# Dynamic convolution with multilevel attention for EEG-based motor imagery decoding

| | |
|---|---|
| Journal: | *IEEE Internet of Things Journal* |
| Manuscript ID | IoT-25991-2022.R1 |
| Manuscript Type: | Special issue on Smart Cities and Systems: Theories, Tools, Trends, Applications, Challenges, and Opportunities |
| Date Submitted by the Author: | 06-Mar-2023 |
| Complete List of Authors: | Altaheri, Hamdi; King Saud University, Department of Computer Engineering, College of Computer and Information Sciences<br>Muhammad, Ghulam; King Saud University, College of Computer and Information Sciences, Dept. of Computer Engineering<br>Alsulaiman, Mansour; King Saud University, Department of Computer Engineering, College of Computer and Information Sciences |
| Keywords: | Sensor Signal Processing < Sub-Area 1: Sensors and Devices for IoT, eHealth and mHealth < Sub-Area 3: Services, Applications, and Other Topics for IoT, Smart Cities < Sub-Area 3: Services, Applications, and Other Topics for IoT |
| | |

*

# Dynamic convolution with multilevel attention for EEG-based motor imagery decoding

Hamdi Altaheri, *Member, IEEE*, Ghulam Muhammad, *Senior Member, IEEE*, and Mansour Alsulaiman, *Senior Member, IEEE*

***Abstract*— Brain-computer interface (BCI) is an innovative technology that utilizes artificial intelligence (AI) and wearable electroencephalography (EEG) sensors to decode brain signals and enhance the quality of life. EEG-based motor imagery (MI) brain signal is used in many BCI applications including smart healthcare, smart homes, and robotics control. However, the restricted ability to decode brain signals is a major factor preventing BCI technology from expanding significantly. In this study, we introduce a dynamic attention temporal convolutional network (D-ATCNet) for decoding EEG-based motor imagery signals. The D-ATCNet model uses dynamic convolution and multilevel attention to enhance the performance of MI classification with a relatively small number of parameters. D-ATCNet has two main blocks: dynamic and temporal convolution. Dynamic convolution uses multilevel attention to encode low-level MI-EEG information and temporal convolution uses shifted window with self-attention to extract high-level temporal information from the encoded signal. The proposed model performs better than the existing methods with an accuracy of 71.3% for subject-independent and 87.08% for subject-dependent using the BCI Competition IV-2a dataset.**

***Index Terms*— Smart healthcare, convolutional neural network, dynamic convolution, attention mechanism, multi-head self-attention, classification, motor imagery, EEG.**

## I. INTRODUCTION

Recent advances in artificial intelligence (AI), cloud computing, edge computing, and communication technologies have broadened the scopes of smart cities to include all possible smart solutions and systems, including healthcare, environment, security, industry, and economics. Brain-computer interface (BCI) is an emerging technology that uses wearable sensors and AI to decode brain signals and utilize them to further enhance the quality of life. BCI has a wide range of smart city applications including smart healthcare and smart environments. BCIs are currently a growing technology due to the invention of various types of related Internet of Things (IoT) units that use 5G/beyond wireless communications. In this research, we focus on AI-based motor imagery (MI) decoding using electroencephalography (EEG) brain signals (MI-EEG) considering the limitations of IoT and wearable edge devices. In particular, we developed a lightweight deep learning (DL) model based on dynamic convolution and multilevel attention.

BCI is a system that establishes a direct connection between the human brain and a device, such as an IoT device, robotic limb, or computer. The BCI system captures the brain signal by measuring metabolic (functional near-infrared spectroscopy, fNIRS), magnetic (magnetoencephalography, MEG), or electrical (EEG) brain activities. EEG is broadly used and suitable for IoT and smart city applications due to its portability, minimal risk, low cost, and ease of use.

Motor imagery is an imaginative process where we simulate an action in our brains, without actually doing it. The action is usually a physical movement such as imagining the movement of a hand or a leg. MI is a popular BCI model that has many applications in both the medical and non-medical sectors. Medical applications include rehabilitation after a stroke, translating thoughts into texts, and controlling various medical equipment such as wheelchairs, prostheses, exoskeletons, electrical stimulation, and screen pointer [1]. Non-medical applications include augmenting human capabilities using an exoskeleton or robotic arm, controlling unmanned aerial/ground vehicles, environment control in smart homes, games and virtual reality, and even in security for authentication and identification [1]. Many of these applications are building blocks in smart cities, specifically in smart healthcare and smart homes.

BCI is a revolution in technology, but it is still restricted by the performance of brain signal decoding. Decoding the MI-EEG signal is challenging for the following reasons. The amplitude of the brain signal acquired in the scalp using EEG sensors has an approximate value of 10 - 100 μV representing about 5% of the original signal generated in the brain [1]. This weak signal has a very low signal-to-noise ratio (SNR) due to interference with a large amount of noise and artifacts from various sources, including non-MI brain activity, muscle movement, eye blinking, room lighting, power lines, and nearby electronic devices. These artifacts, along with the high dimensionality of EEG signals, channel correlation, and subject dependence make it difficult to interpret and decode brain signals.

Several traditional machine learning (ML) and deep learning methods have been introduced to overcome the challenges in decoding the MI-EEG signal. Unlike traditional approaches, DL can recognize deep and distinct features from raw brain

*Manuscript received on 01 October, 2022. It is IEEE style to display support information, including sponsor and financial support acknowledgment, here and not in an acknowledgment section at the end of the article. The acknowledgment will be added later. *Corresponding author*: Ghulam Muhammad (e-mail: ghulam@ksu.edu.sa)

All authors are affiliated with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia. E-mail: haltaheri@ksu.edu.sa (H.A.); ghulam@ksu.edu.sa (G.M.); msuliman@ksu.edu.sa (M.A.).

signals without the need to preprocess the data or extract features manually. DL has proven successful in several applications, such as video recognition, image classification, and audio forensics [2]–[4]. Inspired by these successes, several DL architectures have been proposed for brain signal decoding.

Over the past five years, studies using DL approaches to decode MI-EEG brain signals have expanded dramatically [1]. The most popular DL network is the convolutional neural network (CNN) [5]–[14]. Standard CNNs have been used to decode EEG-based MI signals using deep [11] and light [9] architectures. Other variants of CNN have also been proposed, including residual-based CNN [12], multi-layer CNN [14], multi-scale CNN [10], multi-branch CNN [6], [12], [13], 3D-CNN [12], inception-based CNN [7], [8], and attention-based CNN [5]–[8], [10], [15].

Several architectures other than CNNs have been proposed for decoding EEG-based MI signals, including recurrent neural network (RNN) [16], [17], autoencoder (AE) [18], deep belief network (DBN) [19], and hybrid DL networks [5], [14]. A DBN was proposed by Xu et al. [19] to decode four MI tasks using restricted Boltzmann machines. The DL model was used for feature extraction and the support vector machine (SVM) as a classifier. A stacked AE (SAE) was used by Hassanpour et al. [18] to decode MI tasks using spectral features. RNNs have been proposed by several studies to extract temporal MI features in EEG data. For instance, the study in [16] used a long short-term memory (LSTM) network and filter bank common spatial pattern (FBCSP) for feature extraction and SVM for classification. Another variant of RNN, a gated recurrent unit (GRU), was proposed by researchers in [17] and fused with FBCSP features. This study showed that GRU outperformed LSTM. In general, CNNs perform better at decoding the MI-EEG signal than other DL architectures [1]. As a result, many researchers proposed integrating CNN with other deep learning networks, such as SAE in [14] and LSTM in [5], and promising results were obtained from hybrid DL networks.

Recently, a temporal convolutional network (TCN) has been proposed for temporal data classification [20]. TCN has shown superior performance compared to other RNN models such as GRU and LSTM in a variety of sequence-related tasks [20]. TCNs have a wider receptive field with lower parameters than typical CNNs and avoid the issues of exploding and vanishing gradients that can occur with RNNs. TCN has been suggested by recent studies to decode EEG-based MI signals [15], [21], [22]. Ingolfsson et al. [21] proposed an EEG-TCN architecture by fusing a TCN with a CNN network called EEGNet [9]. The study in [22] improved the EEG-TCN architecture by utilizing multilevel feature fusion. Recent work by Altaheri et al. [15] introduced attention fusion with CNN and TCN to decode MI-EEG signals, and promising results were obtained.

In recent years, the attention mechanism has had a significant impact on the field of deep learning, allowing for more efficient and effective processing of complex datasets. Attention in deep learning is an attempt to mimic the way the brain selectively focuses on important items while ignoring surrounding information. By integrating attention with DL architectures, it becomes possible to automatically concentrate on significant elements in input data, feature maps, and even layer kernels.

In 2015, Bahdanau et al. [23] and Luong et al. [24] developed the first attention-based models, now known as additive and multiplicative attentions. In 2017, Vaswani et al. [25] introduced a new attention design called Transformer based on multi-head self-attention (MSA) and multilayer perceptron (MLP). These attention models were first introduced in the context of machine translation, specifically in encoder decoder-based neural machine translation, to address the problem of capturing the dependencies of tokens (e.g., words) in long sequences (e.g., texts). Later, attention has been employed in various applications other than natural language processing (NLP), such as computer vision and speech processing. Recently, several attention models have been proposed, especially for computer vision, such as convolutional block attention module (CBAM) [26], squeeze-and-excitation (SE) [27], and vision transformer (ViT) [28].

The attention mechanism can be applied to MI-EEG data in multiple dimensions. MI-EEG signals are high-dimensional data, i.e., sequences of time points recorded from tens to hundreds of channels distributed in the human scalp. Effective MI information in such high-dimensional data occurs at specific channel locations, frequency bands, and time intervals. Therefore, developing a deep learning architecture that automatically focuses (through learning) on the most effective data in the temporal, spatial, and spectral domains is a promising research direction to enhance the decoding accuracy of MI tasks, which is still low.

Several recent studies have proposed attention-based DL models for MI classification [5]–[8], [10], [15]. For example, researchers in [5] used graph neural representation and LSTM with self-attention to classify four MI activities. The authors in [7], [8] proposed an attention-based inception-CNN and LSTM architecture for MI classification. Altuwaijri et al [6] presented a multi-branch CNN model with SE attention blocks for decoding EEG-based MI tasks using a raw signal. Inspired by the vision transformer [28], Altaheri et al. [15] introduced an attention-based architecture for decoding MI-EEG signals, called ATCNet. ATCNet used a convolutional-based sliding window and multi-head self-attention (MSA) combined with a temporal convolutional network, which resulted in outstanding performance.

The attention models described above have been proposed for use in input data (feature maps). Recent studies suggested the use of the attention mechanism at the network kernel level [29], [30]. This approach, known as dynamic convolution, allows a neural network to have dynamic kernels that change based on the input data during inference. Dynamic convolution allows the learning of a rich representation with a fixed network structure (constant depth and width), resulting in higher performance with lightweight architectures suitable for IoT and edge devices.

In this article, we propose a dynamic convolution architecture, D-ATCNet, with multilevel attention at the input/feature level and kernel level for decoding EEG-based MI tasks. We highlight the following contributions in this study:

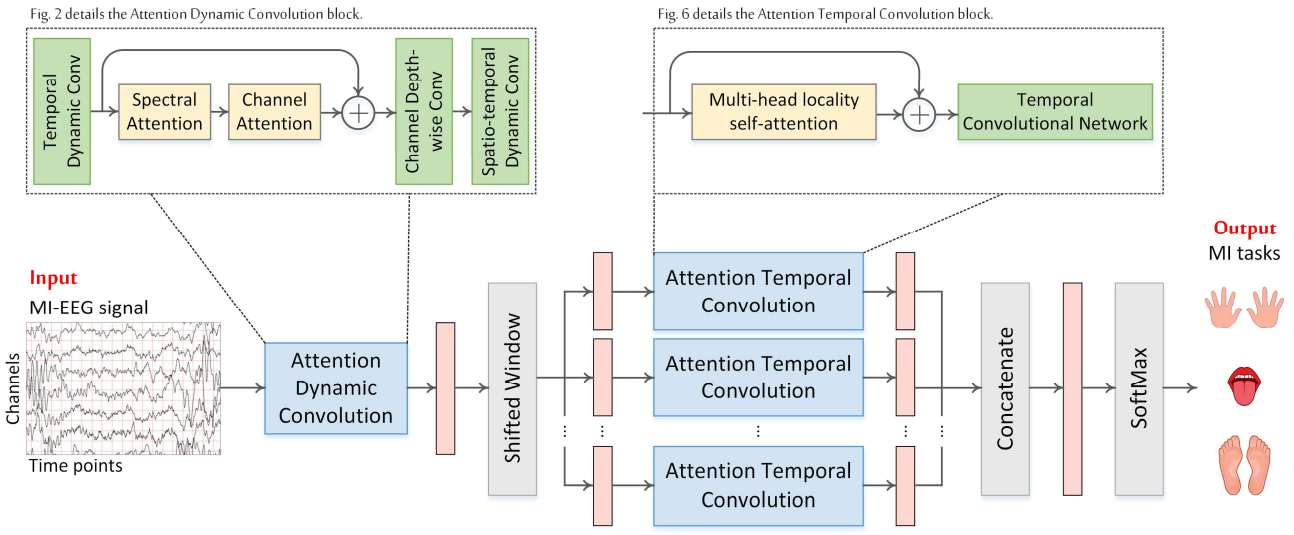


**Fig. 1.** Components of the D-ATCNet architecture. The temporal, spatial (channel), and spectral attentions refer to the attention along the temporal, spatial, and spectral dimensions of the EEG signal.

1. We propose an efficient and high-performance D-ATCNet architecture that takes advantage of multilevel attention, dynamic convolution, temporal convolution, and convolutional-based sliding window.
2. Multilevel attention is implemented at the kernel level and the input data level. We investigate the attention over the temporal, spatial, and spectral dimensions of the EEG signal. This helps the DL model to automatically focus on valuable features in multiple dimensions.
3. Dynamic convolution allows rich representations to be learned at the kernel level without increasing the width and depth of the network, resulting in better performance with lightweight networks.
4. The D-ATCNet architecture achieves superior performance on the BCI Competition IV-2a (BCI-IV2a) benchmark dataset [31].

The trained D-ATCNet model and its implementation code will be available on GitHub for reproduction. The remaining parts of the article are divided into the following sections. In Section II, we describe the proposed D-ATCNet architecture. Section III presents and discusses the results. Finally, we conclude in Section IV.

## II. Proposed Architecture

The proposed D-ATCNet model is an improvement of our previous ATCNet architecture [15]. ATCNet is inspired in part by the vision transformer (ViT) proposed by Dosovitskiy et al. [28]. ATCNet differs from ViT by the following:

- ViT uses single-layer linear projection while ATCNet uses multi-layer nonlinear projection, i.e., convolutional projection specifically designed for EEG-based brain signals.
- ViT consists of a stack of encoders where the output of the previous encoder is the input of the subsequent. ATCNet consists of parallel encoders and the outputs of all encoders are concatenated.
- The encoder block in ViT consists of an MSA followed by a multilayer perceptron (MLP) while in ATCNet the MSA is followed by a TCN.
- The first encoder in ViT receives the entire input sequence while each encoder in ATCNet receives a shifted window from the input sequence.

D-ATCNet differs from ATCNet by using multilevel attention at the kernel level using dynamic convolution and at the features level using MSA and SE. In features level attention, we investigate the temporal, spectral, and spatial domains of the EEG signal. D-ATCNet also uses multi-head locality self-attention (MLSA) [32] instead of multi-head self-attention [25], which helps learn attention representations for small datasets.

D-ATCNet has two main blocks: the attention dynamic convolution (ADC) block and the attention temporal convolution (ATC) block. The ADC block comprises three convolutional (conv) layers that utilize multilevel attention at the kernel and feature levels. This block captures low-level spatiotemporal features from the MI-EEG data and outputs a high-level temporal sequence. The temporal sequence is then divided into multiple windows using a shifted window (SW) with one step stride. These windows are fed into separate ATC blocks in parallel, as shown in Fig. 1. The ATC block consists of a multi-head locality self-attention followed by a temporal convolutional network. Multi-head attention highlights the most valuable features in each window and feeds the highlighted features into a TCN that extracts high-level temporal features. Finally, these features from all windows are aggregated and fed into a fully connected (FC) layer with SoftMax activation. SoftMax generates a probability prediction for the executed MI task. The following sections detail the D-ATCNet architecture.

### *A. Input Representation and Preprocessing*

From a machine-learning perspective, brain signal decoding can be viewed as a supervised classification problem [33]. This

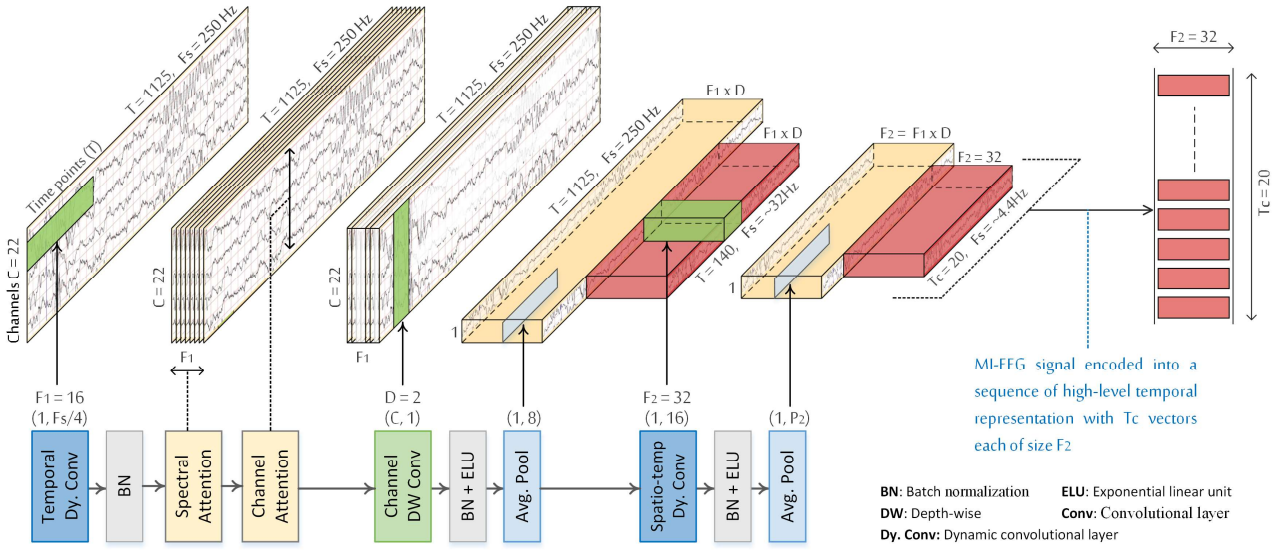


**Fig. 2.** Attention dynamic convolutional (ADC) block.

section demonstrates how brain signal decoding is represented as a classification problem.

Suppose that $S = \{X_i, y_i\}_{i=1}^{m}$ denotes a set of $m$ labeled MI trials (time segments of recorded MI brain signals each belonging to one of several classes), where $X_i \in \mathbb{R}^{C\times T}$ is an MI trial comprising of $C$ EEG electrodes (channels) and $T$ time points, and $y_i \in \{1, \dots, n\}$ is the class of trial $X_i$, where $n$ is the number of classes defined for the set $S$. Each class belongs to one of the MI activities performed in the trials, for example $\forall y_i: y_i \in L = \{l_1 = \text{"Left hand"}, l_2 = \text{"Right hand"}, l_3 = \text{"Feet"}, l_4 = \text{"Tongue"}\}$. The objective is to train a decoder $f$, a deep learning model, using the given set $S$ to map a new unseen MI trial $X_i$ to its corresponding class label $y_i$, where $f$ is a parametric classifier $f(X_i; \theta): \mathbb{R}^{C\times T} \rightarrow L$ with parameters $\theta$. For the BCI-IV2a [31] dataset, $C = 22$ EEG channels, $T = 1125$ time points, $m = 5184$ MI trials, and $n = 4$ MI classes.

MI-EEG signals are usually preprocessed and transformed into different input representations before being fed into ML or DL models. Preprocessing is an essential step in traditional ML methods and has also been used in many studies adopting DL methods. Preprocessing includes artifact removal, channel selection, signal filtering, and signal normalization [1]. After signal preprocessing, traditional ML methods extract features from preprocessed EEG data and feed them into an ML classifier. In DL methods, raw EEG data can be used as input for DL models as well as extracted features, spectral images, and topological maps. Our review [1] statistically showed that DL methods can produce competitive results using raw EEG signals compared to other input formulations.

This research uses raw EEG data without preprocessing, i.e., the full frequency range as the original signal (0.5 - 100 Hz), all channels (22), and without removing artifacts. MI-EEG signals are standardized before being fed into the D-ATCNet model as in Eq. 1, where $c$ denotes the number of EEG channels:

$$x_i' = \frac{x_i - mean(x_i)}{std(x_i)}, \; i = 1,2,3, \dots, c \qquad (1)$$

### *B. Attention Dynamic Convolution Block*

The ADC block is partially inspired by EEGNet [9] and dynamic convolution (Dy-conv) [30]. The ADC block comprises three conv layers: temporal (dynamic) convolution, spatial (depthwise) convolution, and spatiotemporal (dynamic) convolution. The first conv layer is followed by two SE attention blocks over the spectral and spatial (channel) dimensions of the EEG signal, as illustrated in Fig. 2. Attention blocks along with dynamic convolution allow the proposed model to attend to the most valuable features at the kernel level as well as the features level in multiple dimensions (i.e., temporal, spatial, and spectral). This multilevel attention is described in the next section.

Except for the multilevel attention, the ADC block has similar architecture and kernel parameters as the convolutional block explained in our previous work [15]. The architecture details of the ADC block are depicted in Fig. 2. Given an MI trial $X_i \in \mathbb{R}^{C\times T}$, the first conv layer in the ADC block performs a temporal convolution to learn spectral features on $X_i$ at different frequency bands [9]. This conv layer outputs a Tensor $S_{i,1} \in \mathbb{R}^{C\times T\times F_1}$ consisting of $F_1$ feature maps each representing a specific frequency band. The second conv layer uses depthwise convolution to learn spatial features on each feature map separately. Thus, each depthwise kernel extracts spatial information from a specific frequency band. Spatial depthwise conv outputs a temporal sequence $S_{i,2} \in \mathbb{R}^{T\times DF_1}$, where $D$ denotes the number of spatial kernels connected to each feature map. We empirically set $D = 2$. Finally, the spatio-temporal convolution learns how to optimally blend spatial and temporal features together and outputs a sequence of high-level spatio-temporal representation $S_{i,3} \in \mathbb{R}^{T_1\times F_2}$. This conv layer is computed along the temporal direction $T$, yet it performs spatio-temporal convolution because each timepoint in $S_{i,2}$ embeds spatial information from the previous spatial convolution.

Similar to our previous work [15], we used batch normalization (BN) [34], exponential linear unit (ELU), and

average pooling after each of the second and third conv layers to enhance trainability, add nonlinearity, and reduce dimensionality, respectively.

The output of the ADC block is a temporal sequence $z_i \in \mathbb{R}^{T_c \times d}$ consisting of $T_c$ temporal vectors each of length $d$. For simplicity, we set $d = DF_1 = F_2 = 32$.

### 1. Multilevel attention

The attention mechanism is usually implemented using attention scores. Attention scores can be calculated using different methods including additive attention [23] and multiplicative attention [24]. These scores are used by different approaches, such as SE [27], CBAM [26], MSA [25], and MLSA [32], on different dimensions of the EEG data, e.g., temporal, spectral, or spatial dimensions. This research employs multiplicative attention using MLSA and SE on the three dimensions of the EEG signal.

D-ATCNet implements multilevel attention at both the features level and the kernel level. At the features level, attention scores are generated from the input data (feature maps) and used to highlight important features in the same input data. In contrast, in kernel-level attention, attention scores are generated from the input data but are used to highlight the most valuable kernels in the CNN layer. Thus, the CNN kernel is selected dynamically based on the input data.

Kernel-level attention is implemented using dynamic convolution as described in the next section. Features level attention is employed in the temporal, spatial, and spectral dimensions of the EEG data. Spectral and spatial attention are employed earlier in the ADC block using the SE approach while temporal attention is employed later in the ATC block using the MLSA approach. MLSA temporal attention is described in Section C.2.

Spectral and spatial attention is performed after the first conv layer in the ADC block, as shown in Fig. 1 and 2. As previously shown, the first conv layer in the ADC block outputs a 3D tensor $S_{i,1} \in \mathbb{R}^{C \times T \times F_1}$ representing the channel (spatial) and spectral information of the EEG data in dimensions $C$ and $F_1$, respectively. Spectral attention is first performed along the $F_1$ dimension to highlight valuable spectral features in the MI-EEG data. The second attention is applied to the EEG channels $(C)$ to highlight valuable channels in the MI-EEG data. For both spectral and spatial attention, we used SE as defined in [27] and described in the next section.

### 2. Dynamic convolution

Dynamic convolution [29], [30] is a type of convolutional operation that allows the convolutional kernel to dynamically adapt its parameters to better fit the features of the input data. It uses attention over several convolutional kernels in the same layer. This allows the convolution parameters of each layer to be changed dynamically with different inputs during inference, rather than having the parameters fixed for all inputs in a traditional convolution.

Dynamic convolution aims to strike a balance between computational efficiency and network performance. To improve network performance, traditional approaches tend to increase computational costs, i.e., designing deeper or wider neural networks, which is not an efficient network design. Dynamic convolution, on the other hand, does not increase the width and depth of the DL model but instead uses attention to aggregate multiple convolutional kernels. This improves the representation capability of the network and thus its performance.

We adopt dynamic convolution as described in [30]. Traditional convolutions have a single static kernel per layer $(W, b)$, where $W$ and $b$ are the trainable parameters: weights and biases, respectively. Dynamic convolution uses $K$ parallel convolution kernels $(\widetilde{W}_k, \tilde{b}_k)$ that are aggregated dynamically for each input $x$, as defined in Eq. 2, where $\pi_k$ represents the attention weights (attention scores) that vary with input $x$.

$$\widetilde{W} = \sum_{k=1}^{K} \pi_k(x)\widetilde{W}_k\,, \qquad \tilde{b} = \sum_{k=1}^{K} \pi_k(x)\tilde{b}_k, \qquad 0 \le \pi_k(x) \le 1, \sum_{k=1}^{K} \pi_k(x) = 1 \tag{2}$$

Attention weights $\pi_k$ are calculated using the SE method [27]. The original SE network applied attention weights to the input channels, while in dynamic convolution, they are applied to the convolutional kernels. The SE method first squeezes the spatial information of the input data using a global average pooling layer. Next, attention weights are generated by passing the input throw two FC layers. The first FC layer is connected to a rectified linear unit (ReLU) activation for nonlinearity and the second FC layer is connected to SoftMax activation to normalize attention weights, as shown in Fig. 3. SoftMax is computed using a temperature $\tau_d$ to reduce the variation of attention weights, as in Eq. 3, where $m_k$ is the output of the second FC layer. The larger the temperature the more uniform attention scores. This allows all kernels in each layer to be optimized well, especially in the early training epochs. The temperature can be set to a fixed value or change dynamically during training, where $\tau_d$ being set to a large value in the early training epochs and decreasing in the later epochs. Both static and dynamic temperatures give good results [30]. The temperature can also be set as a trainable parameter. We investigate different settings of $\tau_d$ in the Experiments section.

$$\pi_k = \frac{\exp(m_k/\tau_d)}{\sum_i^K \exp(m_j/\tau_d)} \tag{3}$$

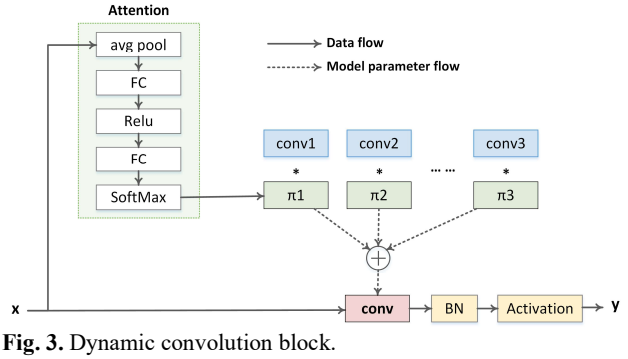


**Fig. 3.** Dynamic convolution block.

After the attention weights $\pi_k$ are generated, the aggregated convolution is computed, as shown in Fig. 3. The aggregation output can be fed into a batch normalization layer followed by an activation function, as a typical CNN design. We implemented dynamic convolution in the first and third conv layers of the ADC block, as shown in Fig. 2.

### C. Attention Temporal Convolution Block

The ATC block consists of an MLSA block followed by a temporal convolutional network. MLSA first highlights the valuable temporal features in the temporal sequence $z_i$, generated from the ADC block, and then the TCN extracts high-level temporal features from the highlighted sequence. Here we have two options, either input the entire $z_i$ sequence into the ATC block, or split $z_i$ into local windows using a shifted window. Using local windows as inputs to parallel ATC blocks enhances the performance of D-ATCNet, as will be shown later in the Experiments section. SW, MLSA, and TCN are detailed in the following sections.

#### 1. Shifted window

The Shifted window (sliding window) divides the temporal sequence $z_i$ into local sequences $z_i^w \in \mathbb{R}^{T_w \times d}$, which helps to extract the local features separately. We used a shifted window of size $T_w = T_c - 5$ with one element stride, i.e., $z_i$ is divided into 5 local windows. For different configurations on the shifted window and detailed discussion, the reader can refer to the study in [15].

#### 2. Multi-head locality self-attention

The MLSA, proposed in [32], differs from the original MSA [25] by introducing a diagonal mask and a learnable temperature. The MLSA layer consists of multiple self-attention heads that perform scaled dot-product attention [25]. The three primary parts of each attention head are the query $Q$, the keys $K$, and the values $V$. The query interacts with the keys to generate attention scores that highlight the valuable elements in the values, as depicted in Fig. 4. Below is a description of how this interaction is carried out in detail.

First, the query/key/value vectors are linearly projected from the local window $z_i^w$ as:

$$q_t^h = W_Q^h \, LN(z_{i,t}^w) \quad \in \mathbb{R}^{d_H}, \quad W_Q^h \in \mathbb{R}^{d \times d_H} \tag{4}$$

$$k_t^h = W_K^h \, LN(z_{i,t}^w) \quad \in \mathbb{R}^{d_H}, \quad W_K^h \in \mathbb{R}^{d \times d_H} \tag{5}$$

$$v_t^h = W_V^h \, LN(z_{i,t}^w) \quad \in \mathbb{R}^{d_H}, \quad W_V^h \in \mathbb{R}^{d \times d_H} \tag{6}$$

where $t = 1, \dots, T_w$ is an index over the elements of the local window $z_i^w$ and $T_w$ is the total number of elements in the window (the length of the window), $h = 1, \dots, H$ represents the heads index, and $H$ is the number of attention heads. LN refers to Layer Normalization [35]. The head dimension is set experimentally to $d_H = d/2H$.

Second, the context vector for each head is computed by multiplying the attention scores by the values $V$ as follows. Given a minibatch with $n$ queries $Q \in \mathbb{R}^{n \times d_H}$ and $m$ key-value pairs ($K \in \mathbb{R}^{m \times d_H}, V \in \mathbb{R}^{m \times v}$), the context vector for each head $C^h$ is computed by Eq. 7, where $f$ is a diagonal mask function, which set the diagonal values of alignment scores to $-\infty$, and $\tau_h$ is a learnable temperature. In this implementation, we set $n = m = T_w$ and $v = d_H = 8$.

$$C^h = \text{SoftMax}\left(f\left(\frac{Q^h (K^h)^T}{\sqrt{d_H}}\right)/\tau_h\right) V^h \in \mathbb{R}^{n \times v} \tag{7}$$

where $Q^h \in \mathbb{R}^{n \times d_H}$, $K^h \in \mathbb{R}^{m \times d_H}$, and $V^h \in \mathbb{R}^{m \times v}$

After that, the multi-head self-attention is obtained by concatenating the context vectors of all heads and projecting the results linearly, then adding them to the input sequence $z_i^w$ as in Eq. 8.

$$z_i^w = W_O \, [C^1, \dots, C^H] + z_i^w \in \mathbb{R}^{T_w \times d}, \quad W_O \in \mathbb{R}^{d_H \times d} \tag{8}$$

#### 3. Temporal Convolutional Network

The TCN architecture is similar to the TC block described in [15] with the same hyperparameters. The only change is that we examined the effect of replacing the 1D conv layers with Dy-conv. The effect of this change is discussed in the Experiments section.

TCN is made up of a sequence of residual blocks. Each block consists of two causal dilated conv layers linked to a BN [34] and exponential linear unit (ELU), as illustrated in Fig. 5. For more details about the TCN architecture, the reader can refer to [15].

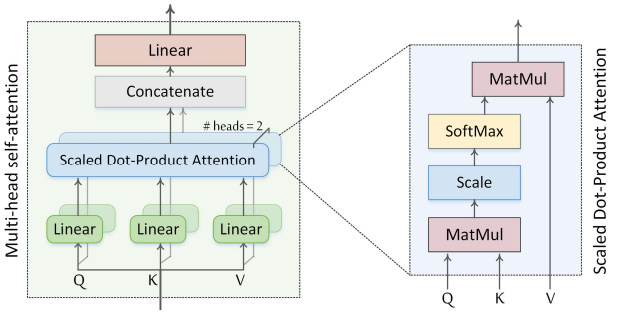


**Fig. 4.** Multi-head self-attention.

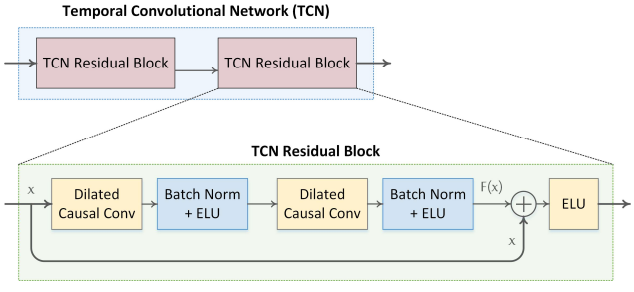


**Fig .5.** The temporal convolutional network.

Fig. 6 depicts 16 temporal components ($T_w$=16) entering the temporal convolutional network. Each component is a vector of size $F_2$ (equal to the number of kernels in the last conv layer in the ADC block). The TCN outputs the last component in the sequence, which is a vector of length $F_T$ (equal to the number of kernels in the TCN). In our experiments, we set $F_2 = F_T = 32$.

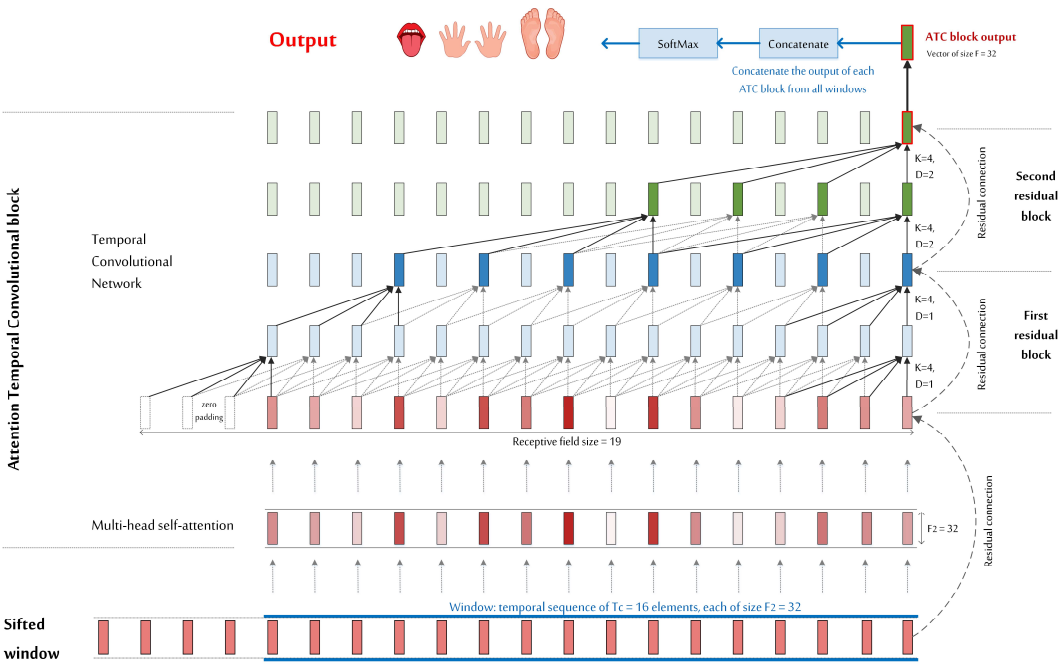


**Fig. 6.** Visualization of feature maps in the attention temporal convolution (ATC) block using an window of 16 elements ($T_w$=16).

## III. Experiments and Discussion

### A. Dataset

We evaluate the D-ATCNet model on the BCI-IV2a dataset [31]. BCI-IV2a is a public well-known EEG-based MI dataset produced in 2008 by Graz University. BCI-IV2a is a benchmark dataset for EEG-based motor imagery classification [1]. Decoding MI tasks using this dataset is challenging due to the limited number of MI trials performed in an uncontrolled environment with significant artifacts.

The BCI-IV2a dataset consists of a total of 5184 MI trials belonging to 4 MI tasks. Each trial is eight seconds long and MI activities were performed during the four seconds in the middle. The dataset was recorded by 9 subjects using 22 EEG sensors in two sessions. The model is trained using one of the two sessions and its performance is evaluated using the other.

### B. Evaluation Method and Performance Metrics

Both subject-dependent and subject-independent methods are used in evaluating the proposed model. We used original competition training and testing data for the subject-dependent evaluation. Specifically, we used 9 x 288 trials during session 1 to train the models, and then 9 x 288 trials during session 2 for evaluation. For the subject-independent evaluation, we used the 'leaving one subject out' (LOSO) assessment method [1].

Kappa score and accuracy metrics are used to evaluate the performance of the proposed and reproduced models as defined in [15].

All training and evaluation experiments in this study were run on one of the following GPUs: GTX 1080 Ti 12GB or 2070 8GB, using the TensorFlow framework with Python 3.7. We used the same training configurations as specified in our previous work [15]. Detailed configurations can be found in the code published on GitHub [1].

The D-ATCNet model outperforms the current state-of-the-art findings on the BCI-IV2a dataset with a κ-score of 0.828 and an accuracy of 87.08%.

### C. Analysis of dynamic convolution

This section first analyses the effect of using dynamic convolution on 2D conv layers (in the ADC block) and 1D conv layers (in the ATC block), using default parameters, i.e., $K = 3$ kernels per layer and $\tau_d = 30$. Then we analyze the best configuration of these parameters for MI-EEG data.

Table 1 compares the performance of the proposed model in four scenarios: without using Dy-conv, using 2D Dy-conv in ADC block only, using 1D Dy-conv in ATC block only, and using Dy-conv in both ATC and ADC blocks. In all cases, Dy-conv enhanced the learning ability of the model. Using Dy-conv in either ADC or ATC blocks performed better than using Dy-conv in both blocks. Although the model has more representation power by using dynamic convolution at all conv layers, this increases the complexity of the model and makes the network more prone to overfitting. The best performance was achieved by implementing 2D Dy-conv in the ADC block. This indicates that dynamic convolution is more effective on 2D kernels in earlier layers than on 1D kernels in deeper layers.

In the following experiments, we used Dy-conv in the ADC block and the standard static conv in the ATC block.

The hyperparameter $K$ determines the complexity of the model. Dynamic convolution performs better than its static version, even with a low value of $K = 2$. However, the accuracy stops improving beyond $K = 3$ due to the increased representational power of the model that comes with higher values of $K$. In this case, optimizing all convolutional and attention parameters simultaneously becomes more challenging, and overfitting becomes more likely as $K$ increases.

The temperature $\tau_d$ of the SoftMax in dynamic convolution determines the sparsity of the attention weights. This is a critical

element for the successful training of Dy-conv. Fig. 7 shows the classification accuracy for different temperatures, with $\tau_d = 30$ being the most effective.

### *D. Ablation analysis on multilevel attention*

We performed an ablation analysis to evaluate the performance of multilevel attention. The effect of deleting attention layers from D-ATCNet at the kernel and input features levels is shown in Table 2. Attention layers were removed before the training and testing procedures. According to the results, kernel level attention using Dy-conv improved the overall accuracy by 0.97% and features level attention, including temporal, spectral, and channel attention, by 1.43%. The results show that both levels of attention, across the kernels and input features, can collaborate and individually enhance the performance of D-ATCNet. Table 2 also shows the impact of applying attention to the entire temporal sequence (before the shifted window) rather than applying it to each shifted window. Results show that using separate temporal attention on each shifted window improves accuracy by 1.36%. This is because shifted window attention helps to highlight local features separately in each window.

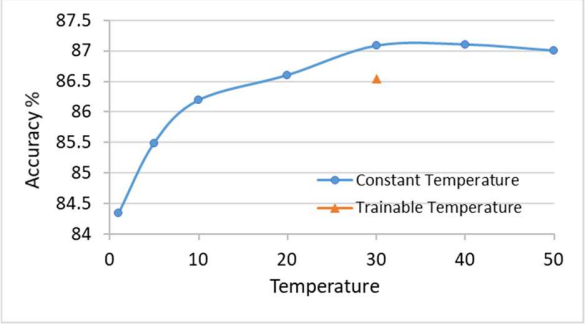


**Fig .7.** Performance of D-ATCNet as a function of SoftMax temperature $\tau_d$ in dynamic convolution. A lower temperature results in more sparse attention scores, while a higher temperature leads to less sparsity and better performance. The best performance is achieved at $\tau_d = 30$, and the performance remains stable as the temperature increases.

Table 1. Performance comparison of the D-ATCNet model using 2D dynamic convolution (2D Dy-conv) in the ADC block and/or (1D Dy-conv) in the ATC block with the BCI-IV2a dataset.

| Method | Accuracy % | κ-score |
|---|---|---|
| without Dy-conv | 86.11 | 0.815 |
| 2D Dy-conv in the ADC block | **87.08** | **0.828** |
| 1D Dy-conv in the ATC block | 86.57 | 0.821 |
| 1D and 2D Dy-conv in both blocks | 86.50 | 0.820 |

Table 2. Ablation analysis of the multilevel in the D-ATCNet model. TC: temporal attention, SA: Spectral attention, CA: channel attention, DC: dynamic convolution, SW: shifted window. In shifted window attention (SWA), we apply temporal attention (MLSA) to each shifted window $z_i^w$ separately rather than to the entire temporal sequence $z_i$.

| Removed attention | Accuracy % | κ-score |
|---|---|---|
| none (D-ATCNet) | 87.08 | 0.828 |
| kernel level attention (DC) | 86.11 | 0.815 |
| features level attention (TA, SA, CA) | 85.65 | 0.809 |
| multilevel attention (DC, TA, SA, CA) | 85.46 | 0.806 |
| shifted window attention (SWA) | 85.72 | 0.809 |

### *E. Comparison to recent studies*

In this section, we compare the performance of the D-ATCNet for each subject in the BCI-IV2a dataset with other reproduced models: TCNet_Fusion [22], EEG-TCNet [21], EEGNet [9], and ATCNet [15]. The results of these models rely on the parameters defined in the original articles, while the training, evaluation, and preprocessing were conducted following the methodology specified in this research. Table 3 shows the best and average performance for each model based on 10 random runs. The results in Table 3 show that D-ATCNet performed better than the other models for all subjects, achieving an overall κ score of 0.83 and an accuracy of 87.1%. The average performance of the D-ATCNet model over 10 runs, i.e., an average of 90 separate and random runs (9 subjects x 10 runs), is also higher than other models. This indicates that the proposed model has strong learning ability and can reproduce similar stable results over several runs. D-ATCNet also achieves the lowest standard deviation over subjects, demonstrating that its performance is more consistent across different subjects.

Fig. 8 shows the confusion matrices for D-ATCNet and reproduced networks. Compared with other models, D-ATCNet showed an increase in MI classification performance for all MI activities.

In Table 4, we compare recent studies in MI decoding according to the preprocessing technique, input formulation, network architecture, and network performance in both subject-dependent and subject-independent approaches. The performance of the proposed network is superior to that of previous research using raw MI-EEG signals and without preprocessing. D-ATCNet achieves a distinguished performance in both the subject-dependent and subject-independent evaluation with the ability to generalize well on new subjects.

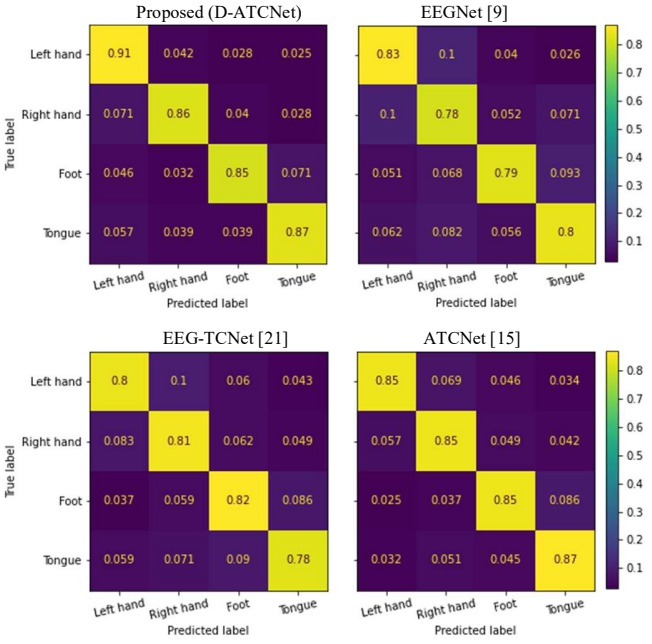


**Fig. 8.** Confusion matrices for the proposed and reproduced models.

## IV. Conclusion

A new attention-based dynamic convolutional network (D-ATCNet) was proposed in this paper for the decoding of MI activities based on EEG signals. D-ATCNet comprises two main blocks: the attention dynamic convolution (ADC) block and the attention temporal convolution (ATC) block. The ADC block extracts low-level spatiotemporal features from the MI-EEG data using three conv layers with multilevel attention. Multilevel attention was implemented at the kernel level using dynamic convolution and at the features level using MLSA and SE. Features-level attention was investigated over the temporal, spatial, and spectral dimensions of the EEG signal. The ATC block then extracts high-level temporal information using shifted window and self-attention. The results of the ablation study demonstrated the significant contributions made by dynamic convolution and multilevel attention to the overall network performance. The proposed model outperformed current DL architectures in decoding MI-EEG signals with an accuracy of 71.3% for subject-independent and 87.08% for subject-dependent using the BCI-IV2a benchmark dataset. D-ATCNet demonstrated a strong ability to detect MI activities from raw brain signals without removing artifacts and with little preprocessing while training on a small and challenging dataset. D-ATCNet improved the performance of MI-EEG classification for all subjects and all MI activities in the BCI-IV2a dataset, demonstrating its ability to detect generic MI features across various subjects and classes. The D-ATCNet architecture is suitable for IoT and edge devices due to its high performance and a relatively small number of parameters (150 k).

Table 3. Subject-specific performance comparison between D-ATCNet and reproduced DL networks using the BCI-IV2a dataset.

| Sub. | Proposed (D-ATCNet) | | | | ATCNet [15] | | | | EEGNet [9] | | | | EEG-TCNet [21] | | | | TCNet_Fusion [22] | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | best | | average | | best | | average | | best | | average | | best | | average | | best | | average | |
| | % | κ | % | κ | % | κ | % | κ | % | κ | % | κ | % | κ | % | κ | % | κ | % | κ |
| 1 | 88.9 | 0.85 | 87.5 | 0.83 | 89.6 | 0.86 | 87.3 | 0.83 | 88.9 | 0.85 | 86.9 | 0.83 | 85.8 | 0.81 | 83.0 | 0.77 | 88.2 | 0.84 | 83.2 | 0.78 |
| 2 | 72.6 | 0.63 | 70.0 | 0.60 | 72.2 | 0.63 | 68.9 | 0.59 | 66.3 | 0.55 | 62.6 | 0.50 | 63.9 | 0.52 | 58.8 | 0.45 | 66.0 | 0.55 | 63.1 | 0.51 |
| 3 | 97.9 | 0.97 | 94.9 | 0.93 | 97.2 | 0.96 | 95.8 | 0.94 | 95.1 | 0.94 | 91.7 | 0.89 | 95.1 | 0.94 | 92.6 | 0.90 | 92.4 | 0.90 | 91.2 | 0.88 |
| 4 | 86.1 | 0.81 | 80.5 | 0.74 | 78.5 | 0.71 | 76.8 | 0.69 | 68.8 | 0.58 | 64.7 | 0.53 | 72.6 | 0.63 | 67.7 | 0.57 | 73.6 | 0.65 | 70.9 | 0.61 |
| 5 | 83.0 | 0.77 | 79.5 | 0.73 | 82.6 | 0.77 | 79.5 | 0.73 | 76.0 | 0.68 | 71.6 | 0.62 | 78.5 | 0.71 | 73.0 | 0.64 | 78.5 | 0.71 | 74.8 | 0.66 |
| 6 | 78.5 | 0.71 | 74.4 | 0.66 | 77.1 | 0.69 | 73.0 | 0.64 | 63.9 | 0.52 | 60.3 | 0.47 | 63.2 | 0.51 | 60.1 | 0.47 | 65.6 | 0.54 | 63.4 | 0.51 |
| 7 | 95.5 | 0.94 | 93.2 | 0.91 | 93.8 | 0.92 | 91.6 | 0.89 | 90.6 | 0.88 | 88.9 | 0.85 | 91.0 | 0.88 | 86.0 | 0.81 | 90.3 | 0.87 | 89.0 | 0.85 |
| 8 | 89.2 | 0.86 | 87.6 | 0.83 | 89.6 | 0.86 | 87.7 | 0.84 | 87.2 | 0.83 | 85.0 | 0.80 | 85.4 | 0.81 | 81.8 | 0.76 | 85.8 | 0.81 | 84.3 | 0.79 |
| 9 | 92.0 | 0.89 | 89.6 | 0.86 | 90.3 | 0.87 | 88.8 | 0.85 | 84.0 | 0.79 | 80.0 | 0.73 | 84.7 | 0.80 | 81.4 | 0.75 | 87.9 | 0.84 | 83.6 | 0.78 |
| Mean | **87.1** | **0.83** | **84.1** | **0.79** | 85.6 | 0.81 | 83.3 | 0.78 | 80.1 | 0.73 | 76.9 | 0.69 | 80.0 | 0.73 | 76.1 | 0.68 | 80.9 | 0.75 | 78.2 | 0.71 |
| St. D. | **8.08** | **0.11** | **8.53** | **0.11** | 8.41 | 0.11 | 9.09 | 0.12 | 11.6 | 0.15 | 12.2 | 0.16 | 11.4 | 0.15 | 11.8 | 0.16 | 10.4 | 0.14 | 10.6 | 0.14 |

Table 4. Subject-dependent and subject-independent performance comparison between D-ATCNet and recent studies using the BCI-IV2a dataset. In all studies, the original BCI-IV2a competition division (Session 1 for training and Session 2 for testing) was used for subject-dependent evaluation and leave-one-subject-out (LOSO) cross-validation was used for subject-independent evaluation. The average κ score and accuracy (%) of all subjects is presented.

| Study *(* Reproduced)* | Description | | | Performance | | | |
|---|---|---|---|---|---|---|---|
| | Pre-processing [1] | Input data [2] | DL approach | Subject-dependent | | Subject-independent | |
| | | | | % | κ | % | κ |
| Schirrmeister et al 2017 [33] * | FB: 4-$f_{end}$ | RS | CNN (DeepConvNet) | 75.4 | 0.67 | 70.4 | 0.60 |
| Lawhern et al. 2018, [9]* | FB: 8-35 | RS | CNN (EEGNet) | 80.1 | 0.73 | 68.8 | 0.58 |
| Hassanpour et al. 2019, [18] | FB: 8-35 Hz, AR: SWT | SF | DBN-AE | 71.0 | - | - | - |
| Amin et al. 2019, [14] | FB: 0.5-40 Hz | RS | Multi-layer-CNN, MLP | 75.0 | - | 55.3 | - |
| Ingolfsson et al. 2020, [21] * | AR: manual | RS | CNN + TCN (EEG-TCNet) | 80.0 | 0.73 | 69.5 | 0.59 |
| Zhang et al. 2020, [5] | no preprocessing | TM | Attention, graph CNN, LSTM | - | - | 60.1 | - |
| Musallam et al. 2021, [22] * | AR: manual | RS | Multi-layer CNN, TCN (TCNet_Fusion) | 80.9 | 0.75 | 70.6 | 0.61 |
| Amin et al. 2022, [7] | FB: 8-35 | RS | Attention, inception CNN, LSTM | 82.8 | - | - | - |
| Altuwaijri et al. 2022 [6] * | no preprocessing | RS | Attention, multi-branch CNN | 82.2 | 0.76 | 68.7 | 0.58 |
| Altuwaijri et al. 2022 [13] | no preprocessing | RS | Attention, multi-branch CNN | 83.7 | 0.78 | 68.0 | - |
| Altaheri et al. 2023 [15] * | no preprocessing | RS | Attention, CNN, TCN (ATCNet) | 85.7 | 0.81 | 70.9 | 0.61 |
| Proposed method | no preprocessing | RS | Dy. CNN, Attention, TCN (D-ATCNet) | **87.1** | **0.83** | **71.3** | **0.62** |

[1] *This table defines preprocessing as three operations: artifact removal (**AR**), channel selection (**CS**), and signal filtering (**SF**). **SWT**: Synchrosqueezed wavelet transforms.*
[2] *This table identifies three types of input formulation: Raw signal (**RS**), Spectral features (**SF**), and Topological maps (**TM**).*

## REFERENCES


[1] H. Altaheri *et al.*, "Deep learning techniques for classification of electroencephalogram (EEG) motor imagery (MI) signals: a review," *Neural Comput. Appl.*, vol. 33, pp. 1–42, 2021.

[2] H. Altaheri, M. Alsulaiman, and G. Muhammad, "Date Fruit Classification for Robotic Harvesting in a Natural Environment Using Deep Learning," *IEEE Access*, vol. 7, no. 1, pp. 117115–117133, Aug. 2019.

[3] M. Qamhan, H. Altaheri, A. H. Meftah, G. Muhammad, and Y. A. Alotaibi, "Digital Audio Forensics: Microphone and Environment Classification Using Deep Learning," *IEEE Access*, vol. 9, pp. 62719–62733, 2021.

[4] M. Al-Hammadi *et al.*, "Spatial Attention-Based 3D Graph Convolutional Neural Network for Sign Language Recognition," *Sensors*, vol. 22, no. 12, p. 4558, 2022.

[5] D. Zhang, K. Chen, D. Jian, and L. Yao, "Motor imagery classification via temporal attention cues of graph embedded EEG signals," *IEEE J. Biomed. Heal. informatics*, vol. 24, no. 9, pp. 2570–2579, 2020.

[6] G. A. Altuwaijri, G. Muhammad, H. Altaheri, and M. Alsulaiman, "A Multi-Branch Convolutional Neural Network with Squeeze-and-Excitation Attention Blocks for EEG-Based Motor Imagery Signals Classification," *Diagnostics*, vol. 12, no. 4, pp. 1–16, 2022.

[7] S. U. Amin, H. Altaheri, G. Muhammad, M. Alsulaiman, and A. Wadood, "Attention-Inception and Long Short-Term Memory-based Electroencephalography Classification for Motor Imagery Tasks in Rehabilitation," *IEEE Trans. Ind. Informatics*, 2022.

[8] S. U. Amin, H. Altaheri, G. Muhammad, M. Alsulaiman, and W. Abdul, "Attention based Inception model for robust EEG motor imagery classification," in *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2021, pp. 1–6.

[9] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, p. 56013, 2018.

[10] D. Li, J. Xu, J. Wang, X. Fang, and J. Ying, "A Multi-Scale Fusion Convolutional Neural Network based on Attention Mechanism for the Visualization Analysis of EEG Signals Decoding," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 12, pp. 2615–2626, 2020.

[11] M.-A. Li, J.-F. Han, and L.-J. Duan, "A Novel MI-EEG Imaging With the Location Information of Electrodes," *IEEE Access*, vol. 8, pp. 3197–3211, 2019.

[12] T. Liu and D. Yang, "A Densely Connected Multi-Branch 3D Convolutional Neural Network for Motor Imagery EEG Decoding," *Brain Sci.*, vol. 11, no. 2, pp. 1–24, 2021.

[13] G. A. Altuwaijri and G. Muhammad, "Electroencephalogram-Based Motor Imagery Signals Classification Using a Multi-Branch Convolutional Neural Network Model with Attention Blocks," *Bioengineering*, vol. 9, no. 7, p. 323, 2022.

[14] S. U. Amin, M. Alsulaiman, G. Muhammad, M. A. Mekhtiche, and M. S. Hossain, "Deep Learning for EEG motor imagery classification based on multi-layer CNNs feature fusion," *Futur. Gener. Comput. Syst.*, vol. 101, pp. 542–554, 2019.

[15] H. Altaheri, G. Muhammad, and M. Alsulaiman, "Physics-Informed Attention Temporal Convolutional Network for EEG-Based Motor Imagery Classification," *IEEE Trans. Ind. Informatics*, vol. 19, no. 2, pp. 2249–2258, 2023.

[16] S. Kumar, R. Sharma, and A. Sharma, "OPTICAL+: a frequency-based deep learning scheme for recognizing brain wave signals," *PeerJ Comput. Sci.*, vol. 7, p. e375, 2021.

[17] T. Luo and F. Chao, "Exploring spatial-frequency-sequential relationships for motor imagery classification with recurrent neural network," *BMC Bioinformatics*, vol. 19, no. 1, pp. 1–18, 2018.

[18] A. Hassanpour, M. Moradikia, H. Adeli, S. R. Khayami, and P. Shamsinejadbabaki, "A novel end-to-end deep learning scheme for classifying multi-class motor imagery electroencephalography signals," *Expert Syst.*, vol. 36, no. 6, p. e12494, 2019.

[19] J. Xu, H. Zheng, J. Wang, D. Li, and X. Fang, "Recognition of EEG signal motor imagery intention based on deep multi-view feature learning," *Sensors*, vol. 20, no. 12, pp. 1–16, 2020.

[20] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv Prepr. arXiv1803.01271*, 2018.

[21] T. M. Ingolfsson, M. Hersche, X. Wang, N. Kobayashi, L. Cavigelli, and L. Benini, "EEG-TCNet: An Accurate Temporal Convolutional Network for Embedded Motor-Imagery Brain-Machine Interfaces," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 2958–2965.

[22] Y. K. Musallam *et al.*, "Electroencephalography-based motor imagery classification using temporal convolutional network fusion," *Biomed. Signal Process. Control*, vol. 69, p. 102826, 2021.

[23] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv Prepr. arXiv1409.0473*, 2014.

[24] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412--1421.

[25] A. Vaswani *et al.*, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[26] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[27] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[28] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv Prepr. arXiv2010.11929*, 2020.

[29] B. Yang, G. Bender, Q. V Le, and J. Ngiam, "Condconv: Conditionally parameterized convolutions for efficient inference," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[30] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11030–11039.

[31] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "BCI Competition 2008–Graz data set A," *Inst. Knowl. Discov. Graz Univ. Technol.*, vol. 16, pp. 1–6, 2008.

[32] S. H. Lee, S. Lee, and B. C. Song, "Vision transformer for small-size datasets," *arXiv Prepr. arXiv2112.13492*, 2021.

[33] R. T. Schirrmeister *et al.*, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapp.*, vol. 38, no. 11, pp. 5391–5420, 2017.

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456.

[35] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv Prepr. arXiv1607.06450*, 2016.