLU-based artificial neurons in lth layer is $\lambda_l = max\{a_l\}$, we have:

$$0 \le a_l \le \lambda_l \tag{9}$$

Let $z_l = \frac{a_l}{\lambda_l}$, then $0 \le z_l \le 1$ $(l\ =\ 1, 2, \ldots,\ L)$. lth layer and $(l-1)$-th layer activations satisfy:

$$z_l = clip\left(\frac{w_l^{ANN}z_{l-1}\lambda_{l-1} + + b_l^{ANN}}{\lambda_l}, 0, 1\right) \tag{10}$$

Comparing Eq. (8) and Eq. (10), we can conclude that if Eq. (8) is equal to Eq. (10) the following equation holds:

$$\frac{w_l^{SNN}}{v_{th,l}} = \frac{w_l^{ANN}\lambda_{l-1}}{\lambda_l}, \frac{b_l^{SNN}}{v_{th,l}} = \frac{b_l^{ANN}}{\lambda_l} \tag{11}$$

From Eq. (11), it can be seen that converting the model from the ANN structure to the SNN structure requires scaling operations, such as scaling the weights (i.e., weight normalization) or setting a threshold (i.e., threshold balancing). Therefore, we can consider weight normalization as equivalent to threshold balancing to some extent.

Batch Normalization (BN) was initially introduced for ANN to expedite training and convergence, with the objective of normalizing the ANN output to an average value of 0, which is contrary to SNN characteristics. Consequently, we propose the integration of the Batch Normalization parameters into the weights w and bias b of the previous parameter layers (Linear, Conv2d). For this, the new weights $\bar{w}$ and bias $\bar{b}$ are:

$$\bar{w} = \frac{\gamma}{\sigma}w \tag{12}$$

$$\bar{b} = \frac{\gamma}{\sigma}(b - \mu) + \beta \tag{13}$$

where μ, σ are the running mean and standard deviation, and γ, β are the parameters after the BN layer transformation.

2.2.4. Conversion error

Based on the aforementioned derivation, a thorough analysis reveals that the conversion from ANN to SNN is susceptible to the following errors:

- **Timestep error**: To satisfy Eq. (8) based on the requirement in Eq. (7), it becomes necessary to allow the time step, denoted as T, to approach infinity. However, achieving this condition in practice is not feasible, inevitably resulting in an error of magnitude $\frac{v_l(T)-v_l(0)}{Tv_{th,l}}(l = 1, 2, \ldots, L)$ for neurons at each layer.
- **Quantization error**: The firing rate of spiking neurons in the lth layer is given by $r_l(T) = \frac{\sum_{i=1}^{T}s_l(i)}{T}$. This is a discontinuous step value, taking values only as $\frac{k}{T}(0 \le k \le T)$. On the other hand, after normalization, the output value $z_l = \frac{a_l}{\lambda_l}$ of artificial neurons in the lth layer is continuous. There is an inevitable quantization error between the step function and the continuous values. This discrepancy arises due to the inherent differences in the nature of these representations.

Observing the errors above, it is evident that both exist among neurons in each layer, persistently propagating with the weights. Therefore, it is crucial to design a lightweight model for converting ANNs to SNNs, which is essential to minimize the errors inherent in the conversion process and ensure the fidelity of information transfer between the layers.

2.3. Architecture of LENet

Building on the preceding analysis, it becomes apparent that the design of a lightweight decoding model is crucial. To balance the relationship between parameters and performance, we introduce LENet, a lightweight and efficient neural network for decoding MI signals.

LENet is a fully convolutional neural network comprising four distinct convolution blocks: (1) temporal convolution block, (2) spatial convolution block, (3) feature fusion convolution block, and (4) classification convolution block. The initial convolutional block acts as the sensory field of the network and the time domain processing unit for EEG signals. Initially, three convolutions with different kernel sizes extract temporal features from each EEG signal channel, followed by the concatenation and fusion of the features. The second convolutional block focuses on learning spatial features from the fused time-feature maps. The third block consolidates all feature maps using separable convolutions. Subsequently, pointwise convolution is applied to match the number of feature maps with the number of categories. Finally, a global average pooling is employed to fuse information within each feature map, treating each map as category confidence corresponding to a specific category. The overview and detailed configuration of LENet is shown in Fig. 2. The specific description of each Convolution block is as follows:

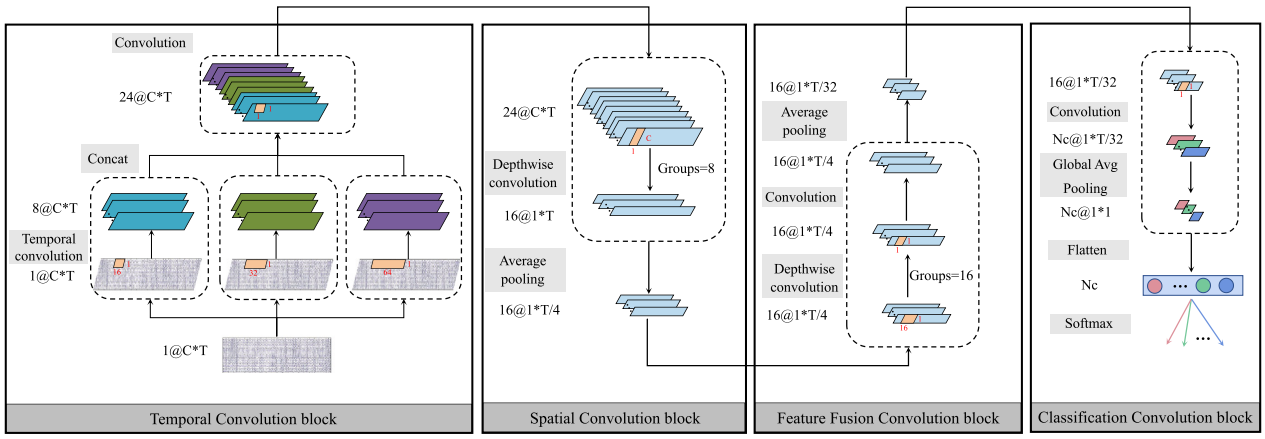


Fig. 2. The architecture of LENet, where C is the number of channels, T is the number of time points, N_C is the number of classes.

2.3.1. Temporal convolution block

The dimension of a one-dimensional convolution in the time domain is intricately linked to the sampling frequency of the signal [28]. To better extract temporal information at various scales in EEG, we leverage three convolutions of different sizes as the receptive field of the network and the time-domain processing unit of the EEG. These convolutions serve as effective temporal filters operating at different scales to extract temporal features from each channel. Subsequently, the three sets of feature maps obtained are concatenated to yield novel time series features. We employ the pointwise convolution to capture the significance of these time series features. A Batch Normalization (BN) layer follows each convolution operation in this block.

2.3.2. Spatial convolution block

The block initiates with a depthwise convolution. Following the convolution layer, a sequence of operations is applied, including a BN layer, the Rectified Linear Unit (ReLU) activation function, and a dropout layer. Simultaneously, to diminish the dimensionality of the feature map, an average pooling layer is employed after the activation function. For effective spatial filtering, the convolution kernel size of the depthwise convolution is set to (C, 1), ensuring coverage across all channels.

2.3.3. Feature fusion convolution block

The block begins with a separable convolution, followed by a sequence of operations, including the BN layer, the ReLU activation function, an average pooling layer, and a dropout layer. Notably, the separable convolution is a composite of depthwise convolution and pointwise convolution. The depthwise convolution autonomously captures the temporal information of each feature mapping, while the subsequent pointwise convolution optimally fuses these feature mappings.

2.3.4. Classification convolution block

For a long time, fully connected layers have served as the conventional structure in CNN classification networks. However, the substantial parameter in fully connected layers often leads to slow training and susceptibility to overfitting. This study proposes a substitution strategy by combining pointwise convolution with global average pooling to replace the fully connected layer. This approach reduces the number of parameters and enhances training efficiency. In contrast to fully connected layers, pointwise convolution efficiently scales down the feature map from the feature fusion convolution block to match the number of categories, facilitating effective information fusion between features. Subsequently, a vector corresponding to N_c is obtained after the global average pooling layer. This innovative approach offers three key advantages:

- **Suppressed model overfitting.** The proposed method alleviates overfitting by scaling down feature maps through pointwise convolution and global average pooling. This approach reduces network parameters, regulating the structure and preventing overfitting, which is exacerbated by the large parameter associated with direct flattened feature maps followed by input to fully connected layers.
- **Enhanced model interpretability.** The vector acquired through global average pooling establishes a more natural and strengthened connection between convolutional output features and categories than the traditional fully connected layer.
- **Increased input size flexibility.** The classification block parameters of the model are no longer related to the size of the EEG signal but only to the final category.

2.4. Network training

We use Pytorch [29] and SpikingJelly [30] frameworks to implement the proposed method. The training process uses an NVIDIA RTXA5000 GPU. The loss function is optimized for the model parameters using an Adam optimizer with an initial learning rate of 0.01 and a cosine annealing learning rate decay strategy. The total number of training epochs for the model is 500.

3. Experiments

This section introduces the utilized dataset, outlines the baseline methods, and details the experimental setup.

3.1. Data description

The datasets used to evaluate the proposed methods are outlined below:

- BCI IV-2a [31]: The BCI IV-2a dataset consists of EEG signals from nine subjects engaging in four different motor imagery tasks: imagining left-hand, right-hand, foot, and tongue movements. Each subject participated in experiments on two separate days, conducting 288 trials per day, with each MI category performed 72 times.
- BCI IV-2b [32]: The BCI IV-2b dataset consists of EEG signals from nine subjects engaging in two different motor imagery tasks: imagining left-hand and right-hand. Each subject participated in feedback and no-feedback experiments, with 240 tasks conducted in the feedback paradigm and 480 in the no-feedback paradigm.

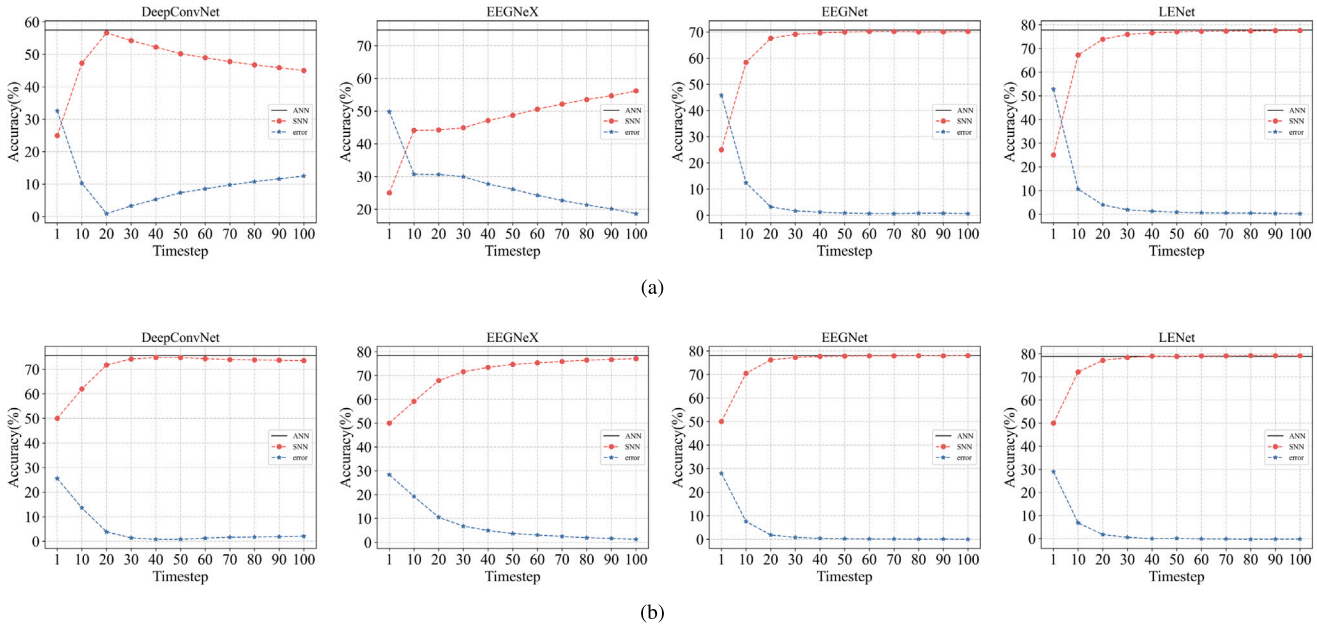


Fig. 3. Performance of baseline methods and LENet after conversion to SNN in relation to time step. Error calculated as $Acc_{ANN} - Acc_{SNN}$. (a) BCI IV-2a. (b) BCI IV-2b.

3.2. Baseline methods

We utilized three commonly used ANN methods for testing to validate our analysis of the errors in the ANN to SNN conversion process. The model parameters are outlined in Table 1. To satisfy the conditions for converting the model from ANN to SNN, we replaced all activation functions in all models with ReLU activation functions.

The baseline methods are outlined below:

- DeepConvNet [33]: DeepConvNet is inspired by architectures in the field of computer vision and has a total of four convolutional pooling layers, with the first layer designed for EEG signals and the following three layers being standard convolutional-pooling layers.
- EEGNeX [34]: EEGNeX is a novel architecture based solely on convolutional neural networks. It represents an improvement over EEGNet in various dimensions, leading to enhanced decoding performance of the model.
- EEGNet [28]: EEGNet is a compact CNN architecture that effectively captures differentiated EEG information, showcasing excellent performance in different BCI tasks. It operates in two dimensions of EEG signals, temporal and channel, and integrates the extracted features.

3.3. Experimental setups

To be fair, we use the same testing strategy when testing the baseline methods. Specifically, we use a 5-fold cross-validation experimental test.

4. Results

This section illustrates the relationship between the performance of the different models after conversion to SNN and the time step. Subsequently, we conduct an analysis of the energy consumption of LENet under both ANN and SNN structures. Finally, we provide a comprehensive analysis of the performance of different models under the ANN structure.

Table 1
The parameter of baseline methods and LENet on BCI IV 2a and 2b datasets.

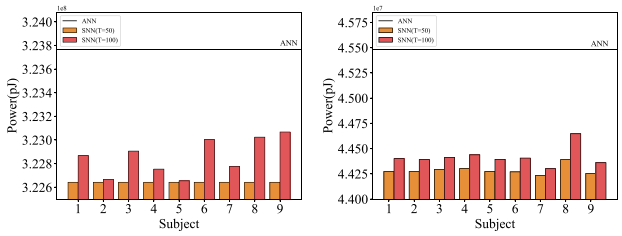
Models	BCI IV-2a	BCI IV-2b
DeepConvNet	282,879	268,202
EEGNeX	48,004	46,290
EEGNet	3444	2146
LENet	3264	2320

4.1. ANN to SNN analysis

We evaluated the performance of ANN to SNN conversion, which focused on evaluating the decoding performance of different models converted to SNN with varying time steps on two MI datasets. Subsequently, we compared the results with the performance metrics observed under the original ANN structure. Fig. 3 visually depicts the performance of different models on these datasets.

BCI IV-2a Theoretical derivations suggest that the conversion error should gradually decrease as the time step increases. Therefore, we analyzed the performance of models with different parameters converted to SNNs when the maximum time step is set to 100. At $T = 100$, the accuracy of LeNet after conversion to SNN is 77.61%, while the accuracy of LeNet in the ANN structure is 77.84%. This small difference indicates a negligible error between the two. Similarly, for the other lightweight model, EEGNet, at $T = 100$, the accuracy after conversion to SNN is 70.18%, compared to 70.78% in the ANN structure, indicating negligible errors. However, for the other two over-parameterized models, DeepConvNet and EEGNeX, their performance errors are more significant than the original ANN when converted to SNN at $T = 100$. The performance error of DeepConvNet reaches 12.50%$(Acc_{ANN} - Acc_{SNN})$, and the performance error of EEGNeX is 18.64%. This shows the importance of designing lightweight and effective models to minimize performance errors introduced during the conversion process.

BCI IV-2b At $T = 100$, the accuracy of LeNet after conversion to SNN is 79.17%, surpassing the accuracy of LeNet under the ANN structure, which is 78.95%. Remarkably, at this point, the performance after conversion to SNN has already exceeded that of the original ANN. Similarly, for the other lightweight model EEGNet, at $T = 100$, the



(a) (b)

Fig. 4. Energy consumption of LENet in both ANN and SNN structures, respectively. (a) BCI IV-2a. (b) BCI IV-2b.

accuracy after conversion to SNN is 78.03%, equivalent to the accuracy of EEGNet under the ANN structure. In contrast, for the other two over-parameterized models, DeepConvNet and EEGNeX, their performance errors concerning the original ANN are more substantial when converted to SNN at $T = 100$. The performance error for DeepConvNet reaches 2.06%, and for EEGNeX, it is 1.28%. In this dataset, it can be inferred that a well-structured model with small parameters can surpass the performance of the original ANN after conversion to SNN.

Observing the results on both datasets, it is evident that the error of most decoding models during the conversion process decreases as the time step increases. This observation validates the presence of a timestep error. Meanwhile, compared to lightweight models (e.g., EEGNet and LENet) for models with many parameters (e.g., DeepConvNet and EEGNeX), the quantization error increases as model parameters.

The results on these two datasets for models with different parameters effectively validate our theoretical derivation of the error from ANN to SNN. This emphasizes the importance of designing a lightweight model for the conversion from ANN to SNN.

4.2. Energy consumption calculation

In addition to the performance evaluation, we also compared the theoretical energy consumption of SNN and ANN by calculating the number of operations during inference [35].

Calculating the theoretical energy cost of the SNN requires first calculating the synaptic operation:

$$SOPs(l) = fr \cdot T \cdot FLOPs(l) \tag{14}$$

where l represents the lth layer in the network, fr denotes the firing rate of the spikes sequence input to the layer, T is the time step of the spiking neural network, and FLOPs denotes the number of floating-point operations, i.e., multiple-and-accumulate (MAC). SOPs are the spike-based accumulate (AC) operations times.

Referring to previous SNN work [36,37], we assume that the MAC and AC operations are realized on 45 nm hardware [38], where $E_{MAC} = 4.6pJ$, $E_{AC} = 0.9pJ$. The theoretical energy consumption of the ANN is calculated as follows:

$$E_{ANN} = E_{MAC} \cdot \sum_{l=1}^{N} FLOPs(l) \tag{15}$$

For SNNs, the energy measurement is relatively complex because the FLOPs before the first neuron are MACs, and the theoretical energy consumption is calculated as follows:

$$E_{SNN} = E_{MAC} \cdot FLOPs(1) + E_{AC} \cdot \sum_{l=2}^{L} SOPs(l) \tag{16}$$

where FLOPs(1) denotes the operation before the first neuron.

We conducted energy consumption tests on LENet under both ANN and SNN structures. The results are shown in Fig. 4. In the case of SNN, energy consumption varies due to different firing rates for each subject. In contrast, the energy consumption of ANN is solely associated with the floating-point operation value of the model, resulting in a constant energy consumption for each subject in the ANN state. Given that the decoding performance of LENet is comparable to ANN at T=50, also analyzed the energy consumption at T=50. As depicted in Fig. 4, the energy consumption of LENet under SNN is notably lower than that under ANN. Additionally, the energy consumption at $T = 50$ is smaller than that at T=100, attributed to the larger number of spikes released by neurons when the time step is greater. Therefore, our model effectively balances the energy consumption and performance within fewer time steps.

Furthermore, we observed that energy consumption is relatively low for subjects with lower decoding accuracy, indicating a lower firing rate of spiking neurons in such cases. Since subjects with low decoding performance may have inactive brain activity, the spiking neuron firing rate can effectively correspond to actual brain activity.

The experimental results suggest that SNN can effectively reduce energy loss in practical BCI applications. Moreover, our proposed LENet is lightweight, making it suitable for deployment on mobile devices with stringent requirements on energy consumption and model parameters.

Table 2
Classification accuracies(%, ± SD), and kappa values for baseline methods and LENet under the ANN structure on BCI IV-2a and BCI IV-2b datasets.

Dataset	Models	Accuracy	Kappa
BCI IV-2a	DeepConvNet	57.52 ± 22.56	0.431
	EEGNeX	74.81 ± 14.69	0.664
	EEGNet	70.77 ± 18.51	0.610
	LENet	**77.84 ± 12.24**	**0.704**
BCI IV-2b	DeepConvNet	75.56 ± 14.23	0.494
	EEGNeX	78.35 ± 13.84	**0.579**
	EEGNet	78.03 ± 14.28	0.549
	LENet	**78.95 ± 13.22**	0.570

4.3. Overall ANN performance comparison

The two datasets' classification results using LENet and baseline methods under ANN structure are shown in Table 2 and Fig. 5. Among all methods, the proposed LENet achieves the best average classification accuracy for all datasets.

Meanwhile, upon meticulous examination of the experimental results on both datasets, it is observed that DeepConvNet does not exhibit an advantage in the MI task, despite having the highest number of parameters. Conversely, LENet, utilizing pointwise convolution and a global average pooling classification layer, achieves superior performance with fewer model parameters. This underscores the effectiveness of LENet in balancing the trade-off between parameters and performance. For both datasets, Fig. 5 shows certain subjects present challenges for classification, yet our proposed method consistently demonstrates high classification performance. This observation indicates that our classification convolution block effectively contributes to the regularization of the model.

To better demonstrate the discrimination of the feature representations extracted by the proposed model, as shown in Fig. 6 we utilized T-distributed Stochastic Neighborhood Embedding (T-SNE) to compare the feature distributions of the extracted by the last hidden layer of the baseline methods and LENet under ANN structure. We can notice that compared with the feature distribution of the baseline methods the features extracted by LENet are more discriminative which effectively verifies the ability of LENet to represent different classes. In particular, this feature representations is more discriminative compared to EEGNet, which is also a lightweight model, and shows that our model can effectively learn different features with fewer parameters.

In supervised learning, when the number of samples is limited, neural networks often face challenges related to overfitting. We assessed the sensitivity of LENet and other baseline models to the number of training samples in two datasets. Specifically, all methods underwent

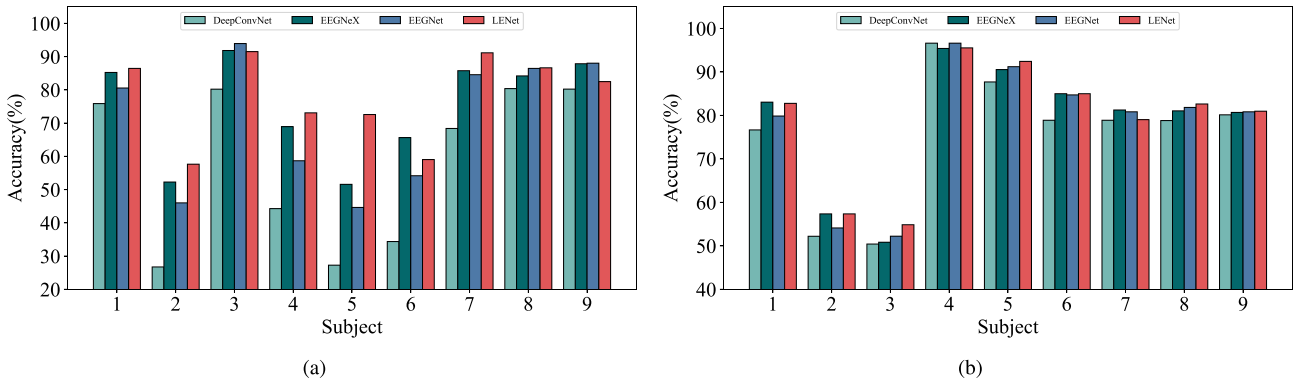


Fig. 5. Subject-specific decoding accuracy of baseline methods and LENet under the ANN structure. (a) BCI IV-2a. (b) BCI IV-2b.

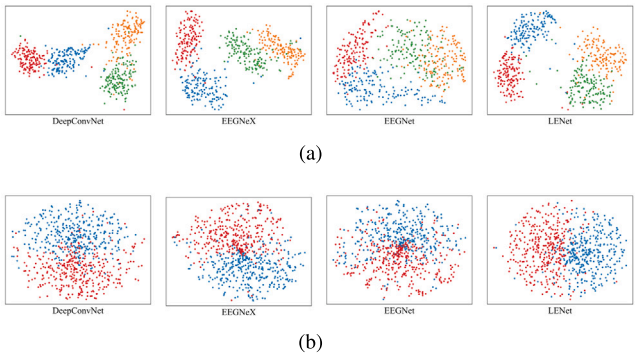


Fig. 6. T-SNE in projected 2-D space visualization of the feature representations of baseline methods and LENet under the ANN structure from one randomly selected subject. Different colors correspond to different categories. ((a) BCI IV-2a. (b) BCI IV-2b.

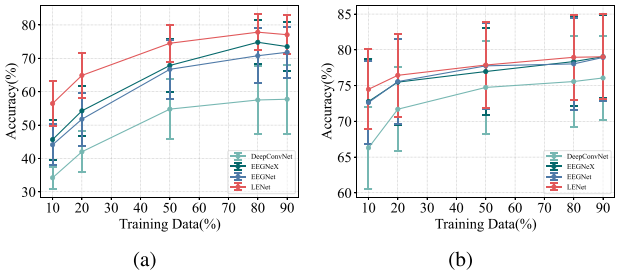


Fig. 7. Performance of baseline methods and LENet under the ANN structure on BCI IV-2a (a) and BCI IV-2b (b) dataset with different numbers of training data. The results show all subjects' classification accuracy as mean ± SEM (standard error of the mean).

validation using 10%, 20%, 50%, 80%, and 90% of the data as the training set, with the remainder designated as the test set. As depicted in Fig. 7, the three baseline methods exhibit high sensitivity to the reduction in sample size compared to LENet. The accuracy of these three baseline models significantly decreases when the training samples are reduced to 20%. In the range between 50% and 90% of the training samples, LENet's model performance shows no significant decline when the number of samples is reduced. Moreover, it is noteworthy that LENet maintains high performance even with a reduction in training data to less than 50%. Consequently, this affirms that LENet can achieve superior classification performance with fewer samples.

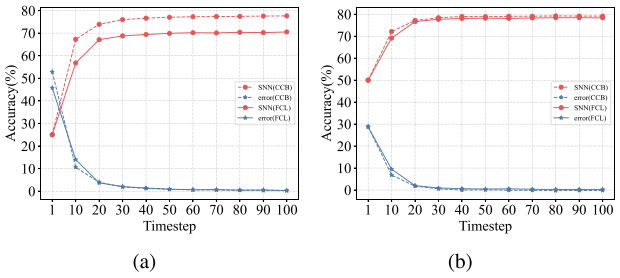


Fig. 8. Performance of LENet utilizing the CCB and the FCL as the classification layer after conversion to SNN in relation to the time step. (a) BCI IV-2a. (b) BCI IV-2b.

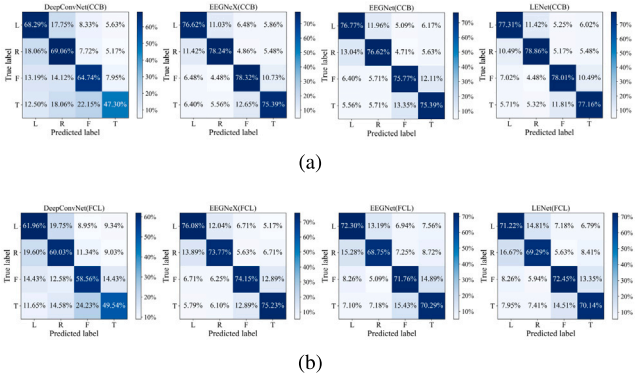


Fig. 9. Confusion matrices of LENet and baseline models, utilizing the CCB (a) and the FCL (b) as the classification layer on the BCI IV-2a dataset.

5. Discussion

5.1. Effects of classification convolution block

Typically, common EEG decoding models incorporate a classification layer that involves flattening the final feature map before feeding it into a fully connected layer (FCL). In contrast, our LENet takes the no-flattened feature map and inputs it into a classification convolution block (CCB) for categorization.

We conducted ablation experiments to analyze the role of LENet's classification convolution block by replacing the CCB in LENet with an FCL. As depicted in Fig. 8, we examined LENet's performance and the relationship between conversion errors and time steps when using CCB and FCL as classification layers during the conversion to SNN.

Table 3
Classification accuracy (%, ± SD), kappa, and parameters when baseline methods and LENet use the fully connected layer and the classification convolution block as the classification output layer under the ANN structure, respectively.

Datasets	Models	Fully connected layer			Classification convolution block		
		Parameters	Accuracy	Kappa	Parameters	Accuracy	Kappa
BCI IV-2a	DeepConvNet	282,879	57.52 ± 22.56	0.431	278,075	**62.35 ± 19.51**	**0.496**
	EEGNeX	48,004	74.81 ± 14.69	0.664	47,040	**77.14 ± 12.33**	**0.695**
	EEGNet	3444	70.77 ± 18.51	0.610	1520	**76.14 ± 13.42**	**0.682**
	LENet	5188	70.78 ± 18.23	0.608	3264	**77.84 ± 12.24**	**0.704**
BCI IV-2b	DeepConvNet	268,202	**75.56 ± 14.23**	0.494	265,800	75.25 ± 12.81	**0.504**
	EEGNeX	46,290	78.35 ± 13.38	0.579	45,808	**78.82 ± 12.72**	**0.581**
	EEGNet	2146	78.03 ± 14.28	0.549	1184	**79.14 ± 12.79**	**0.593**
	LENet	3282	78.65 ± 14.02	**0.576**	2320	**78.95 ± 13.22**	0.570

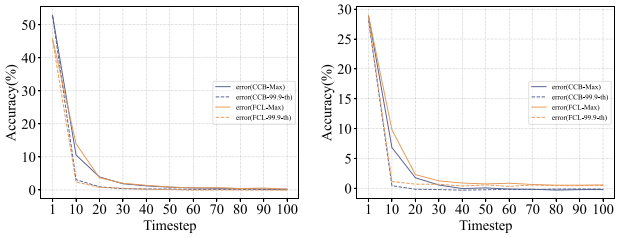


Fig. 10. Effects of scaling factor λ_l on the performance of LENet converted to SNN using CCB and FCL as the classification layer, respectively. (a) BCI IV-2a. (b) BCI IV-2b.

Fig. 8 illustrates that utilizing CCB as the classification layer effectively enhances LENet's performance after conversion to SNN. Notably, when the time step is less than 20, the conversion error is markedly reduced compared to using FCL in LENet. This further substantiates that the lightweight model can effectively minimize errors during the conversion process.

Furthermore, we assessed the performance and parameters of baseline models and LENet under ANN when individually utilizing CCB and FCL as the classification layer. Specifically, we replaced the FCL in DeepConvNet, EEGNeX, and EEGNet with the CCB. As elucidated in Table 3, the CCB not only reduces model parameters compared to FCL but also substantially enhances performance. Results from Fig. 9 demonstrate a significant improvement in outcomes for each category using CCB. These findings demonstrate the effectiveness of the proposed CCB for different models, affirming its plug-and-play characteristics for versatile applicability.

5.2. Effects of scaling factor λ_l

Since the output of each layer of the ANN obeys a specific distribution, there are often large outliers in the output values of the activation function at each layer. Based on the previous theory of ANN to SNN conversion, it is known that this leads to a decrease in the overall neuron firing rate. Rueckauer et al. [25] introduced a method called robust normalization that makes the scaling factor λ_l to be the 99.9-th quantile of the maximum activation value, which increases the robustness to outliers and ensures that a sufficient number of neurons are fired. Here, we discuss the effect of scaling factor λ_l on the performance of the converted LENet.

We conducted separate experiments to assess the relationship between LENet, using the CCB and FCL as the classification layer, and the scaling factor after conversion to SNN. As illustrated in Fig. 10, it is apparent that, irrespective of whether CCB or FCL is employed as the classification layer, utilizing the 99.9-th percentile of activation function values as the scaling factor effectively diminishes conversion errors. Furthermore, for LENet with reduced model parameters after integrating CCB, employing the 99.9-th percentile as the scaling factor proves effective in reducing conversion errors when the time step is less than 20.

Table 4
Classification accuracy (%) of LENet and state-of-the-art methods on two datasets.

Dataset	Models	Type	Year	Accuracy
BCI IV-2a	ShallowConvNet[a] [33]	ANN	2017	66.63
	CP-MixedNet [39]	ANN	2019	74.60
	TS-SEFFNet [40]	ANN	2021	74.61
	CM-ConvNet [41]	ANN	2021	74.90
	SCNet [21]	SNN	2023	67.43
	LENet	ANN	–	**77.84**
	LENet (Max)	SNN	–	77.61
BCI IV-2b	ShallowConvNet[a] [33]	ANN	2017	76.65
	RSMM [42]	ANN	2018	77.96
	3-D-CNN [43]	ANN	2021	75.85
	SMT [44]	ANN	2023	77.36
	2D CNN-LSTM [45]	ANN	2023	75.40
	LENet	ANN	–	78.95
	LENet (Max)	SNN	–	**79.17**

[a] Indicates the result is trained

5.3. Comparison with state-of-the-art methods

To further evaluate the model's performance, we selected a state-of-the-art method for comparison with our proposed method. Several prevalent EEG signal classification methods [21,39–44], both for ANN and SNN, were chosen. As indicated in Table 4, the performance of our LENet under the SNN structure outperforms that of state-of-the-art methods on BCI IV-2a and BCI IV-2b datasets, affirming the superiority of our proposed method.

6. Conclusions

This paper introduces a lightweight and efficient spiking neural network model designed for non-stationary EEG signals on the MI task. Given the inherent challenges in directly training deep SNNs to achieve performance comparable with established ANNs, we opt for the ANN to SNN conversion method. We theoretically derived the ANN to SNN conversion process and subsequently analyzed the two errors intrinsic to the conversion process, which propagate with the model weights. LENet adeptly strikes a balance between parameters and performance, effectively reducing conversion errors in comparison to existing models. Moreover, through energy consumption analysis, we observe a substantial reduction in power consumption for SNN compared to ANN, thereby enhancing the potential applications of SNNs, particularly in scenarios necessitating lightweight and low-power solutions such as on mobile devices.

The advantages inherent in bio-inspired SNNs hold promise for advancing the development of EEG-based BCIs. Concurrently, when

combined with ongoing research progress in ANN, it is anticipated to foster the emergence of novel brain-like SNN models. These models may exhibit potential advantages in accuracy, performance, robustness, generalization, flexibility, and other aspects.

CRediT authorship contribution statement

Xiaojian Liao: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Data curation, Conceptualization. **Guang Li:** Supervision, Formal analysis. **You Wang:** Validation, Investigation. **Lining Sun:** Validation, Supervision. **Hongmiao Zhang:** Writing – review & editing, Validation, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Zahra Emami, Tom Chau, Investigating the effects of visual distractors on the performance of a motor imagery brain-computer interface, Clin. Neurophysiol. 129 (6) (2018) 1268–1275.

[2] Rui Zhang, Qihong Wang, Kai Li, Shenghong He, Si Qin, Zhenghui Feng, Yang Chen, Pingxia Song, Tingyan Yang, Yuandong Zhang, et al., A BCI-based environmental control system for patients with severe spinal cord injuries, IEEE Trans. Biomed. Eng. 64 (8) (2017) 1959–1971.

[3] Kai Keng Ang, Cuntai Guan, EEG-based strategies to detect motor imagery for control and rehabilitation, IEEE Trans. Neural Syst. Rehabil. Eng. 25 (4) (2016) 392–401.

[4] Xiaoxu Li, Wenming Zheng, Yuan Zong, Hongli Chang, Cheng Lu, Attention-based spatio-temporal graphic LSTM for EEG emotion recognition, in: 2021 International Joint Conference on Neural Networks, IJCNN, IEEE, 2021, pp. 1–8.

[5] Jiarong Wang, Luzheng Bi, Weijie Fei, Cuntai Guan, Decoding single-hand and both-hand movement directions from noninvasive neural signals, IEEE Trans. Biomed. Eng. 68 (6) (2020) 1932–1940.

[6] Xiang Liao, Dezhong Yao, Dan Wu, Chaoyi Li, Combining spatial filters for the classification of single-trial EEG in a finger movement task, IEEE Trans. Biomed. Eng. 54 (5) (2007) 821–831.

[7] Hongxin Wang, Huatian Wang, Jiannan Zhao, Cheng Hu, Jigen Peng, Shigang Yue, A time-delay feedback neural network for discriminating small, fast-moving targets in complex dynamic environments, IEEE Trans. Neural Netw. Learn. Syst. 34 (1) (2023) 316–330.

[8] Luyao Zhu, Wei Li, Yong Shi, Kun Guo, SentiVec: learning sentiment-context vector via kernel optimization function for sentiment analysis, IEEE Trans. Neural Netw. Learn. Syst. 32 (6) (2020) 2561–2572.

[9] Hamdi Altaheri, Ghulam Muhammad, Mansour Alsulaiman, Syed Umar Amin, Ghadir Ali Altuwaijri, Wadood Abdul, Mohamed A Bencherif, Mohammed Faisal, Deep learning techniques for classification of electroencephalogram (EEG) motor imagery (MI) signals: A review, Neural Comput. Appl. (2021) 1–42.

[10] Ke Liu, Mingzhao Yang, Zhuliang Yu, Guoyin Wang, Wei Wu, FBMSNet: A filter-bank multi-scale convolutional neural network for EEG-based motor imagery decoding, IEEE Trans. Biomed. Eng. 70 (2) (2022) 436–445.

[11] Hongli Li, Man Ding, Ronghua Zhang, Chunbo Xiu, Motor imagery EEG classification algorithm based on CNN-LSTM feature fusion network, Biomed. Signal Process. Control 72 (2022) 103342.

[12] Wolfgang Maass, Networks of spiking neurons: the third generation of neural network models, Neural Netw. 10 (9) (1997) 1659–1671.

[13] Samanwoy Ghosh-Dastidar, Hojjat Adeli, Spiking neural networks, Int. J. Neural Syst. 19 (04) (2009) 295–308.

[14] Jun Hu, Huajin Tang, Kay Chen Tan, Haizhou Li, How the brain formulates memory: A spatio-temporal model research frontier, IEEE Comput. Intell. Mag. 11 (2) (2016) 56–68.

[15] Michael V DeBole, Brian Taba, Arnon Amir, Filipp Akopyan, Alexander Andreopoulos, William P Risk, Jeff Kusnitz, Carlos Ortega Otero, Tapan K Nayak, Rathinakumar Appuswamy, et al., TrueNorth: Accelerating from zero to 64 million neurons in 10 years, Computer 52 (5) (2019) 20–29.

[16] Eustace Painkras, Luis A Plana, Jim Garside, Steve Temple, Francesco Galluppi, Cameron Patterson, David R Lester, Andrew D Brown, Steve B Furber, SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation, IEEE J. Solid-State Circuits 48 (8) (2013) 1943–1953.

[17] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al., Loihi: A neuromorphic manycore processor with on-chip learning, IEEE Micro 38 (1) (2018) 82–99.

[18] Nikola K. Kasabov, NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, Neural Netw. 52 (2014) 62–76.

[19] Xuanyu Wu, Yixiong Feng, Shanhe Lou, Hao Zheng, Bingtao Hu, Zhaoxi Hong, Jianrong Tan, Improving NeuCube spiking neural network for EEG-based pattern recognition using transfer learning, Neurocomputing 529 (2023) 222–235.

[20] Clarence Tan, Marko Šarlija, Nikola Kasabov, NeuroSense: Short-term emotion recognition and understanding based on spiking neural network modelling of spatio-temporal EEG patterns, Neurocomputing 434 (2021) 137–148.

[21] Xiaojian Liao, Yuli Wu, Zi Wang, Deheng Wang, Hongmiao Zhang, A convolutional spiking neural network with adaptive coding for motor imagery classification, Neurocomputing (2023) 126470.

[22] Peiliang Gong, Pengpai Wang, Yueying Zhou, Daoqiang Zhang, A spiking neural network with adaptive graph convolution and LSTM for EEG-based brain-computer interfaces, IEEE Trans. Neural Syst. Rehabil. Eng. 31 (2023) 1440–1450.

[23] Emre O. Neftci, Hesham Mostafa, Friedemann Zenke, Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks, IEEE Signal Process. Mag. 36 (6) (2019) 51–63.

[24] Xudong Yang, Hongli Yan, Anguo Zhang, Pan Xu, Sio Hang Pan, Mang I Vai, Yueming Gao, Emotion recognition based on multimodal physiological signals using spiking feed-forward neural networks, Biomed. Signal Process. Control 91 (2024) 105921.

[25] Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, Shih-Chii Liu, Conversion of continuous-valued deep networks to efficient event-driven networks for image classification, Front. Neurosci. 11 (2017) 682.

[26] Jianhao Ding, Zhaofei Yu, Yonghong Tian, Tiejun Huang, Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks, 2021, arXiv preprint arXiv:2105.11654.

[27] Yifei Feng, Shijia Geng, Jianjun Chu, Zhaoji Fu, Shenda Hong, Building and training a deep spiking neural network for ECG classification, Biomed. Signal Process. Control 77 (2022) 103749.

[28] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, Brent J Lance, EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces, J. Neural Eng. 15 (5) (2018) 056013.

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Adv. Neural Inform. Process. Syst. 32 (2019).

[30] Wei Fang, Yanqi Chen, Jianhao Ding, Zhaofei Yu, Timothée Masquelier, Ding Chen, Liwei Huang, Huihui Zhou, Guoqi Li, Yonghong Tian, SpikingJelly: An open-source machine learning infrastructure platform for spike-based intelligence, Sci. Adv. 9 (40) (2023) eadi1480.

[31] Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, Gert Pfurtscheller, BCI competition 2008–Graz data set A, Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology 16 (2008) 1–6.

[32] R Leeb, C Brunner, G Müller-Putz, A Schlögl, GJGUOT Pfurtscheller, BCI competition 2008–Graz data set B, Graz University of Technology, Austria 16 (2008) 1–6.

[33] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, Tonio Ball, Deep learning with convolutional neural networks for EEG decoding and visualization, Human brain mapping 38 (11) (2017) 5391–5420.

[34] Xia Chen, Xiangbin Teng, Han Chen, Yafeng Pan, Philipp Geyer, Toward reliable signals decoding for electroencephalogram: A benchmark study to EEGNeX, Biomed. Signal Process. Control 87 (2024) 105475.

[35] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, YAN Shuicheng, Yonghong Tian, Li Yuan, Spikformer: When spiking neural network meets transformer, in: The Eleventh International Conference on Learning Representations, 2022.

[36] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, Guoqi Li, Attention spiking neural networks, IEEE Trans. Pattern Anal. Mach. Intell. 45 (8) (2023) 9393–9410.

[37] Bojian Yin, Federico Corradi, Sander M. Bohté, Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks, Nat. Mach. Intell. 3 (10) (2021) 905–913.

[38] Mark Horowitz, 1.1 Computing's energy problem (and what we can do about it), in: 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers, ISSCC, IEEE, 2014, pp. 10–14.

[39] Yang Li, Xian-Rui Zhang, Bin Zhang, Meng-Ying Lei, Wei-Gang Cui, Yu-Zhu Guo, A channel-projection mixed-scale convolutional neural network for motor imagery EEG decoding, IEEE Trans. Neural Syst. Rehabil. Eng. 27 (6) (2019) 1170–1180.

[40] Yang Li, Lianghui Guo, Yu Liu, Jingyu Liu, Fangang Meng, A temporal-spectral-based squeeze-and-excitation feature fusion network for motor imagery EEG decoding, IEEE Trans. Neural Syst. Rehabil. Eng. 29 (2021) 1534–1545.

[41] Weifeng Ma, Yifei Gong, Gongxue Zhou, Yang Liu, Lei Zhang, Boxian He, A channel-mixing convolutional neural network for motor imagery EEG decoding and feature visualization, Biomed. Signal Process. Control 70 (2021) 103021.

[42] Qingqing Zheng, Fengyuan Zhu, Pheng-Ann Heng, Robust support matrix machine for single trial EEG classification, IEEE Trans. Neural Syst. Rehabil. Eng. 26 (3) (2018) 551–562.

[43] Ji-Seon Bang, Min-Ho Lee, Siamac Fazli, Cuntai Guan, Seong-Whan Lee, Spatio-spectral feature representation for motor imagery classification using convolutional neural networks, IEEE Trans. Neural Netw. Learn. Syst. 33 (7) (2021) 3038–3049.

[44] Jing Luo, Yaojie Wang, Shuxiang Xia, Na Lu, Xiaoyong Ren, Zhenghao Shi, Xinhong Hei, A shallow mirror transformer for subject-independent motor imagery BCI, Comput. Biol. Med. (2023) 107254.

[45] Jialing Wang, Shiwei Cheng, Jieming Tian, Yuefan Gao, A 2D cnn-lstm hybrid algorithm using time series segments of EEG data for motor imagery classification, Biomed. Signal Process. Control 83 (2023) 104627.