Expert Systems With Applications 187 (2022) 115968

Contents lists available at ScienceDirect

Expert Systems With Applications

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

EEG-based motor imagery classification using convolutional neural networks with local reparameterization trick

Wenqie Huang [a], Wenwen Chang [a], Guanghui Yan [a,*], Zhifei Yang [a], Hao Luo [a,b], Huayan Pei [a]

[a] School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070 China
[b] School of Information Science and Engineering, Gansu University of Chinese Medicine, Lanzhou 730000 China

ARTICLE INFO

Keywords:
Electroencephalogram (EEG)
Motor Imagery (MI)
Deep learning (DL)
Convolutional neural networks (CNNs)
Local Reparameterization Trick
Classification

ABSTRACT

Objectives: Deep learning (DL) method has emerged as a powerful tool in studying the behavior of Electroencephalogram (EEG)-based motor imagery (MI). Although prospective studies have demonstrated promising performance, most of these studies have been affected by the lack of research between groups and individual subjects, and the accuracy of MI classification still has room for improvement. Due to the inter-individual variability in the EEG classification, enhancing the adaptability and robustness between different individuals is especially critical.

Methods & experiments: We developed a novel DL model based on the EEG signals to improve MI classification performance by introducing the local reparameterization trick into convolutional neural networks (LRT-CNN). 109 subjects from PhysioNet Dataset were used to test the proposed model. Firstly, a global classifier was evaluated by four groups. Secondly, individual variability was examined by testing individual subjects.

Results: The classification accuracy of global classifier in 20 subjects, 50 subjects, 80 subjects, and 109 subjects are 93.86%, 98.94%, 93.04%, and 92.41%, respectively. The maximum classification accuracy of one individual subject is 99.79%, which is better than the state-of-the-art method and proves the proposed method can handle the challenge of individual variability.

Conclusion: We conclude that introducing the local reparameterization trick into convolutional neural networks can significantly improve the accuracy of the MI tasks based on the EEG signals without any complicated and tedious feature engineering works. Besides, encouraging results were obtained both between groups (multiple subjects) and on a single subject.

Implication: The experimental results add to the rapidly expanding field of brain science and contribute to our understanding of applying the DL method to address EEG-based classification problems (not limited to MI classification issues).

1. Introduction

Brain-computer interface (BCI) systems have recently attracted much more attention in the medical field (Santhanam, Ryu, Yu, Afshar, & Shenoy, 2006). BCI is an innovative technology that bypasses the human's normal peripheral-nerve pathways and controls external devices directly. Hence, the BCI systems can decode brain activity patterns to help patients manipulate auxiliary equipment, recovering their motor functions due to stroke or spinal cord injury (Koutsou, Summa, Nasser, Martinez, & Thangaramanujam, 2014). There are two types of BCI: invasive BCI and non-invasive BCI (Sitaram et al., 2007). For invasive BCI (such as electrocorticography, ECoG), electrodes are implanted into the cortex through surgery to directly obtain signals in the brain parenchyma, acquiring the highest-quality EEG signals. But due to the rejection of the body and the high risk of neurosurgery, invasive BCI is mainly aimed at blind and paralyzed patients. For non-invasive BCI (such as electroencephalogram, EEG), we can measure EEG signals from the brain by directly measuring the scalp electrodes (not requires surgery). The most critical advantages of EEG are high temporal resolution and its portability, but the disadvantage is the low spatial resolution. Compared with the weakness, many researchers prefer the merits of EEG because of its characteristics (such as accessibility, etc.).

* Corresponding author.
E-mail addresses: huangwenqie@126.com (W. Huang), changww2013@126.com (W. Chang), yanguanghui@mail.lzjtu.cn (G. Yan), yzf@mail.lzjtu.cn (Z. Yang), luoh804@gszy.edu.cn (H. Luo), pei123com@126.com (H. Pei).

https://doi.org/10.1016/j.eswa.2021.115968
Received 31 January 2021; Received in revised form 16 July 2021; Accepted 21 September 2021
Available online 27 September 2021

Due to the higher temporal resolution, portability, low hardware cost, and non-invasiveness, EEG-based BCI are among the most widely used in BCI applications, thus driving many researchers to study BCI based on EEG to control external equipment (LaFleur et al., 2013; Liao et al., 2012), such as wheelchairs (Zhang et al., 2016), robot arms (Silver, Ress, & Heeger, 2005; Wang, Dong, Chen, & Shi, 2015), etc. Based on EEG signals, researchers have investigated the study of MI. MI can be understood as people imagine the movement without actual movement, and certain brain areas are still active so that we can record the corresponding EEG signal through imagination. So far, MI-based BCI systems (MI-BCI) have been widely applied in the field of rehabilitation engineering. According to the studies (LaFleur et al., 2013; Liao et al., 2012; Silver et al., 2005; Zhang et al., 2016), most researchers indicate that MI-BCI could be of high utility in rehabilitating limb diseases, which has prompted many researchers have invested in MI-BCI research.

Several research pieces (Aghaei, Mahanta, & Plataniotis, 2016; Ang, Chin, Zhang, & Guan, 2008; Bentlemsan, Zemouri, Bouchaffra, Yahya-Zoubir, & Ferroudji, 2014; Grosse-Wentrup & Buss, 2008; Kim et al., 2016; Park, Took, & Mandic, 2014; Wang, Gao, & Gao, 2005) have investigated that specialized preprocessing was used for EEG signals to extract relevant features; next, supervised machine learning (such as support vector machine) was used to classify the features. However, the primary limitation of those approaches was that we need to spend lots of time and energy to compute a large number of features. For instance, Grosse-Wentrup and Buss (2008) and Wang et al. (2005) proposed the Common Spatial Patterns (CSP) algorithm that requires extracting CSP features before the classification phase leads to an enormous workload. Therefore, it is necessary to consider new methods that can directly complete the feature extraction and classification process from the EEG time series, which can speed up the processing work and save computing resources.

Recently, researchers have introduced deep learning methods to the EEG classification task, as it can help deliver improved accuracy and reduce the number of EEG channels required. Yang, Sakhavi, Ang, and Guan (2015) proposed a frequency complementary feature map selection (FCMS) method to extract EEG signal features and explore MI classification based on convolutional neural networks. Kumar, Sharma, Mamun, and Tsunoda (2016) presented a deep learning method that used a common spatial pattern (CSP) to extract features as the input of the deep neural network (DNN) to complete the MI tasks. Bashivan, Rish, Yeasin, and Codella (2016) focused on the spatial, spectral, and temporal structure of EEG signals and evaluated the cognitive load classification task by utilizing a deep recurrent-convolutional network. Lu, Li, Ren, and Miao (2017) applied the fast Fourier transform (FFT) and wavelet package decomposition (WPD) to obtain frequency-domain representation and then performed MI classification based on the restricted Boltzmann machine (RBM). Jiao, Gao, Wang, Li, and Xu (2018) proposed improved convolutional neural network frameworks that contained both single-model and double-model methods. Considering the spatial, spectral, and temporal information of EEG data, the proposed method's performance was improved compared with state-of-the-art studies. Hou, Zhou, Jia, and Lun (2020) proposed an innovative approach that combines the Scout EEG Source Imaging and CNNs to decode EEG signals, which obtained competitive performance on PhysioNet Dataset (Goldberger et al., 2000).

However, these methods mentioned above require complex preprocessing procedures (such as feature map extraction), which takes much more time for an online BCI system. So far, there has little research on utilizing DL methods to deal with EEG signals without any feature engineering work in EEG-based classification tasks. Shen, Lu, and Jia (2017) combined recurrent neural networks (RNNs) and convolutional neural networks (CNNs). Besides, they also employed the stacked random forest model to enhance feature extraction and classification capabilities. Dose, Møller, Iversen, and Puthusserypady (2018) built an end-to-end model to learn generalized features and dimension reduction without needing a specific method to extract EEG signal features. On this basis, continue to introduce transfer learning method, the classification accuracy has been improved. Sun, Lo, and Lo (2019) proposed an architecture that combined the Convolutional Neural Network and Long Short-term Memory Neural Network to extract spatiotemporal features and classify EEG data, achieving very high accuracy with benchmark methods.

Although prospective studies have shown encouraging performance, the accuracy of EEG-based classification still has room for improvement with the emergency of some new algorithms. Most of these studies have been affected by the lack of research between groups and individual subjects. Due to the inter-individual variability in the EEG classification, enhancing the adaptability and robustness between different individuals is especially critical. Significantly, the following two issues will be addressed: (1) Experiment between groups (such as 20 subjects as a group) on MI classification tasks. (2) Single-subject (such as one individual subject signed as S1) experiment on MI classification tasks.

This study aimed to propose a deep learning architecture for EEG-based classification. Our primary hypothesis was to introduce the local reparameterization trick into convolutional neural networks to directly process the EEG signals. An alternative estimator with higher computational efficiency is proposed in the local reparameterization trick, which can generate a (statistically) effective gradient estimator. This gradient estimator can effectively reduce the variance on the profile, thereby making the classification accuracy higher and converging faster. To test this hypothesis, we applied the local reparameterization trick for each convolutional layer and fully connected layer during the training phase. Further, we used empirical data from PhysioNet Dataset to analyze the issues between a group of subjects and a single subject. The experimental results have illustrated that the proposed method improves the EEG classification accuracy compared with the state-of-the-art works.

The rest of this paper is organized as follows: the definition and foundation of mathematics were introduced in Section 2. Section 3 presents the proposed model and implementation details. Section 4 shows the results of experiments with the proposed model. Discussion on the comparison with related work and the performance of the proposed model are provided in Section 5. Finally, Section 6 gives the conclusions of this article. As shown in Fig. 1, we display the schematic diagram of the MI classification system.

2. Background

2.1. Variational inference

Variational inference (VI) is an essential deterministic approximate inference method (Jordan, Ghahramani, Jaakkola, & Saul, 1999; Wainwright & Jordan, 2008), which computes the approximate expectation of random variables through variational iterations. In deep learning applications, the underlying layer of the well-known variational autoencoders is based on variational reasoning.

First, based on the model assumption $p(x,z)$, which includes latent variables $z = z_{1:m}$ and observations $x = x_{1:n}$, posterior probability $p(z|x)$ can be calculated through the approximate reasoning by Bayesian model,

$$p(x)p(z|x) = p(x,z), p(x) = \int \mathrm{p(x,z)dz} \tag{1}$$

Here, $p(x)$ is often impossible to calculate directly, so $p(z|x)$ cannot be calculated. Therefore, a convenient distribution $q(z)$ (we always make z and x independent when we pick q) is used to approximately represent the exact posterior probability $p(z|x)$. Take the log of both sides of Eq. (1),

$$ln p(x) = ln\frac{\mathrm{p(x,z)}}{\mathrm{q(z)}} - ln\frac{\mathrm{p(z|x)}}{\mathrm{q(z)}} = ln\mathrm{p(x,z)} - ln\mathrm{q(z)} - ln\frac{\mathrm{p(z|x)}}{\mathrm{q(z)}} \tag{2}$$

Take the expectation of $q(z)$ on both sides of Eq. (2) ,

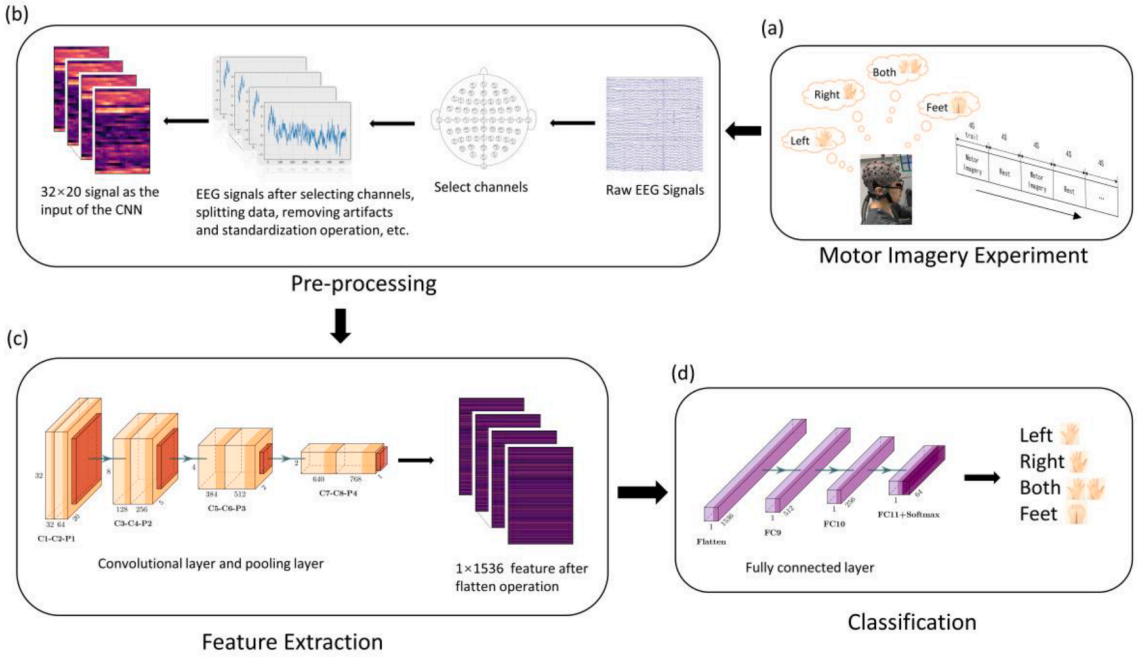


Fig. 1. Schematic diagram of the MI classification system. Firstly, as shown in (a), a participant performs the MI experiment, and the EEG signals will be recorded. Secondly, as shown in (b), we will use the raw EEG signal for preprocessing. Herein, we select channels from all channels (such as 64 channels) for the sensor and motor area. After selecting the channel, splitting data, removing artifacts, and standardization operation, etc., we convert the 1×640 signal into the 32×20 dimension as the CNN model's input. Thirdly, as shown in (c), the EEG signals will be extracted with valuable information for classification. Finally, as shown in (d), the EEG signals will be classified into four categories in this classification phase.

$$lnp(x) = \underbrace{\int q(z)lnp(x,z)dz - \int q(z)lnq(z)dz}_{ELBO} + \underbrace{\left[-\int q(z)ln\frac{p(z|x)}{q(z)}dz\right]}_{KL-divergence} \tag{3}$$

The first part is called the evidence lower bound (ELBO) (Blei, Kucukelbir, & McAuliffe, 2017), and the second is KL-divergence. Then the goal is to update $q(z)$ by adjusting the parameter z to minimize KL (It is minimized when $q(\cdot) = p(\cdot)$). In fact, minimizing the KL is the same thing as maximizing the ELBO.

On the other hand, Jensen's inequality is used for $lnp(x)$ (Jordan et al., 1999),

$$\begin{aligned} lnp(x) &= ln\int \mathrm{p(x,z)}\frac{\mathrm{q(z)}}{\mathrm{q(z)}}\mathrm{dz} = ln\left(\mathrm{E_q}\left[\frac{\mathrm{p(x,z)}}{\mathrm{q(z)}}\right]\right) \\ &\geqslant E_q\left(ln\left[\frac{p(x,z)}{q(z)}\right]\right) = \mathrm{E_q}(lnp(x,z)) - \mathrm{E_q}(lnq(z)) \\ &\triangleq \mathscr{L}(q) \end{aligned} \tag{4}$$

For the $q(z)$ with $z = z_{1:m}$, here we assume that z_i is independent from $z_j (i \neq j, i, j = 1, \cdots, m)$, the following result is obtained by using Mean-field theory,

$$q(z) = \prod_{i=1}^{m} q_i(z_i) \tag{5}$$

$$\begin{aligned} \mathscr{L}(q) &= \int q(z)ln\frac{p(x,z)}{q(z)}dz \\ &= \int \prod_{i=1}^{m} q_i(z_i)lnp(x,z)dz - \int \prod_{i=1}^{m} q_i(z_i)lnq(z)dz = \int_{z_j} q_j(z_j)E_{i\neq j}[lnp(x,z)]dz_j - \int_{z_j} q_j(z_j)lnq_j(z_j)dz_j + const \end{aligned} \tag{6}$$

Here we introduce the symbol$\widetilde{p}$, $E_{i\neq j}[lnp(x,z)] = ln\left(\widetilde{p}_j(x,z_j)\right)$, then

$$\mathscr{L}(q_j) = \int_{z_j} q_j(z_j) ln \frac{\widetilde{p}_j(x,z_j)}{q_j(z_j)} dz_j + const \tag{7}$$

So each iteration of q follows the following formula,

$$lnq_j^*(z_j) = E_{i\neq j}[lnp(x,z)] \tag{8}$$

However, since the hypothesis is too strong, and the integral of$E_{i\neq j}[lnp(x,z)]$ is intractable, it is urgent to find a more efficient method.

2.2. Stochastic gradient variational Bayes (SGVB)

In general, the estimation of the generated model is difficult to calculate with the traditional gradient descent method. In order to solve the problem, the Stochastic Gradient Variational Bayes (SGVB) estimator, an effective method, was proposed by Kingma and Welling (2013), and the computational difficulty of VI is transformed into an optimization problem.

To make it clear, we use the symbol $X = \{x^{(i)}\}_{i=1}^N$ to represent datasets of the random variable x. Then we write $q(z)$ in terms of $q_\phi(z)$ (shorthand for q_ϕ) and $p(x)$ in terms of $p_\theta(x)$, here ϕ and θ are parameters. Eq. (3) and Eq. (4) can be rewritten as follows:

$$ln[p_\theta(x^{(i)})] = \mathscr{L}(\phi) + KL(q||p) \geqslant \mathscr{L}(\phi) \tag{9}$$

$$\mathscr{L}(\phi) = E_{q_\phi}\{ln[p_\theta(x^{(i)},z)] - ln[q_\phi(z)]\} \tag{10}$$

We consider the gradient of $\mathscr{L}(\phi)$ to find $\widehat{\phi}$ that maximizes $\mathscr{L}(\phi)$:

$$\begin{aligned}
\nabla_\phi \mathscr{L}(\phi) &= \nabla_\phi E_{q_\phi}[lnp_\theta(x^{(i)},z) - lnq_\phi] \\
&= \nabla_\phi \int q_\phi[lnp_\theta(x^{(i)},z) - lnq_\phi]dz \\
&= \int \nabla_\phi q_\phi[lnp_\theta(x^{(i)},z) - lnq_\phi]dz \\
&= \int q_\phi \nabla_\phi lnq_\phi[lnp_\theta(x^{(i)},z) - lnq_\phi]dz \\
&= E_{q_\phi}[\nabla_\phi lnq_\phi(lnp_\theta(x^{(i)},z) - lnq_\phi)]
\end{aligned} \tag{11}$$

Then the Monte Carlo method is used to approximate the expectation. Here rewrite the $\nabla_\phi \mathscr{L}(\phi)$ as follows: $z^{(t)} \sim q_\phi(z), t = 1,\cdots,T$,

$$\nabla_\phi \mathscr{L}(\phi) \approx \frac{1}{T}\sum_{t=1}^{T} \nabla_\phi lnq_\phi(z^{(t)})(lnp_\theta(x^{(i)},z^{(t)}) - lnq_\phi(z^{(t)})) \tag{12}$$

However, it is possible that sampling $z^{(t)}$ will cause a high variance, and that must be large enough of T to get a good approximation. Considering this situation, the SGVB method has utilized to reparameterize the random variable $z^{(t)} \sim q_\phi(z)$:

$$\widetilde{z} = g_\phi(\varepsilon, x) \ with \ \varepsilon \sim p(\varepsilon) \tag{13}$$

where the transformation $g_\phi(\varepsilon,x)$ is differentiable and $\varepsilon \sim p(\varepsilon)$ is an auxiliary noise variable. Then use Monte Carlo estimates of expectations to the variational lower bound with the function $f(z)$ w.r.t. $q_\phi(z|x)$:

$$\nabla_\phi E_{q_\phi(z)}[f(z)] = E_{q_\phi(z)}\left[f(z)\nabla_{q_\phi(z)} lnq_\phi(z)\right] \simeq \frac{1}{T}\sum_{t=1}^{T} f(z)\nabla_{q_\phi(z^{(t)})} lnq_\phi(z^{(t)}) \tag{14}$$

$$E_{q_\phi(z|x^{(i)})}[f(z)] \simeq \frac{1}{T}\sum_{t=1}^{T} f(g_\phi(\varepsilon^{(t)},x^{(i)})) \tag{15}$$

yielding the generic Stochastic Gradient Variational Bayes (SGVB) estimator:

$$\widetilde{\mathscr{L}}^A(\theta,\phi;x^{(i)}) \simeq \mathscr{L}(\theta,\phi;x^{(i)}) \tag{16}$$

Here is

$$\mathscr{L}(\theta,\phi;x^{(i)}) = -KL(q_\phi(z|x^{(i)})||p_\theta(z)) + E_{q_\theta}[lnp_\theta(x^{(i)}|z)] \tag{17}$$

$$\widetilde{\mathscr{L}}^A(\theta,\phi;x^{(i)}) = -KL(q_\phi(z|x^{(i)})||p_\theta(z)) + \frac{1}{T}\sum_{t=1}^{T} lnp_\theta(x^{(i)}|z^{(i,t)}) \tag{18}$$

where

$$z^{(i,t)} = g_\phi(\varepsilon^{(i,t)},x^{(i)}) \text{ and } \varepsilon^{(t)} \sim p(\varepsilon) \tag{19}$$

And the expected likelihood can be formed:

$$E_{q_\theta(z)} lnp(x) \simeq L^{SGVB}(\phi) \tag{20}$$

where $X_M = \{x^{(1)},\cdots,x^{(M)}\}$ is a randomly drawn sample of M datapoints from the dataset X, and$L^{SGVB}(\phi) = \frac{T}{M}\sum_{i=1}^{M} lnp_\theta(x^{(i)}|z^{(i,t)})$, here the gradient is unbiased:

$$\nabla_\phi E_{q_\phi(z|x^{(i)})}[f(z)] \simeq \nabla_\phi\left[\frac{1}{T}\sum_{t=1}^{T} f(g_\phi(\varepsilon^{(t)},x^{(i)}))\right] \tag{21}$$

2.3. Local reparameterization trick

The problem of high variance in SGVB is caused by the randomness of samples $z^{(t)}(z^{(t)} \sim q_\phi(z))$, so it is often to express the variable z as a differentiable variable $z = g_\phi(\varepsilon,x)$, which is parameterized by ϕ, and $\varepsilon(\varepsilon \sim p(\varepsilon))$ is an auxiliary noise variable. Then we knew that $|q_\phi(z|x^{(i)})dz| = |p(\varepsilon)d\varepsilon|$,

$$\begin{aligned}
\nabla_\phi \mathscr{L}(\phi) &= \nabla_\phi \int [lnp_\theta(x^{(i)},z) - lnq_\phi] q_\phi dz \\
&= \nabla_\phi \int [lnp_\theta(x^{(i)},z) - lnq_\phi] p(\varepsilon) d\varepsilon \\
&= E_{p(\varepsilon)}[\nabla_\phi(lnp_\theta(x^{(i)},z) - lnq_\phi(z|x^{(i)}))\cdot\nabla_\phi z] \\
&= E_{p(\varepsilon)}[\nabla_\phi(lnp_\theta(x^{(i)},z) - lnq_\phi(z|x^{(i)}))\cdot\nabla_\phi g_\phi(\varepsilon,x^{(i)})]
\end{aligned} \tag{22}$$

and it simplifies the problem.

As described in the previous section, an alternative estimator was proposed by Kingma, Salimans, and Welling (2015) while $Cov[L_i,L_j] = 0$, herein, $p\left(x^{(i)},g_\phi(\varepsilon,x^{(i)})\right)$ was expressed as L_i so that the variance scale of the random gradient is $1/M$. Then the new estimator is made computationally efficient by sampling the intermediate variables $g(\varepsilon)$ through which ε influences $L^{SGVB}(\phi)$. Then the source of global noise can be translated to local noise. The local reparameterization can be used to yield a statistically efficient gradient estimator, which can reduce the high variance generated in section 2.2. This type of reparameterization is known as the local reparameterization trick.

This paper still considers a standard fully connected neural network with a hidden layer consisting of 640 neurons. The hidden layer receives an input feature matrix X ($M \times 640$) from the layer below, multiplied by a 640×640 weight matrix W, before neuron activations, i.e., A = XW. Then we specify the posterior approximation on the weights to be a fully factorized Gaussian, i.e.,$q_\phi(w_{i,j}) = N\left(\mu_{i,j},\sigma_{i,j}^2\right)$, $\forall w_{i,j} \in W$, where the weights are sampled as$w_{i,j} = \mu_{i,j} + \sigma_{i,j}\varepsilon_{i,j}$, with $\varepsilon_{i,j} \sim N(0,1)$. Then we need to sample a large random number for just a single layer of the convolution neural network. The calculation following this step is more complicated: the form of the matrix–matrix product of the form A = XW turns into M separate local vector–matrix products. This kind of computation introduces higher complexity and is not easy to perform. And the optimization library for convolution cannot handle a separate filter matrix for each example.

We sample the neuron activations A directly to obtain an efficient

Monte Carlo estimator in an easier way. And the posterior for the activations A is factorized Gaussian:

$$q_\phi(a_{m,j}|\mathrm{X}) = N(\gamma_{m,j}, \delta_{m,j}),\ \text{with}\ \gamma_{m,j} = \sum_{i=1}^{640} x_{m,i}\mu_{i,j},\ \text{and}\ \delta_{m,j} = \sum_{i=1}^{640} x_{m,i}^2\sigma_{i,j}^2 \tag{23}$$

$$\text{Here}, a_{m,j} = \gamma_{m,j} + \sqrt{\delta_{m,j}}\varsigma_{m,j},\ \text{with}\ \varsigma_{m,j} \sim N(0,1) \tag{24}$$

Then the number of samples required has been greatly reduced and more computationally efficient. Then we form the stochastic gradient estimate of the parameter $\sigma_{i,j}^2$ for a small batch of size $M = 1$ using the reparameterization technique:

$$\frac{\partial L^{SGVB}}{\partial \sigma_{i,j}^2} = \frac{\partial L^{SGVB}}{\partial a_{m,j}} \cdot \frac{\partial a_{m,j}}{\partial \delta_{m,j}} \cdot \frac{\partial \delta_{m,j}}{\partial \sigma_{i,j}^2} = \frac{\partial L^{SGVB}}{\partial a_{m,j}} \cdot \frac{\varsigma_{m,j} x_{m,i}^2}{2\sqrt{\delta_{m,j}}} \tag{25}$$

Then the local reparameterization trick leads to an estimator that has lower variance. And the expectation we want is based on a standard normal distribution, which makes sample work very easy.

2.4. Convolutional neural networks

The convolutional neural network (CNN) is a deep neural network that contains convolution operations and achieves parameter sharing in neural network training (Gu et al., 2018). It mainly consists of convolutional, pooling, and fully connected layers. Mathematically, we define some formulas and notations that we will analyze next. The feature value of l^{th} layer,$z^{[l]}$ is computed as:

$$z^{[l]} = w^{[l]}a^{[l]} + b^{[l]} \tag{26}$$

where $w^{[l]}$ is the weight of the l^{th} layer, $a^{[l]}$ is the input of the l^{th} layer (note that $a^{[0]} = x_{input}$), and $b^{[l]}$ is the bias of the l^{th} layer. It should be noted that $w^{[l]}$ is the global sharing hyperparameter. We can utilize the same parameters in different positions in input data to detect temporal information or other features. Even if the upper left corner and the lower right corner of the input data have different distributions or maybe seems different, but they might be similar enough, the entire data shares the feature detector, and the extraction effect is also acceptable. The activation function is designed to add nonlinear factors to the neural network to solve more complex problems better. The activation value $a^{[l]}$ can be calculated by:

$$a^{[l]} = g(z^{[l]}) \tag{27}$$

where $g(\cdot)$ denotes the activation function,$z^{[l]}$ is the feature value of the l^{th} layer. The neural network's commonly used activation functions are sigmoid, tanh (LeCun, Bottou, Orr, & Müller, 2012), ReLU (Nair & Hinton, 2010), ELU (Clevert, Unterthiner, & Hochreiter, 2016), etc. Note that the activation function is differentiable almost everywhere (Gulcehre, Moczulski, Denil, & Bengio, 2016).

In addition to the convolutional layer, CNN often uses a pooling layer to reduce the model's representation size, increase the calculation speed, and improve the extracted features' robustness. Denoting the value after pooling operation as $p^{[l]}$, it can be defined as:

$$p^{[l]} = pooling(a^{[l]}) \tag{28}$$

where $pooling(\cdot)$ denotes the pooling operation and $a^{[l]}$ is the feature after activation operation. There are two types of pooling operations: Avg pooling (Wang, Wu, Coates, & Ng, 2012) and Max pooling (Boureau, Ponce, & Lecun, 2010). Max pooling is the most commonly used in the neural network because it's been found to works well in many experiments. The actual effect of the Max pooling operation is if the feature is extracted by the filter and then keep a maximum value. If this feature is not extracted, then this feature does not exist or is very small in the filter window. The impressive property is that Max operation has a hyperparameters set, but there are no parameters to learn.

Other than the convolutional layer and the pooling layer, most neural network architectures still add a fully connected layer. The fully connected layer integrates all the previous feature representations and usually has several consecutive layers to avoid losing the required feature information. Hence, the fully connected layer's function is to classify useful information, and finally, to separate specific categories (such as four categories) through the softmax operation.

Take the four classification tasks as an example. Softmax maps the output of multiple neurons to the interval (0,1), which gives four values between zero and one. Therefore, which position (four indexes in total) of the four values has the largest value corresponds to this position's category. Mathematically, the softmax value of i^{th} class can be calculated by:

$$Softmax(\widehat{y}_i) = \frac{e^{\widehat{y}_i}}{\sum_j^n e^{\widehat{y}_j}} \tag{29}$$

where $\widehat{y}_i$ denotes the predicted value of i^{th} class after softmax operation, n indicates the numbers of category. In this study, n equals four because we conducted experiments on the four-category problem. So $\widehat{y}_i$ has four values,$\widehat{y}_1$,$\widehat{y}_2$,$\widehat{y}_3$, and$\widehat{y}_4$, respectively. The optimal parameters can be obtained by minimizing the cost function so that the predicted value $\widehat{y}_i$ can be closer and closer to the real value y_i. The cost function can be defined as:

$$J = \frac{1}{m}\sum_{i=1}^{m} \mathscr{L}(\widehat{\mathrm{y}}_\mathrm{i}, \mathrm{y}_\mathrm{i}) \tag{30}$$

where J is the cost value of the loss function, m is the training set size, and $\mathscr{L}(\cdot)$ denotes the loss function that calculates the loss between $\widehat{y}_i$ and y_i. Therefore, the purpose of training the neural network is to optimize all the parameters to reduce the value of cost J by using stochastic gradient descent (SGD) (Wijnhoven & de With, 2010; Zinkevich, Weimer, Li, & Smola, 2010)or other algorithms (such as RMSProp).

3. Method

The proposed model's framework diagram was shown in Fig. 2. More implementation details about the framework were shown in Table 1. In this study, we chose the 32×20 dimension as input because the difference between the height and width of 40×16 or 64×10 will significantly affect the convolution operation, which is also the result of drawing on the presented work (Hou et al., 2020). We utilize the local reparameterization trick for convolutional neural networks to extract EEG signal features and classify four MI tasks, which improved performance compared with other deep learning models. By doing so, the global uncertainty of weights is transformed into a local uncertainty, and the samples are also independent. As long as global noise can be transformed into local noise, the local reparameterization trick can generate a useful gradient estimator. For the proposed model, we introduce an equation $a_{m,j} = \gamma_{m,j} + \sqrt{\delta_{m,j}}\varsigma_{m,j}$, with $\varsigma_{m,j} \sim N(0,1)$(Eq.24) into the feature calculation function (i.e., forward function) of the CNN structure. Since the convolutional layer will pass through this function during forward propagation, the model can extract the features of the EEG signal layer by layer through this operation. Therefore, the model can generate a good estimator and then reduce the variance of the sampling points to adapt to the EEG MI signal. Through the above method, the proposed model will converge faster and promote the classification performance powerfully. Next, we will show more details about the proposed model.

We explored five deep learning models (Convolutional Neural Network, CNN; Recurrent Neural Network, RNN; Long Short-term Memory, LSTM; Gate Recurrent Unit, GRU; Deep Neural Network,

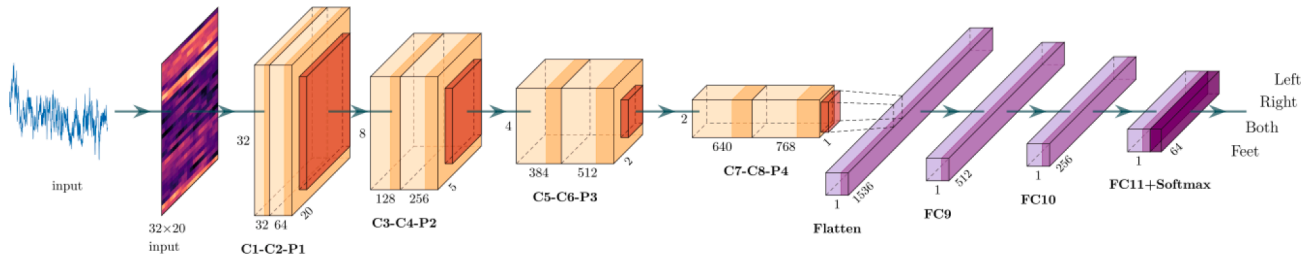


Fig. 2. The architecture of the proposed LRT-CNN. Convert the EEG signal of 1×640 dimension into 32×20 dimension as the input of LRT-CNN. Through the convolutional layer (yellow box), the pooling layer (red box), and the fully connected layer (purple box) in turn, the EEG signal is divided into four categories by the softmax function. We draw C1-C2-P1 as a group for clarity. Here, C1, C2, and P1 denote the first convolutional layer, the second convolutional layer, and the first pooling layer. The following symbols, and so on. Note that FC9, FC10, and FC11 denote three fully connected layers.

Table 1
The implementation details of the presented architecture.

	Activation shape	Activation Size	# parameters
Input:	(32, 20, 1)	640	0
C1(f = 3, s = 1, p = 1)	(32, 20, 32)	20,480	320
C2(f = 5, s = 1, p = 2)	(32, 20, 64)	40,960	51,264
P1(f = 2, s = 2, p = 0)	(16, 10, 64)	10,240	0
C3(f = 3, s = 1, p = 1)	(16, 10, 128)	20,480	73,856
C4(f = 5, s = 1, p = 2)	(16, 10, 256)	40,960	819,456
P2(f = 2, s = 2, p = 0)	(8, 5, 256)	10,240	0
C5(f = 3, s = 1, p = 1)	(8, 5, 384)	15,360	885,120
C6(f = 5, s = 1, p = 2)	(8, 5, 512)	20,480	4,915,712
P3(f = 2, s = 2, p = 0)	(4, 2, 512)	4,096	0
C7(f = 3, s = 1, p = 1)	(4, 2, 640)	5,120	2,949,760
C8(f = 5, s = 1, p = 2)	(4, 2, 768)	6,144	12,288,768
P4(f = 2, s = 2, p = 0)	(2, 1, 768)	1,536	0
FC9	(512, 1)	512	786,944
FC10	(256, 1)	256	131,328
FC11	(64, 1)	64	16,448
Softmax	(4, 1)	4	256

C1: the first convolutional layer, P1: the first pooling layers, C2: the second convolutional layer, P2: the second pooling layers, and so on; FC9, FC10, and FC11 denotes three fully connected layers; f denotes filter size, s denotes stride length, p denotes padding length; the activation shape of the convolutional layer and the pooling layer can be defined as: $(input_{width}, input_{length}, input_{channel})$.

DNN) that have been widely used to classify EEG signals. Here, we took a group of 20 subjects as an example. As shown in Fig. 3 (a), we can see that CNN stood out among the five models. Therefore, we applied the local reparameterization trick for convolutional neural networks (LRT-CNN), which improved the classification accuracy compared with the CNN model. Specifically, the CNN model achieved 87% maximum ACC. However, after introducing the local reparameterization trick into CNN, the ACC increased from 87% to 94%. In order to represent the results of the proposed model better, besides the above discussed ACC, as shown in Fig. 3 (b)-(g), we calculated the confusion matrices which can reveal the accuracy of different categories in each model (Bhatt, Ganatra, & Kotecha, 2021) to make a comparison of the classification performance between these models. From Fig. 3, we can see that the LRT-CNN model leads other deep learning models in global accuracy and has an advantage in a single class (e.g., 94.22%, 93.93%, 94.59%, and 90.65% accuracy of L, R, B, and F class). In this way, we can compare different models' performance by distinguishing the accuracy of the four categories. Next, we discuss the number of convolutional layers, pooling layers, and fully connected layers. As shown in Fig. 4 (a) and (b), increasing the number of convolutional layers could improve accuracy. Although the model could converge faster when we applied six convolutional layers (Fig. 4 (c)), the test loss had shown a rising trend (Fig. 4 (d)).

In summary, the accuracy increased smoothly with the increasing number of convolutional layers, pooling layers, and fully connected layers. But when the layers of the model reach to eight convolutional layers, four pooling layers, and three fully connected layers, the accuracy rate will no longer increase. Therefore, in terms of complete accuracy and loss, we finally choose the model with eight convolutional layers as our training model. More implementation details are shown in Table 2.

Furthermore, we added batch normalization after each convolutional layer, pooling layer, and fully connected layer except for the last fully connected layer; thus, we could avoid network imbalance due to different data dimensions. Meanwhile, to prevent overfitting, 25% dropout was applied to the convolutional layer (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014), and 50% dropout was applied to the fully connected layer. We utilized the ELU activation function in all experiments, which can alleviate the gradient vanishing problem. Finally, Stochastic Gradient Descent (Zhang, 2004) with 512 batch sizes was optimized by the Adam optimizer with a 0.0001 learning rate.

4. Experiment and results

4.1. Description of dataset

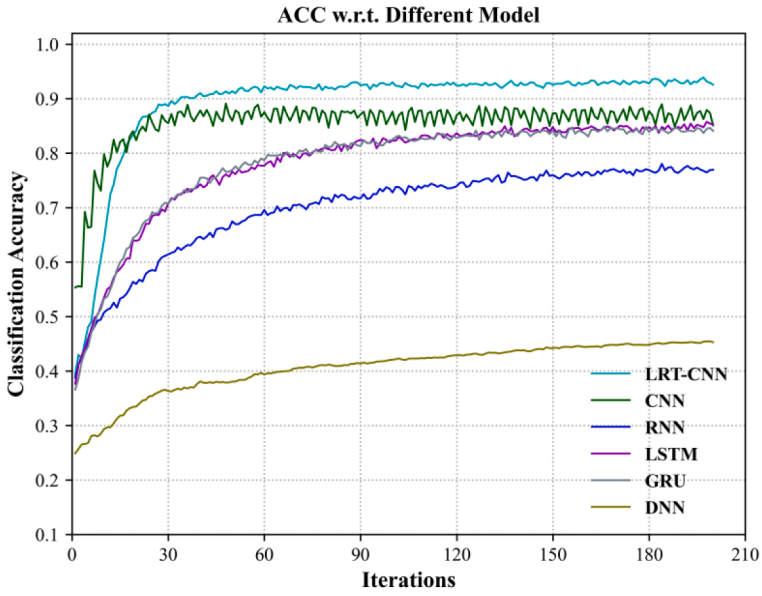
We used PhysioNet Dataset (Goldberger et al., 2000) to train the proposed LRT-CNN model. The dataset is publicly available[1], it contains 109 subjects with a sampling rate of 160 Hz per second, divided into 21 trails, and the MI process is 4 s, i.e., each subject has 84 trails. Each participant was asked to perform four types of motor imaging: left fist MI (L), right fist MI(R), both fists MI(B), feet MI(F), while 64-channel EEG signals were recorded. According to the published study (Handiru & Prasad, 2016), 17-channel (C1-C6, CP1-CP6, P1-P5) were selected in this study. After normalizing the EEG data, we utilized a group of 20 subjects' data with 28,560 samples to train and test our model. To estimate the proposed method's robustness and efficiency between the groups, we validated our approach by training and testing 50 subjects data with 71,401 samples, 80 subjects' data with 114,240 samples, and 109 subjects' data with 155,653 samples.

4.2. 10-fold cross-validation

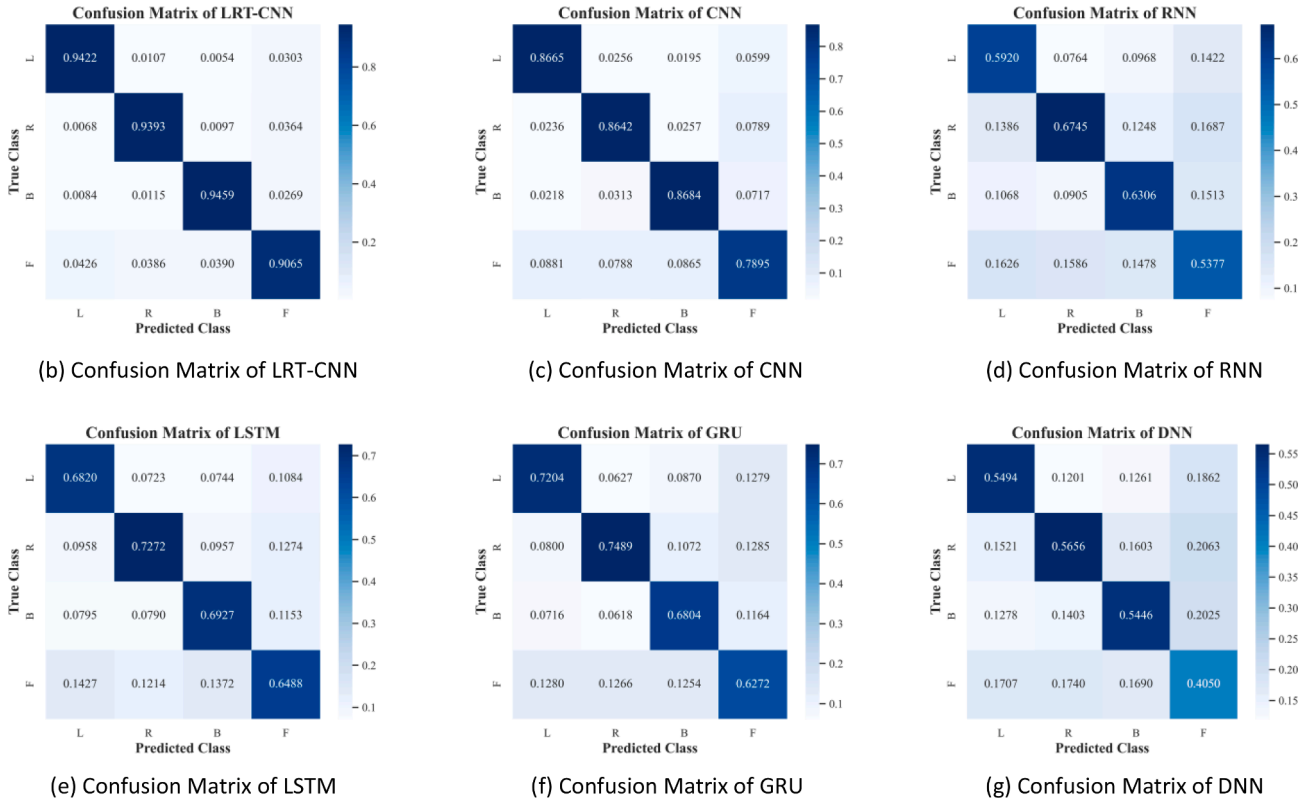
To estimate the proposed method's stability, accuracy, and reliability, we divided a dataset into ten commutative exclusive subsets of nearly equal size, yielding ten random separations of the primitive example. Therefore, we executed cross-validation ten times in all experiments. As shown in Eq. (31), we added each time and average results, proving the results obtained are reliable and robust.

$$Result = \frac{1}{10}\sum_{k=1}^{10} result^{(k)} \tag{31}$$

[1] https://archive.physionet.org/pn4/eegmmidb/.



(a) Accuracy w.r.t. Deep Learning Models



(b) Confusion Matrix of LRT-CNN (c) Confusion Matrix of CNN (d) Confusion Matrix of RNN

(e) Confusion Matrix of LSTM (f) Confusion Matrix of GRU (g) Confusion Matrix of DNN

Fig. 3. The performance between different models. As shown in (a), the LRT-CNN model achieved the highest accuracy. The confusion matrices of different models were shown in (b)-(g). Through the confusion matrix, we can see the performance of the model on a single class. L: Left fist; R: Right fist; B: Both fists; F: Feet.

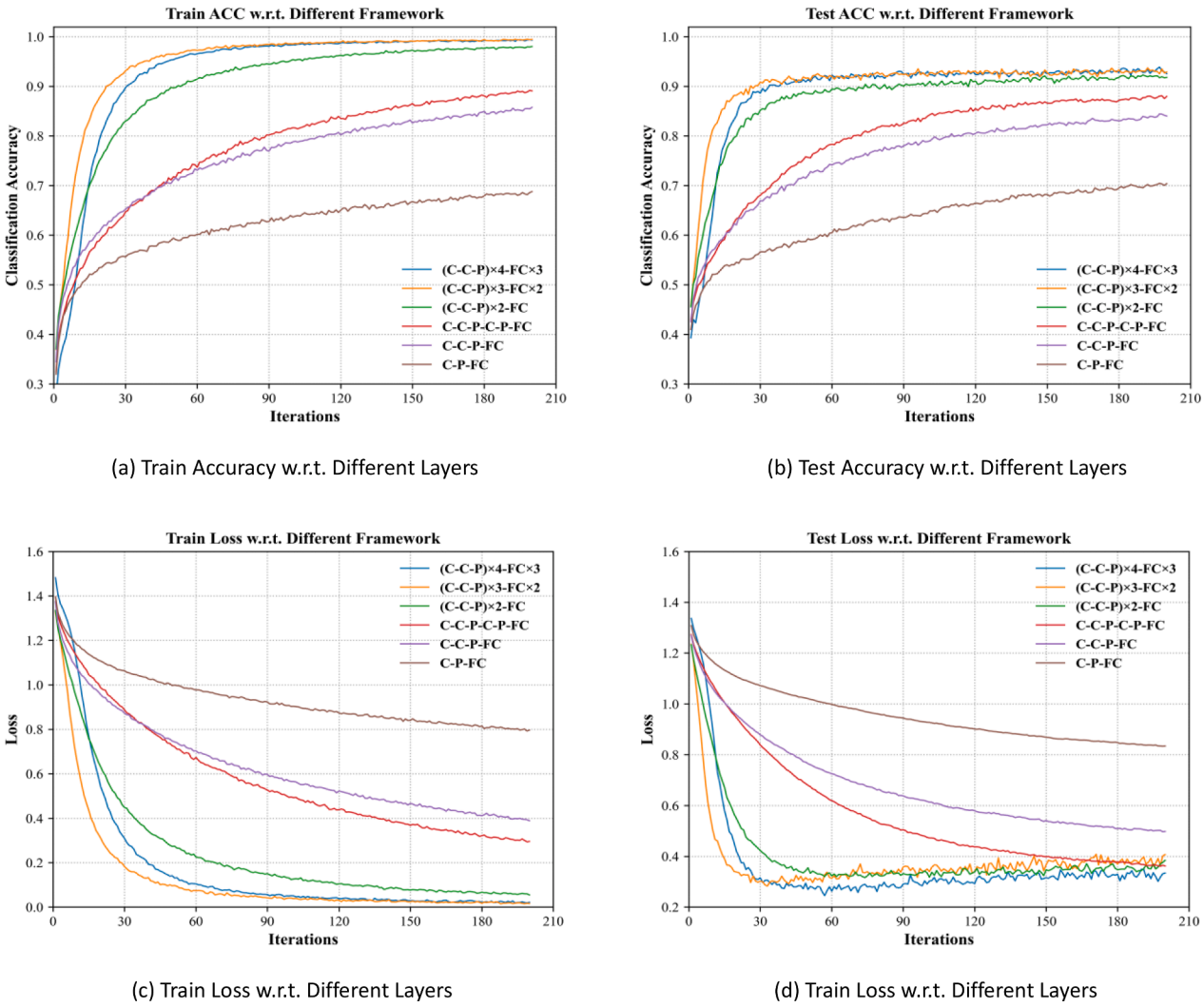
4.3. Statistical analysis

In this study, to estimate and compare with the performance of distinct methods, several statistical parameters were evaluated. For clearness, true positive, true negative, false positive, and false negative are assigned to TP, TN, FP, and FN, respectively. Based on these values, accuracy (ACC), Kappa, individual class accuracy, precision, recall, sensitivity, F1-score, Receiver Operating Characteristic Curve (ROC Curve), and Area Under Curve (AUC) are applied to appraise the performances (Siuly, 2011; Zhu, Li, & Wen, 2014).

Accuracy (ACC): Describes the rate of true positive and true negative to all the predictions.

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{32}$$

Kappa: Kappa is a robust measure since it considers the possibility of the agreement obtained by chance, where$P_e$is the theoretical probability



(a) Train Accuracy w.r.t. Different Layers

(b) Test Accuracy w.r.t. Different Layers

(c) Train Loss w.r.t. Different Layers

(d) Train Loss w.r.t. Different Layers

Fig. 4. Different frameworks of the LRT-CNN model. Here, C represents the convolutional layer, P represents the pooling layer, FC represents the fully connected layer. In detail, (a) and (b) display training accuracy and test accuracy for different frameworks; (c) and (d) display training loss and test loss for different frameworks.

of agreement.

$$Kappa = \frac{ACC - P_e}{1 - P_e} \tag{33}$$

Precision: Describes the probability that the prediction is correct in the result of the positive prediction.

$$Precision = \frac{TP}{TP + FP} \tag{34}$$

Recall: Describes the proportion of true positive cases in all positive cases.

$$Recall = \frac{TP}{TP + FN} \tag{35}$$

F1 score: As a performance of precision and recall score, mathematically, it can be defined as the mean of precision and recall score.

$$F1\ Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{36}$$

Specificity: Describes the proportion of true negative cases in all negative cases.

$$Specificity = \frac{TN}{TN + FP} \tag{37}$$

4.4. Classification w.r.t. group-level

In terms of the number of subjects, it has caused great concern in interpreting EEG signals (Arnau-Gonzalez, Katsigiannis, Ramzan, Tolson, & Arevalillo-Herrez, 2017; Dose et al., 2018; Handiru & Prasad, 2016; Hou et al., 2020; Loboda, Margineanu, Rotariu, & Mihaela, 2014; Ma, Qiu, Du, Xing, & He, 2018; Park et al., 2014). First of all, we used 20 subjects and 50 subjects for the group experiment. Furthermore, to demonstrate the proposed model's performance, we calculated six evaluation metrics for 20 subjects and 50 subjects. As illustrated by Fig. 5, the median values of ACC, Kappa, Precision, Recall, F1 Score, and Specificity are 92.9%, 90.53%, 92.82%, 92.85%, 92.79%, and 97.51%, respectively. Besides, at the 50-subject level, the median values of ACC, Kappa, Precision, Recall, F1 Score, and Specificity are 98.53%, 98.04%, 98.53%, 98.53%, 98.52%, and 99.51%, respectively. The six metrics' median values are greater than 90%, indicating that the proposed model works well.

Further, to verify the proposed model's robustness in terms of the number of subjects, we used 20 subjects (S1–S20), 50 subjects (S1–S50), 80 subjects (S1–S80), and 109 subjects (S1–S109) to validate our model. Fig. 6(a) shows that the group-level accuracy on 20 subjects, 80 subjects, and 109 subjects are 93.86%, 98.94%, 92.91%, and 92.94%. Besides, the AUC of 20 subjects, 50 subjects, 80 subjects, and 109 subjects are

Table 2
The implementation details of the presented architecture.

Model	Conv Layers	Pooling Layers	Fully connected Layers	Filter	Accuracy
C-P-FC	1	1	1	1/32	70.46%
C–C-P-FC	2	1	1	1/32/64	84.44%
C–C-P-C-P-FC	3	2	1	1/32/64/128	88.1%
(C–C-P) × 2-FC	4	2	1	1/32/64/128/256	92.28%
(C–C-P) × 3-FC × 2	6	3	3	1/32/64/128/256/384/512	93.71%
(C–C-P) × 4-FC × 3	**8**	**4**	**3**	**1/32/64/128/256/384/512/640/768**	**93.86%**

C: the convolutional layer, P: the pooling layer, FC: the fully connected layer. For readability, take (C–C-P) × 2-FC as an example, expand (C–C-P) × 2-FC into C–C-P-C–C-P-FC, it means that two convolutional layers and one pooling layer appear twice in a row. Finally, one fully connected layers are connected. To explain the meaning of the filter, take C–C-P-C-P-FC as an example. 1/32/64/128 are all for the convolutional layer, which means 1 is the input channel, and 32 is the output channel in the first convolutional layer. In the second convolutional layer, 32 is the input channel, and 64 is the output channel. Finally, 64 is the input channel, and 128 is the output channel in the third convolutional layer.

0.9918, 0.9996, 0.9923 and 0.9919(Fig. 6 (b)). Significantly, A group of 50 subjects has achieved the best performance at the group level. On the other hand, single-class accuracy is also an essential indicator to evaluate the quality of the model. As shown in Fig. 7, we only discuss 20 subjects and 50 subjects because 80 subjects, 109 subjects, and 20 subjects have similar results. The mean single class accuracy on L, R, B, and F of 20 subjects and 50 subjects are 97.29%, 97.24%, 97.91%, 93.85% and 99.50%, 99.37%, 99.46%, 98.74%, respectively. Also, the mean global accuracy (Combining L, R, B, and F) of 20 subjects and 50 subjects is 93.14% and 98.53%, respectively. Considering the 20 subjects' group and the 50 subjects' group, we can see that the model performed well in class B (at least w.r.t. the 20-subject group and the 50-subject group).

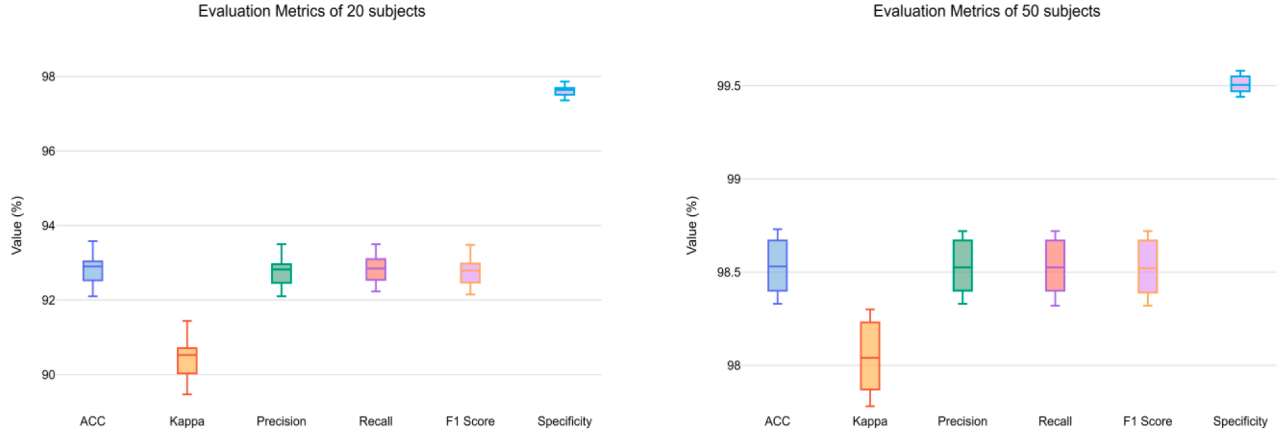
In summary, the proposed model can learn a more extensive range of EEG signal features with low spatial resolution features. The reason is that the LRT-CNN model can handle more subjects and consider the time and spatial dimension of the EEG signal, which achieves good effectiveness and robustness for EEG MI decoding. Besides, these results mean that when an EEG-based MI recognition system is deployed for a large number of users, this may significantly impact the biometric application of MI, thereby providing more unique features to distinguish different subjects.

4.5. Classification w.r.t. subject-level

Except focusing on experiments between groups, some investigators have tried to study and improve the effectiveness and robustness of different subjects under the same model (Dose et al., 2018; Hou et al., 2020; Kim et al., 2016; Tolic & Jovic, 2013). Thus, Evaluating the variability between different individuals is very critical. So, we selected fifty subjects from the PhysioNet Dataset to validate the individual variability. Fig. 8 shown that the highest mean accuracy was 99.79% received by subject 16(S16), and the lowest mean accuracy was 85.45% received by subject 5(S5). Meanwhile, the confusion matrix of subject 5 (S5) and subject 16(S16) were shown in Fig. 9 (a) and Fig. 9 (b), respectively. The single class accuracy on L, R, B, F was 89.19%, 83.78%, 87.5%, 81.08% from S5 and 100%, 97.06%, 100%, 100% from S16.

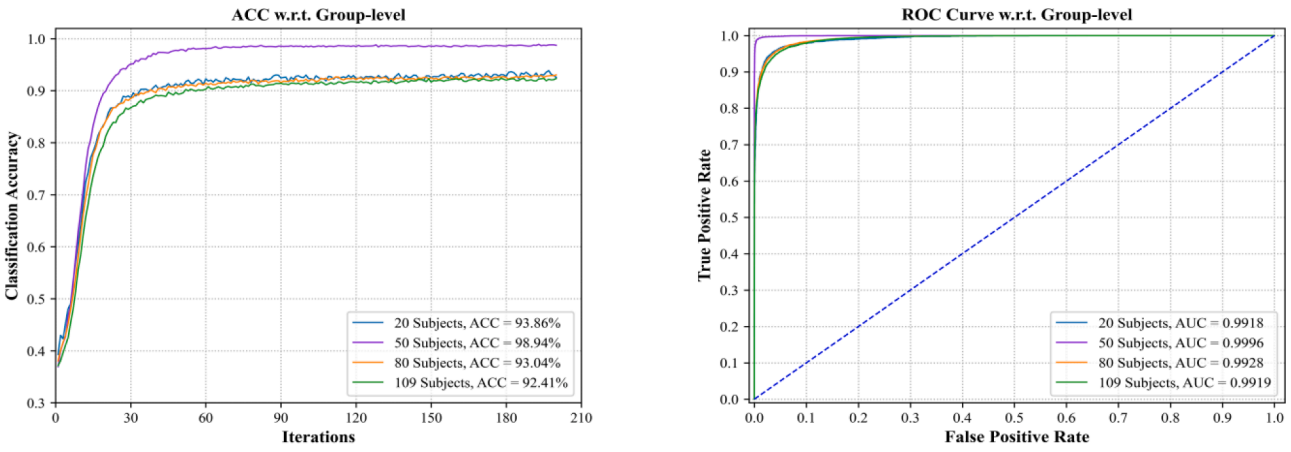
Furthermore, we randomly picked ten subjects to observe the subject-level performance by calculating six significant metrics. On average, we achieved promising results: 94.67% accuracy, 92.87% Kappa, 94.76% Precision, 94.85% Recall, 94.59% F1 Score, and 98.23% Specificity (see Table 3). The ROC curve and the corresponding AUC were shown in Fig. 10, from which the maximum AUC was 0.9999—achieved by subject 37(S37). These results reveal that although different subjects have different EEG characteristics, our model can identify the characteristics of these individual EEG signals better. In other words, the adaptability of the proposed model to different individuals has explained the differences in performance between subjects.

To sum up, in the research experiment on individual variability, we have obtained an average accuracy rate of more than 90%. In terms of the accuracy of a single class, the proposed model achieves a minimum accuracy of over 80% and a maximum accuracy of 100%, which means that if this model is used in the BCI online system, it can significantly help the patients who are paralyzed by a stroke or whose limbs are unable to move normally because of a limb disease. Due to the high accuracy of the model, the patients' recovery time will also be primarily accelerated. The results indicated that due to the robustness and effectiveness of the LRT-CNN model, it could effectively deal with individual variability issues.



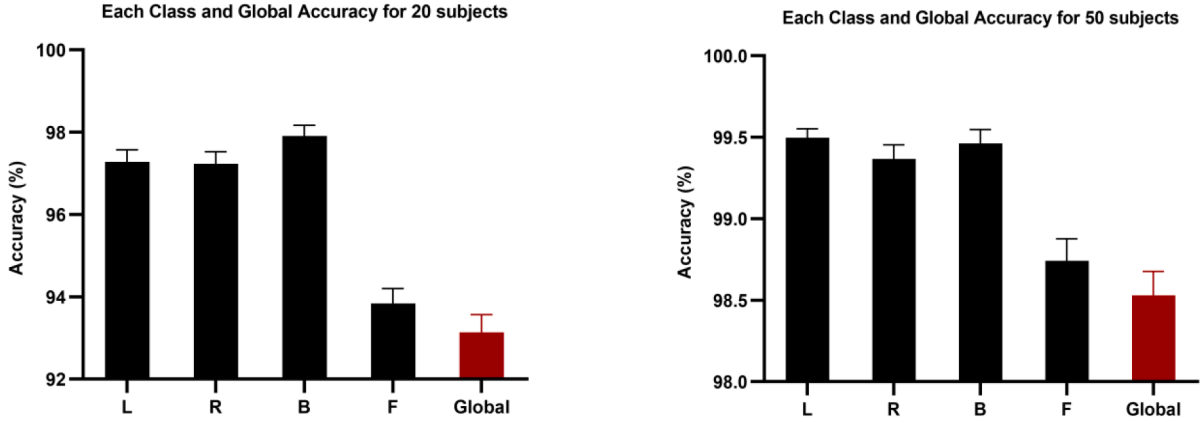
(a) Evaluation Metrics of 20 subjects (b) Evaluation Metrics of 50 subjects

Fig. 5. Evaluation metrics of 10-fold cross-validation for 20 subjects and 50 subjects. These six metrics are commonly used to evaluate the strength of classification performance.



(a) Accuracy w.r.t. Group-level (b) ROC Curve w.r.t. Group-level

Fig. 6. Accuracy and ROC curve between groups. (a): ACC means classification accuracy. (b): AUC is Area Under Curve, which measures the pros and cons of the classifier.



(a) Accuracy of 20 subjects (b) Accuracy of 50 subjects

Fig. 7. Single class accuracy and global accuracy on 10-fold cross-validation of the 20-subject group and the 50-subject group. L: Left fist. R: Right fist. B: Both fists. F: Feet. Global: Overall classification accuracy.

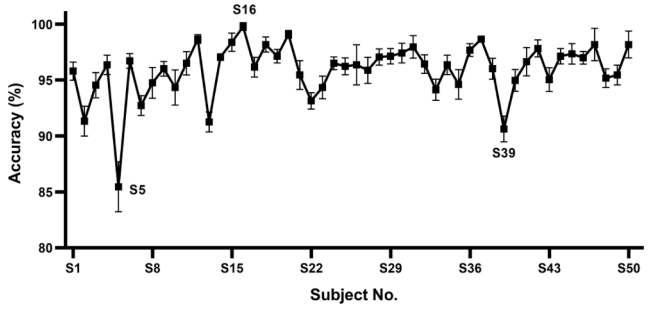


Fig. 8. 10 Times Testing for 50 Individual subjects. Each point (50 points in total) represents the result (ACC) of one subject.

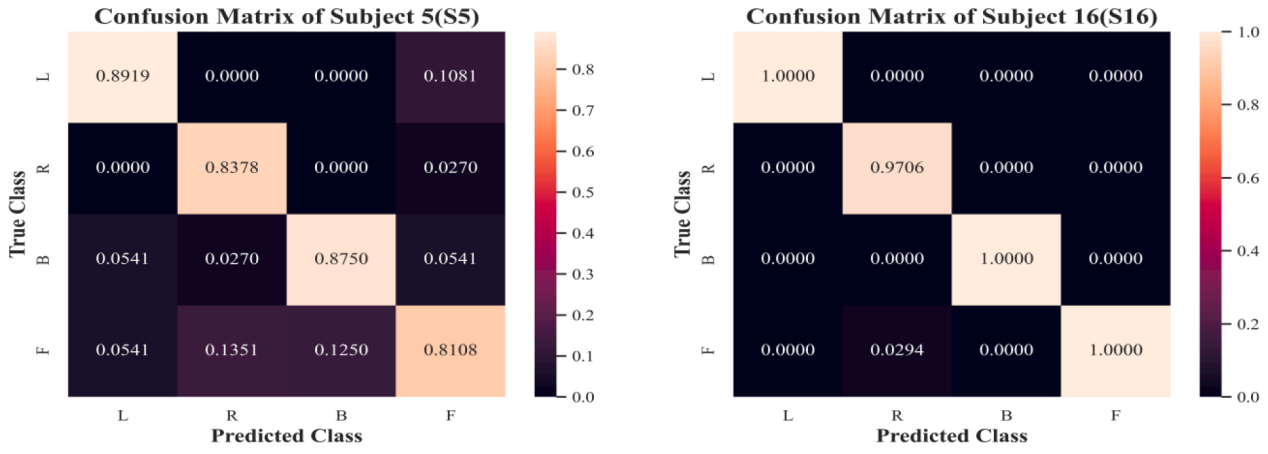
5. Discussion

We found that applying the local reparameterization trick in convolutional neural networks can improve EEG-based classification performance and accelerate the convergence speed during training. Specifically, the assumption is too strong (i.e., the z_i under the q distribution is required to be independent of each other), and the integral is intractable in variational inference (VI). However, the Stochastic Gradient Variational Bayes (SGVB) can solve the problems mentioned above, but SGVB has high variance when we are sampling. Finally, the local reparameterization trick can yield an efficient gradient estimator that can solve the high variance problem caused by SGVB. At the same time, this method can reduce the amount of calculation, which can greatly save computing resources.

To clarify the finding, based on the above methods, we performed two series of experiments: (1) designed a global classifier for a group of subjects. (2) Confirmed the robustness of specific adaptation for individual subjects. Concerning the first research question, it was found that the global classifier is suitable for the situation where multiple subjects are grouped together. The second question in this research was to validate individual variability in a single subject to demonstrate that our model is effective, and it was also found to be the case. Collectively, our results prove that utilizing the local reparameterization trick in convolutional neural networks can handle the challenge of individual variability between subjects.

5.1. Comparison with related works w.r.t. group-level

We use maximum accuracy (Max. ACC) metric and average accuracy (Avg. ACC) as comparative indicators for the related work (Arnau-Gonzalez et al., 2017; Dose et al., 2018; Handiru & Prasad, 2016; Hou



(a) Confusion Matrix of S5 (b) Confusion Matrix of S16

Fig. 9. Confusion Matrix of S5 and S16. L: Left fist, R: Right fist, B: Both fists, F: Feet.

Table 3
Evaluation metrics of ten subjects.

	S2	S5	S7	S13	S17	S28	S36	S37	S47	S50	**Average**
ACC (%)	91.33	85.45	92.73	91.26	96.15	97.06	97.69	98.67	98.18	98.18	**94.67**
Kappa (%)	88.42	80.60	90.25	88.34	94.83	96.04	96.91	98.22	97.57	97.54	**92.87**
Precision (%)	91.79	85.61	92.78	91.34	96.67	97.21	97.52	98.71	98.25	97.68	**94.76**
Recall (%)	91.23	86.27	92.60	91.54	96.55	97.07	97.99	98.66	98.19	98.41	**94.85**
F1 Score (%)	90.70	85.69	92.21	91.34	96.42	97.12	97.68	98.67	98.15	97.96	**94.59**
Specificity (%)	97.11	95.13	97.60	97.09	98.68	99.00	99.26	99.55	99.39	99.44	**98.23**

S2: subject 2, and so on. Note that all the metrics of ten subjects are the average results after ten times experiments.

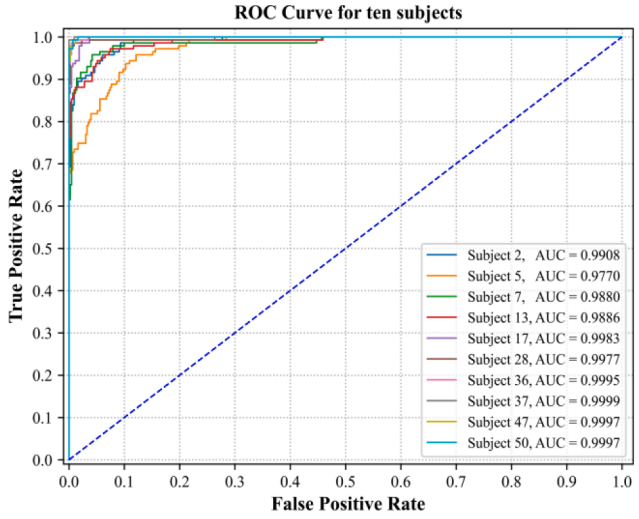


Fig. 10. Roc Curve for ten subjects. AUC: Area Under Curve.

Table 4
Comparison with related work for a group of subjects on the PhysioNet Dataset.

Work	Subjects	Max. ACC	Avg. ACC	Method
Park et al. (2014)	56	72.37%	–	SUT-CCSP + SVM
Loboda et al. (2014)	103	71.55%	–	Phase information
Handiru and Prasad (2016)	85	–	63%	FB-CSP + SVM
	35	–	79.9%	
Arnau-Gonzalez et al. (2017)	23	–	94.01%	1-D CNN
Ma et al. (2018)	12	82.65%	68.2%	RNNs
Dose et al. (2018)	105	–	65.73%	CNNs
Hou et al. (2020)	10	94.5%	–	ESI + CNNs
This work	**20**	**93.86%**	**92.98%**	**LRT-CNN**
	30*	**98.67%**	**97.46%**	
	50	**98.94%**	**98.53%**	
	80	**93.04%**	**92.86%**	
	109	**92.41%**	**92.37%**	

Max. ACC: maximum classification accuracy, Avg. ACC: average classification accuracy. Note that the Avg. ACC in our work is the average result of 10-fold cross-validation. **30***: Subjects from S21-S50.

et al., 2020; Loboda et al., 2014; Ma et al., 2018; Park et al., 2014). As listed in Table 4, all related work using the same dataset, the PhysioNet Dataset (Goldberger et al., 2000), reduces the impact caused by different datasets. Hou et al. (2020) proposed a novel approach based on scout EEG source imaging (ESI) and convolutional neural network (CNN) for MI tasks, which belongs to 2-Dimensional CNNs. They extracted the features through ESI and Morlet wavelet methods, then used these features as the input of CNN, and achieved competitive results (94.5% ACC at 10-subject level). However, this method has to go through a series of steps source computation, scout creation, feature extraction and frequencies separation, which takes lots of time during this feature engineering process. Dose et al. (2018) proposed an end-to-end deep learning model (1-Dimensional CNNs) to extract the signal features from raw EEG data. They reached an average cross-validation accuracy of 65.73% for four-class MI classification tasks. However, this method did not make the preprocessing work for the raw EEG data that contains many external artifacts, such as eye movement artifacts, facial muscle artifacts, etc. These artifacts significantly affect the classification performance of the CNN model, which is the reason for the lack of accuracy. Therefore, there is still room for improvement in classification tasks' accuracy in their research. In addition, although Arnau-Gonzalez's work (1-Dimensional CNN) (Arnau-Gonzalez et al., 2017) has achieved

Table 5
Comparison with related work for a single subject on the PhysioNet Dataset.

Work	Max. ACC	Method
Tolic and Jovic (2013)	72.82%	Wavelet transform DNN
Kim et al. (2016)	96.13%	SUT-CCSP Random forest
Dose et al. (2018)	86.49%	CNNs
Hou et al. (2020)	94.54%	ESI + CNNs
This work	**99.79%**	**LRT-CNN**

Max. ACC: maximum classification accuracy, note that Max. ACC in our work is the average result of 10 times.

promising results, a significant reason is that they have handled the artifact components well in the preprocessing process and extracted the effective time features of the EEGs. However, to learn EEG features in the spectral domain, the network model they proposed needs to be trained one million times, which is a considerable time overhead. Besides, their work did not consider the individual variability between the subjects and only selected a small number of subjects (23 subjects), which is fully considered in our research. In this study, we proposed a novel model that applies the local reparameterization trick to the CNN architecture and fed the EEG signal after preprocessing into this model. The main superiority of the proposed method is that it does not require the previous feature engineering works and considers the individual variability between subjects. Moreover, we conducted grouping experiments on 109 subjects, and the test results have shown that our model improved the overall accuracy of 1-D CNN in MI tasks.

Although the proposed model achieved ordinary results on 20, 80, and 109 subjects, our model performed unprecedentedly on 50 subjects. We achieved an ACC close to 99% on 50 subjects, which can be identified that our method has achieved the best performance at the 50-subject level of decoding EEG-based MI signals. The reason for this result is that the thirty subjects from S21 to S50 who participated in the MI experiment performed excellently, and the EEG signals fully characterized the MI features. Besides, the individual variabilities among these thirty subjects were very small. To verify this conclusion, we calculated the ACC for the thirty subjects from S21 to S50, respectively. As listed in Table 4, we got a maximum accuracy (Max. ACC) of 98.67% and averaged accuracy (Avg. Acc) of 97.46%, which indicates almost all of the thirty subjects achieved excellent results. Furthermore, as shown in Fig. 8, the accuracy of these thirty subjects are maintained at a stable and high level except for S39. But in comparison, the performance of subjects in other intervals (S1-S20 and S51-S109) is not stable. Moreover, some subjects obtained high accuracy (such as S16), and some got a relatively low accuracy (such as S5) (See Fig. 8), which indicates the individual variabilities between subjects in these two intervals are large or some subjects are not sufficiently involved in the MI experiment. Thus, the best performance was obtained at the 50-subject level.

5.2. Comparison with related works w.r.t. subject-level

We have also illustrated in Section 4.5 that the proposed model can be fitted into individual subjects, which achieves the highest accuracy performance in most cases (Dose et al., 2018; Hou et al., 2020; Kim et al., 2016; Tolic & Jovic, 2013). As listed in Table 5, the maximum classification accuracy was 99.79% achieved by the presented method. The experiment results prove that the proposed model can be applied to multiple subjects, and it is also applicable to the case of a single subject, which has an excellent ability to handle individual variability. We surprisingly found that the presented model represents an acceptable fit for small data while testing our approach at the subject level.

5.3. Computational complexity

Although having competitive accuracy, we have to discuss the computational complexity of the proposed model. The number of parameters can be used as an evaluation indicator for the computational complexity (He & Sun, 2015). As listed in Table 1, the number of parameters for the convolutional layers and the fully connected layers were calculated by the following methods, respectively.

5.3.1. Number of parameters for the convolutional layer

In the convolutional layer, the parameters of each layer are composed of weights and biases. Hence, the number of parameters is the sum of weights and biases. Generally, the number of parameters in the convolutional layer can be defined as:

$$P_c = \left(F^2 \times C + B\right) \times N \tag{38}$$

Here P_c is the number of parameters for the convolutional layer. F denotes filter size and C is the number of channels of the input data. B is the bias and equals one. N represents the number of the filter. An example in the first convolutional layer can be written as the complexity involved:

$$(3 \times 3 \times 1 + 1) \times 32 = 320 \tag{39}$$

This means that 32 3×3 filters are used for convolution operation in the first convolutional layer, and the number of parameters is 320.

5.3.2. Number of parameters for the fully connected layer

There are two cases to consider when calculating the number of parameters in the fully connected layer. We have to discuss them separately.

5.3.2.1. Fully connected layer connected to pooling layer. In this section, we discuss the connection between the last polling layer and the first fully connected layer. The number of parameters in this case can be defined as:

$$P_{pf} = (M \times \mathrm{Q} \times N + B) \times F \tag{40}$$

Here P_{pf} is the number of parameters for the first fully connected layer. $M \times \mathrm{Q}$ means the feature map for the last pooling layer. N is the number of filters in the last pooling layer. B is the bias and equals one. F represents the number of neurons in the first fully connected layer. Based on Eq. (40), we can calculate the number of parameters of the first fully connected layer:

$$(2 \times 1 \times 768 + 1) \times 512 = 786944 \tag{41}$$

5.3.2.2. Fully connected layer connected to fully connected layer. Finally, we discuss the connection between the fully connected layer and the fully connected layer, which is also the simplest case. The number of parameters can be defined as:

$$P_{ff} = (F' + B) \times F \tag{42}$$

Here P_{ff} is the number of parameters for the first fully connected layer. F' means the number of neurons in the previous fully connected layer. B is the bias and equals one. F represents the number of neurons in the current fully connected layer. Let's calculate the number of parameters of the first fully connected layer and the second fully connected layer:

$$(512 + 1) \times 256 = 131328 \tag{43}$$

Note that when calculating the number of parameters for the softmax layer, there is no need to calculate bias, so Eq. (42) will be transformed into:

$$P_{fs} = F' \times S \tag{44}$$

Here P_{fs} is the number of parameters for the softmax layer. F' means the number of neurons in the previous fully connected layer. S denotes the output of the softmax layer, i.e., the number of categories for classification tasks. Based on Eq. (44), the number of parameters of the

softmax layer is:

$$64 \times 4 = 256 \tag{45}$$

The computational complexity of each layer has been discussed above. As listed in Table 1, it is worth noticing that the computational complexity seems a little bit high due to numbers of parameters in each layer. But it does not affect the application of the proposed model in MI-EEG classification tasks. More importantly, the focus of this article is to verify the effectiveness of the proposed novel method, which has been proved efficiently by the high accuracy results. However, it does not mean that the computational complexity problem can be ignored. Lower computational complexity is essential for the promotion and expansion of the model. Therefore, based on the high accuracy results in this article, it is necessary to discuss the optimization algorithm of computational complexity for the proposed model. The critical idea is to reduce the overall computational complexity by optimizing the number of parameters in each layer, which is also an aspect that will be focused on in our follow-up research.

5.4. Limitation

Our study has limitations. These limitations mean that study findings need to be interpreted cautiously. Introducing the local reparameterization trick into the convolutional neural network can indeed improve the performance of the model. However, considering that the deep learning method is a "black box", we don't know what we have experienced during the neural network training phase. However, we can discuss the explainability of deep learning through a field called eXplainable Artificial Intelligence (XAI) (Barredo Arrieta et al., 2020; Gite, Khatavkar, Kotecha, Srivastava, Maheshwari, & Pandey, 2021). A deep understanding of XAI will help us better understand and interpret deep learning models' results. Since the main work of this article is to improve the accuracy of the EEG-based MI classification by using deep learning methods, it does not involve the XAI field, which makes the explainability of MI classification based on DL + XAI an essential aspect of our future work. So, this is the first limitation. The second limitation is the computational complexity of the proposed model. Through the discussion in section 5.3, we can quantify the specific parameters of each layer and it turns out that there are numbers of parameters, which leads to a bit of high computational complexity of the proposed model. As we discussed above, although improving accuracy represents the key concern for our research, the computational complexity of the model cannot be ignored. Therefore, our follow-up research will discuss this issue in depth.

6. Conclusions

Taken together, the proposed model applies the local reparameterization trick in the convolutional neural network, achieving encouraging results in EEG-based MI classification. Our findings demonstrate that deep learning (DL) can effectively address EEG classification problems. As a novel approach for improving MI tasks' performance, the proposed model has yielded convincing results that demonstrate our method adapted for multiple groups and individual subjects' testing. Furthermore, considering the rigor, we carefully analyzed the hyperparameters to try to find a better framework. Significantly, 10-fold cross-validation in all experiments has demonstrated the effectiveness and robustness of our proposed model. The proposed method can also reduce the EEG signal channels to save costs to complete the classification tasks, which provide greater possibilities for subsequent online utilization in BCI applications.

Compared with the current commonly used MI-EEG classification method, our method does not require specialists or special tools to complete complicated and tedious feature engineering work but only needs to do simple preprocessing of the original EEG signal to adapt to the DL model. As far as we know, the experimental results prove that our method is superior to that of the existing method. Meanwhile, future research should be focused on different datasets to confirm our method further.

We conclude that these results add to the rapidly expanding field of brain science and contribute to our understanding of applying the DL method to address classification problems (not limited to MI classification issues). Suppose we have good enough data when we are training our model. In that case, we only need to train an excellent framework to improve the accuracy further and possibly replace the current traditional method. More importantly, we speculate that introducing the local reparameterization trick into CNN is not only suitable for MI-EEG problems but also for other EEG problems (such as emotion recognition). We will try to prove this in future work.

CRediT authorship contribution statement

Wenqie Huang: Methodology, Visualization, Investigation, Writing – original draft, Writing – review & editing. **Wenwen Chang:** Supervision, Resources, Conceptualization, Writing – review & editing, Funding acquisition. **Guanghui Yan:** Supervision, Visualization, Methodology, Resources, Funding acquisition. **Zhifei Yang:** Software, Formal analysis, Investigation, Data curation, Visualization. **Hao Luo:** Data curation, Visualization, Conceptualization. **Huayan Pei:** Software, Conceptualization, Data curation.

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.

Acknowledgements

The authors would like to thank Shuyue Jia for his meaningful suggestions in this study. This work was supported by (i) the National Natural Science Foundation of China under grant 62062049, (ii) the Humanities and Social Science Foundation of the Ministry of Education under grant 20YJCZH212, (iii) the Science and Technology Project of Gansu Province under grant 20JR10RA215, and grant 20JR5RA390, and (iv) the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) under grant 202100020.

References

Aghaei, A. S., Mahanta, M. S., & Plataniotis, K. N. (2016). Separable Common Spatio-Spectral Patterns for Motor Imagery BCI Systems. *IEEE Transactions on Biomedical Engineering, 63*(1), 15–29. https://doi.org/10.1109/TBME.2015.2487738

Ang, K. K., Chin, Z. Y., Zhang, H., & Guan, C. (2008). Filter Bank Common Spatial Pattern (FBCSP) in brain-computer interface. In *Proceedings of the International Joint Conference on Neural Networks* (pp. 2390–2397). https://doi.org/10.1109/IJCNN.2008.4634130

Arnau-Gonzalez, P., Katsigiannis, S., Ramzan, N., Tolson, D., & Arevalillo-Herrez, M. (2017). ES1D: A Deep Network for EEG-Based Subject Identification. In *2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE)* (pp. 81–85). https://doi.org/10.1109/BIBE.2017.00-74

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., … Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion, 58*, 82–115.

Bashivan, P., Rish, I., Yeasin, M., & Codella, N. (2016). Learning representations from EEG with deep recurrent-convolutional neural networks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*.

Bentlemsan, M., Zemouri, E., Bouchaffra, D., Yahya-Zoubir, B., & Ferroudji, K. (2014). Random Forest and Filter Bank Common Spatial Patterns for EEG-Based Motor Imagery Classification. In *2014 5th International Conference on Intelligent Systems, Modelling and Simulation* (pp. 235–238). https://doi.org/10.1109/ISMS.2014.46

Bhatt, A. R., Ganatra, A., & Kotecha, K. (2021). Cervical cancer detection in pap smear whole slide images using convNet with transfer learning and progressive resizing. PeerJ Computer Science, 7, e348. doi:10.7717/peerj-cs.348.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association, 112*(518), 859–877. https://doi.org/10.1080/01621459.2017.1285773

Boureau, Y.-L., Ponce, J., & Lecun, Y. (2010). A Theoretical Analysis of Feature Pooling in Visual Recognition. ICML 2010 - Proceedings, 27th International Conference on Machine Learning, 111–118.

Clevert, D.-A. A., Unterthiner, T., & Hochreiter, S. (2016). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). Under Review of ICLR2016 (1997).

Dose, H., Møller, J. S., Iversen, H. K., & Puthusserypady, S. (2018). An end-to-end deep learning approach to MI-EEG signal classification for BCIs. Expert Systems with Applications, 114, 532–542. doi:10.1016/j.eswa.2018.08.031.

Gite, S., Khatavkar, H., Kotecha, K., Srivastava, S., Maheshwari, P., & Pandey, N. (2021). Explainable stock prices prediction from financial news articles using sentiment analysis. PeerJ Computer Science, 7, e340. doi:10.7717/peerj-cs.340.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., … Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet. *Circulation, 101* (23). https://doi.org/10.1161/01.CIR.101.23.e215

Grosse-Wentrup, M., & Buss, M. (2008). Multiclass common spatial patterns and information theoretic feature extraction. *IEEE Transactions on Biomedical Engineering, 55*(8), 1991–2000. https://doi.org/10.1109/TBME.1010.1109/TBME.2008.921154

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., … Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition, 77*, 354–377.

Gulcehre, C., Moczulski, M., Denil, M., & Bengio, Y. (2016). Noisy Activation Functions. Proceedings of the 33rd International Conference on International Conference on Machine Learning, 6, 3059–3068.

Handiru, V. S., & Prasad, V. A. (2016). Optimized Bi-Objective EEG channel selection and cross-subject generalization with brain-computer interfaces. *IEEE Transactions on Human-Machine Systems, 46*(6), 777–786. https://doi.org/10.1109/THMS.2016.2573827

He, K., & Sun, J. (2015). Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5353–5360).

Hou, Y., Zhou, L., Jia, S., & Lun, X. (2020). A novel approach of decoding EEG four-class motor imagery tasks via scout ESI and {CNN}. *Journal of Neural Engineering, 17*(1), 16048. https://doi.org/10.1088/1741-2552/ab4af6

Jiao, Z., Gao, X., Wang, Y., Li, J., & Xu, H. (2018). Deep Convolutional Neural Networks for mental load classification based on EEG data. Pattern Recognition, 76, 582–595. doi:10.1016/j.patcog.2017.12.002.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning, 37*(2), 183–233.

Kim, Y., Ryu, J., Kim, K. K., Took, C. C., Mandic, D. P., & Park, C. (2016). Motor Imagery Classification Using Mu and Beta Rhythms of EEG with strong uncorrelating transform based complex common spatial patterns. *Computational Intelligence and Neuroscience, 2016*, 1–13. https://doi.org/10.1155/2016/1489692

Kingma, D. P., Salimans, T., & Welling, M. (2015). *Variational dropout and the local reparameterization trick*. Advances in Neural Information Processing Systems.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations* (pp. 1–14).

Koutsou, A., Summa, S., Nasser, B., Martinez, J., & Thangaramanujam, M. (2014). Upper Limb Neuroprostheses: Recent Advances and Future Directions. *Biosystems and Biorobotics, 4*, 207–233. https://doi.org/10.1007/978-3-642-38556-8_11

Kumar, S., Sharma, A., Mamun, K., & Tsunoda, T. (2016). A Deep Learning Approach for Motor Imagery EEG Signal Classification. In *2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)* (pp. 34–39). https://doi.org/10.1109/APWC-on-CSE.2016.017

LaFleur, K., Cassady, K., Doud, A., Shades, K., Rogin, E., & He, B. (2013). Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain {\textendash}computer interface. *Journal of Neural Engineering, 10*(4), 46003. https://doi.org/10.1088/1741-2560/10/4/046003

LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K.-R. (2012). *Efficient BackProp BT - Neural Networks: Tricks of the Trade* (Second Edition, pp. 9–48). Berlin Heidelberg: Springer. https://doi.org/10.1007/978-3-642-35289-8_3

Liao, L.-D., Chen, C.-Y., Wang, I.-J., Chen, S.-F., Li, S.-Y., Chen, B.-W., … Lin, C.-T. (2012). Gaming control using a wearable and wireless EEG-based brain-computer interface device with novel dry foam-based sensors. *Journal of NeuroEngineering and Rehabilitation, 9*(1), 5. https://doi.org/10.1186/1743-0003-9-5

Loboda, A., Margineanu, A., Rotariu, G., & Mihaela, A. (2014). Discrimination of EEG-Based Motor Imagery Tasks by Means of a Simple Phase Information Method. *International Journal of Advanced Research in Artificial Intelligence, 3*(10). https://doi.org/10.14569/ijarai.2014.031002

Lu, N.a., Li, T., Ren, X., & Miao, H. (2017). A deep learning scheme for motor imagery classification based on restricted Boltzmann Machines. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 25*(6), 566–576. https://doi.org/10.1109/TNSRE.733310.1109/TNSRE.2016.2601240

Ma, X., Qiu, S., Du, C., Xing, J., & He, H. (2018). Improving EEG-Based Motor Imagery Classification via Spatial and Temporal Recurrent Neural Networks. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society Annual International Conference* (pp. 1903–1906). https://doi.org/10.1109/EMBC.2018.8512590

Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning* (pp. 807–814).

Park, C., Took, C. C., & Mandic, D. P. (2014). Augmented Complex Common Spatial Patterns for Classification of Noncircular EEG From Motor Imagery Tasks. *IEEE Transactions on Neural Systems and Rehabilitation Engineering : A Publication of the IEEE Engineering in Medicine and Biology Society, 22*(1), 1–10. https://doi.org/10.1109/TNSRE.733310.1109/TNSRE.2013.2294903

Santhanam, G., Ryu, S. I., Yu, B. M., Afshar, A., & Shenoy, K. V. (2006). A high-performance brain-computer interface. *Nature, 442*(7099), 195–198. https://doi.org/10.1038/nature04968

Shen, Y., Lu, H., & Jia, J. (2017). Classification of motor imagery EEG signals with deep learning models. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-319-67777-4_16

Silver, M. A., Ress, D., & Heeger, D. J. (2005). Topographic maps of visual spatial attention in human parietal cortex. *Journal of Neurophysiology, 94*(2), 1358–1371. https://doi.org/10.1152/jn.01316.2004

Sitaram, R., Caria, A., Veit, R., Gaber, T., Rota, G., Kuebler, A., & Birbaumer, N. (2007). FMRI brain-computer interface: A tool for neuroscientific research and treatment. *Computational Intelligence and Neuroscience, 2007*, 1–10. https://doi.org/10.1155/2007/25487

Siuly, Li, Y., & Wen, P. (Paul). (2011). Clustering technique-based least square support vector machine for EEG signal classification. Computer Methods and Programs in Biomedicine, 104(3), 358–372. doi:10.1016/j.cmpb.2010.11.014.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research, 15*(56), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html.

Sun, Y., Lo, F. P.-W., & Lo, B. (2019). EEG-based user identification system using 1D-convolutional long short-term memory neural networks. Expert Systems with Applications, 125, 259–267. doi:10.1016/j.eswa.2019.01.080.

Tolic, M., & Jovic, F. (2013). Classification of wavelet transformed EEG signals with neural network for imagined mental and motor tasks. *Kinesiology, 45*(1), 130–138.

Wainwright, M. J., & Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends® Machine Learning, 1*(1–2), 1–305. https://doi.org/10.1561/2200000001

Wang, T., Wu, D. J., Coates, A., & Ng, A. (2012). End-to-end text recognition with convolutional neural networks. Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), 3304–3308.

Wang, H., Dong, X., Chen, Z., & Shi, B. E. (2015). Hybrid gaze/EEG brain computer interface for robot arm control on a pick and place task. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 2015-Novem, 1476–1479. doi:10.1109/EMBC.2015.7318649.

Wang, Y., Gao, S., & Gao, X. (2005). Common spatial pattern method for channel selection in motor imagery based brain-computer interface. *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings, 7*, 5392–5395. https://doi.org/10.1109/iembs.2005.1615701

Wijnhoven, R. G. J., & de With, P. H. N. (2010). Fast Training of Object Detection Using Stochastic Gradient Descent. In *2010 20th International Conference on Pattern Recognition* (pp. 424–427). https://doi.org/10.1109/ICPR.2010.112

Yang, H., Sakhavi, S., Ang, K. K., & Guan, C. (2015). On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 2620–2623). https://doi.org/10.1109/EMBC.2015.7318929

Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-First International Conference on Machine Learning* (p. 116). https://doi.org/10.1145/1015330.1015332

Zhang, R., Li, Y., Yan, Y., Zhang, H., Wu, S., Yu, T., & Gu, Z. (2016). Control of a wheelchair in an indoor environment based on a brain-computer interface and automated navigation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 24*(1), 128–139. https://doi.org/10.1109/TNSRE.2015.2439298

Zhu, Guohun, Li, Yan, & Wen, Peng (2014). Analysis and Classification of Sleep Stages Based on Difference Visibility Graphs From a Single-Channel EEG Signal. *IEEE Journal of Biomedical and Health Informatics, 18*(6), 1813–1821. https://doi.org/10.1109/JBHI.622102010.1109/JBHI.2014.2303991

Zinkevich, M., Weimer, M., Li, L., & Smola, A. (2010). Parallelized Stochastic Gradient Descent. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, & A. Culotta (Eds.), Advances in Neural Information Processing Systems (Vol. 23, pp. 2595–2603). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2010/file/abea47ba24142ed16b7d8fbf2c740e0d-Paper.pdf.