

# EEG-TCNet: An Accurate Temporal Convolutional Network for Embedded Motor-Imagery Brain–Machine Interfaces

Thorir Mar Ingolfsson\*, Michael Hersche\*, Xiaying Wang\*, Nobuaki Kobayashi<sup>†</sup>, Lukas Cavigelli<sup>\*‡</sup>, Luca Benini\*

\*ETH Zurich, Dept. EE & IT, Switzerland

<sup>†</sup>Nihon University, College of Science and Technology, Japan

<sup>‡</sup>Huawei Technologies, Zurich Research Center, Switzerland

**Abstract**—In recent years, deep learning (DL) has contributed significantly to the improvement of motor-imagery brain-machine interfaces (MI-BMIs) based on electroencephalography (EEG). While achieving high classification accuracy, DL models have also grown in size, requiring a vast amount of memory and computational resources. This poses a major challenge to an embedded BMI solution that guarantees user privacy, reduced latency, and low power consumption by processing the data locally. In this paper, we propose EEG-TCNet, a novel temporal convolutional network (TCN) that achieves outstanding accuracy while requiring few trainable parameters. Its low memory footprint and low computational complexity for inference make it suitable for embedded classification on resource-limited devices at the edge. Experimental results on the BCI Competition IV-2a dataset show that EEG-TCNet achieves 77.35% classification accuracy in 4-class MI. By finding the optimal network hyperparameters per subject, we further improve the accuracy to 83.84%. Finally, we demonstrate the versatility of EEG-TCNet on the Mother of All BCI Benchmarks (MOABB), a large scale test benchmark containing 12 different EEG datasets with MI experiments. The results indicate that EEG-TCNet successfully generalizes beyond one single dataset, outperforming the current state-of-the-art (SoA) on MOABB by a meta-effect of 0.25.

**Index Terms**—brain-machine interface, motor-imagery, deep learning, convolutional neural networks, edge computing.

## I. INTRODUCTION

Brain-machine interfaces (BMIs) allow direct communication between humans and external devices by analyzing neural activity of the human brain, typically recorded with noninvasive electroencephalography (EEG) [1]. One promising approach is based on motor-imagery (MI), which is the cognitive process of thinking about the motion of a body part, e.g., the left hand, without actually performing it. MI-BMIs assist people with impairments to regain independence, e.g., by steering a wheelchair [2], controlling a prosthesis [3], [4], or by enabling motor rehabilitation [5].

However, successful decoding of MI-based EEG signals remains a challenging task, mainly due to a low signal-to-noise ratio and high variance among different subjects, which prohibits the use of a single MI-BMI model for all subjects [6]. Conventional approaches rely on domain-specific knowledge, mostly using handcrafted feature extractors, such as filter

bank common spatial pattern (FBCSP) [7] or Riemannian covariance [8] features in combination with robust classifiers like linear discriminant analysis (LDA) or support vector machines (SVMs) [6].

Recently, convolutional neural networks (CNNs) have gained increasing attention in the MI-BMI field, reducing the data pre-processing steps and eliminating the procedure of handcrafting features. One of the first successful CNN in MI classification was Shallow ConvNet [9], which was inspired by FBCSP. The more compact and generally applicable EEGNet [10], as well as more complex and accurate models [11], [12], have extended the landscape of CNNs in MI classification. The most complex network is TPCT [13], which achieves the state-of-the-art (SoA) accuracy of 88.87% on the 4-class MI BCI Competition IV-2a dataset [14].

These networks are commonly deployed on desktop platforms or cloud servers, however, running MI classification on remote computers raises serious concerns in terms of latency, availability, and privacy [15], [16], [17]. Processing the data near the sensor on a low-power microcontroller unit (MCU) allows us to mitigate these concerns. However, accurate networks such as the TPCT model have 7.78 M trainable parameters and require 1.73 billion multiply-accumulate (MAC) operations per inference, which is out of reach of a typical low-power MCU with few MB of Flash and few hundreds of kB of RAM [18]. Alternatively, more compact models such as EEGNet with 2.5 k parameters and 13 MMACs can come to the rescue and have been successfully implemented on MCUs [19], [20]. Still, they come at the cost of significantly lower classification accuracy of 72.40%. A model that combines the best of both worlds (i.e., compactness at high accuracy) is highly desirable.

One viable option to boost the performance in accuracy is to use temporal convolutional networks (TCNs), which are achieving SoA accuracy on many time series classification and modeling tasks [21], [22]. TCNs are capable to exponentially extend their receptive field size with only a linear increase in the number of parameters and number of MACs, unlike traditional CNNs, which show only a linear increase in the receptive field size. Moreover, in contrast to other time series classification networks, like recurrent neural networks (RNNs),

TCNs do not suffer from exploding or vanishing gradient issues particularly, when training on long input sequences.

In this paper, we introduce EEG-TCNET, which features both the compactness of EEGNet and the high accuracy of TCNs. The main contributions of the paper are as follows:

- We propose EEG-TCNET, which requires only 4272 trainable parameters and 6.8 MMACs per inference, making it suitable for resource-limited embedded devices.
- We evaluate the proposed EEG-TCNET on the BCI Competition IV-2a dataset [14], where it achieves a high accuracy of 77.35%. Our model requires significantly fewer parameters, MACs, and memory during inference compared to other networks with similar accuracy.
- We further improve EEG-TCNET by 6.49% classification accuracy, reaching 83.84%, by finding the optimal network hyperparameters per subject based on a grid search in cross-validation on the training data. This outperforms most of the current SoA networks. We then analyze accuracy vs. parameter counts and accuracy vs. number of MAC operations for our proposed model, and obtain that EEG-TCNET achieves Pareto optimality in both cases.
- We extensively benchmark our methods on the Mother of All BCI Benchmarks (MOABB) [23], where EEG-TCNET outperforms the current SoA by a meta-effect of 0.25. To best of our knowledge, this work is the first external submission to MOABB, which will pave the way for reproducible results required to benchmark new BMI classifiers reliably.

Our code and the trained models are available online under a permissible open-source license for reproducibility<sup>1</sup>.

## II. BACKGROUND

### A. BCI Competition IV-2a Dataset

The BCI Competition IV-2a dataset [14] consists of recordings from nine different subjects using 22 EEG electrodes. The data was collected by bandpass filtering the signals between 0.5 Hz and 100 Hz and sampling them at 250 Hz. All subjects were requested to perform imagined movements of four different body parts: left hand, right hand, both feet, and tongue. Besides, three electrooculography (EOG) channels give information about eye movements. The dataset consists of two sessions per subject recorded on different days, where we use one for training and the other for testing, and each session contains 288 trials. Trials containing artifacts (9.41% of the data) were excluded from the dataset after an expert marked them based on EOG data. In order to go by the rules of the BCI Competition IV-2a, we make no further use of the EOG data. Each trial lasted 7.5 s and was recorded according to the timing scheme shown in Fig 1.

### B. Mother of All BCI Benchmarks

The Mother of All BCI Benchmarks (MOABB) [23] is an aggregation of many publicly available EEG datasets,

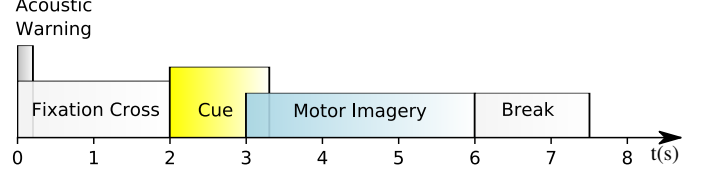


Fig. 1. Timing scheme of the BCI Competition IV-2a [14].

TABLE I  
MOABB DATASETS ATTRIBUTES.

Name	$N$	$C$	# Trials	$S$	# Subjects	Epoch (s)
Cho <i>et al</i>	2	64	200	1	52	0-3
Physionet	2	64	40-60	1	109	1-3
Shin <i>et al</i>	2	25	60	3	29	0-10
BNCI 2014-001	4	22	288	2	9	2-6
BNCI 2014-002	2	15	160	1	14	3-8
BNCI 2014-004	2	3	120-160	5	9	3-7.5
BNCI 2015-001	2	13	200	2/3	13	3-8
BNCI 2015-004	2	30	70-80	2	10	3-10
Alexandre	2	16	40	1	9	0-3
Yi <i>et al</i>	4	60	160	1	10	3-7
Grosse-Wentrup <i>et al</i>	2	128	300	1	10	3-10
Schirmer <i>et al</i>	4	128	260	1	14	0-4
Total:					288	

$N$  = number of classes,  $C$  = number of EEG channels,  $S$  = number of sessions.

converted to a common format, and bundled in a software package. It was initiated because of several problems currently present in the BMI research community. One of those is that while many BMI datasets are made freely available, researchers do not publish code, and reproducing results required to benchmark new algorithms turns out to be more tricky than it should be. Moreover, performance can be significantly impacted by parameters of the pre-processing steps, toolboxes used, and implementation tricks that are rarely reported in the literature.

MOABB aims to provide solutions to these problems by building a comprehensive benchmark of popular BMI algorithms applied on an extensive list of freely available EEG datasets. The code is available online on GitHub; algorithms can be ranked and promoted on a website, providing a clear picture of the different solutions available in the field. MOABB provides a variety of different datasets, both for MI and event-related potential (ERP) classification. In this paper, we consider all MI datasets consisting of 2–4 classes MI experiments. Due to internal errors in the current MOABB package, we had to replace the dataset of Zhou *et al.* 2016 for the high-gamma dataset described in [9]. The used MI datasets are summarized in Table II.

### C. Related Work

The BCI Competition IV-2a submissions [24] are various, but one frequent feature used by many submissions was common spatial patterns (CSP) on bandpass filtered data. Ang *et al.* [7], the winners of the competition, proposed filter bank common spatial pattern (FBCSP), which enhanced the performance of the original CSP algorithm and achieved 67.75% classification accuracy. After the competition, a linear

<sup>1</sup>Will be released on GitHub after review.

support vector machine (SVM) on Riemannian covariance matrices [8] has achieved an accuracy of 75.74%. More recent studies [25] have shown that FBCSP combined with highway networks, random forests, and multiple binary classifiers can further enhance the accuracy to 78.00%, 80.00%, and 81.02%, respectively.

We categorize deep learning-based MI classification into two classes: feature input (FI) networks and raw signal input (RSI) networks. The latter combine and train feature extraction and classification processes simultaneously. In particular, CNN-based RSI networks have been shown to achieve excellent results in MI-BMIs [6]. EEGNet [10] and Shallow ConvNet [9] have achieved high accuracy with a relatively small network size; EEGNet has 1716 trainable parameters at 66.70% accuracy, and Shallow ConvNet has 47 324 parameters at 74.31%. The main difference between the two architectures is that Shallow ConvNet was explicitly designed for oscillatory signal classification and hence utilizes log-band power to extract features, making it ill-suited for other similar tasks such as event-related potential (ERP) classification. In contrast, EEGNet is not only applicable to MI classification but also ERP tasks. By changing the pooling layers and expanding the network to 2036 trainable parameters, the accuracy of EEGNet has been increased to 72.40% [26].

Another competitive RSI-network is the MSFBCNN [27], which utilizes multi-scale temporal convolution to extract features and achieves an accuracy of 75.80%. CNN++ [12] uses not only the 22 EEG channels but also the 3 EOG channels, which in the original competition was strictly forbidden. Inspired by CSP, CNN++ starts with a linear layer applied to each time sample, expanding the 25 channels to 30. It is followed by a CNN, which finally achieves an accuracy of 81.1%.

In FI-networks, the MI classification is achieved in two stages. First, features are extracted from EEG signals with various approaches (e.g., CSP, spectrograms, or wavelets), and then fed into a classifier model. DFFN [11] makes use of CSP to extract unique spatial filters. Temporal log-power features of the spatially filtered signal are then fed into a CNN that considers the correlation between adjacent layers and cross-layer features. This architecture closely resembles *Dense Net* [28]. Network architecture hyperparameters were altered for each subject separately, achieving an accuracy of 79.71%.

TPCT [13] uses the information of electrode locations to improve classification results. The MI time frame was divided into 10 time windows and three sub-bands. Then, for each channel, the Fast Fourier Transform (FFT) was employed to transform each time window to a spectrum, and its inverse FFT was calculated for each sub-band. The time-domain power features of the 10 time windows are then averaged for the same sub-band. Hence, three average power features are generated as the time-frequency features of each electrode and fed into the Clough-Tocher interpolation algorithm to generate an image with electrode information that was then classified with a VGG-like CNN. TPCT achieved an accuracy of 88.87%, at

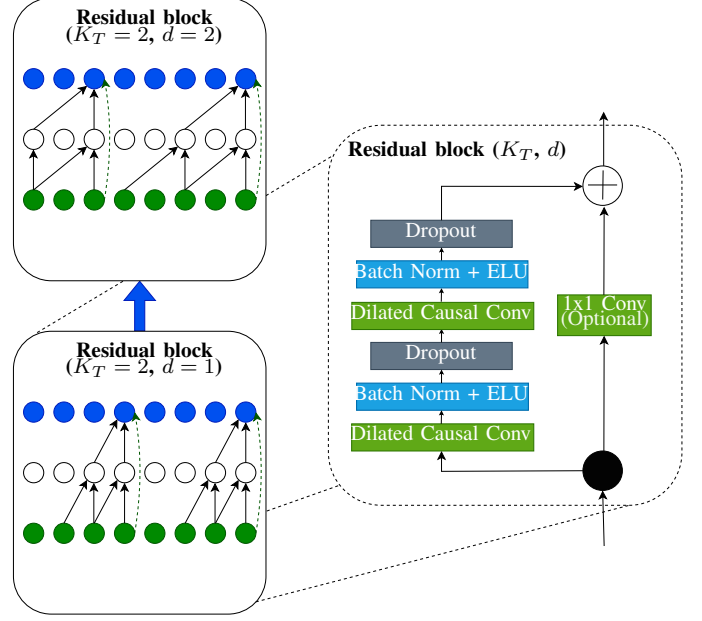


Fig. 2. Architectural elements in a TCN. Left: Stacking of two residual blocks highlighting the dilated convolutions with kernel size  $K_T = 2$  and dilation  $d = \{1, 2\}$ . Right: Detailed layers in TCN residual block.

the cost of 7.78 M parameters and 1.73 GMACs per inference.

#### D. Temporal Convolutional Networks

In the following, we describe the generic architecture concept known as the temporal convolutional network (TCN) [21], depicted in Fig. 2. Three properties distinguish TCNs from conventional CNNs:

1) *Causal Convolutions*: TCNs produce an output of the same length as the input. To this end, TCNs use a 1D fully-convolutional network (FCN) architecture [29], where each hidden layer is the same size as the input layer, and zero-padding of length (kernel size - 1) is added to keep subsequent layers the same length as the previous ones. Further, causal convolutions are used to force no information flow from the future to the past. Simply put, the output at time  $t$  depends only on inputs from time  $t$  and earlier.

2) *Dilated Convolutions*: A regular causal convolution is only able to increase its receptive field size linearly in the depth of the network. This is a major disadvantage since either an extremely deep network or one with a huge kernel size is needed to obtain a large receptive field size. To combat this problem, TCNs use a sequence of dilated convolutions [30], which allows the network to increase its receptive field exponentially in size proportional to the network depth by employing a scheme of exponentially increasing dilation factors  $d$ .

3) *Residual Blocks*: The residual block of a TCN consists of two layers of dilated convolutions, with batch normalization, non-linearity, and a dropout layer in-between the convolutions. Even though TCNs feature only 1D convolutions, they are still capable of processing 2D feature maps by

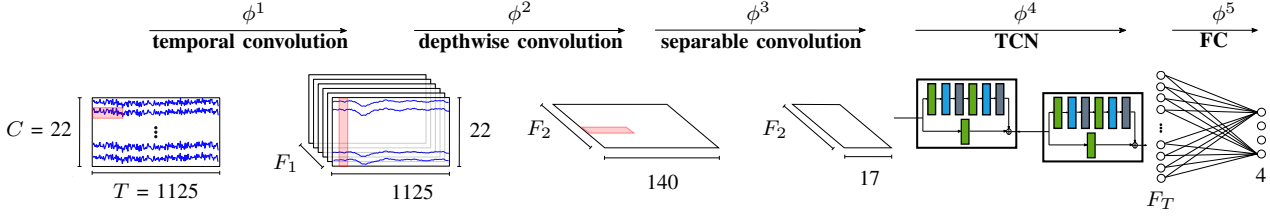


Fig. 3. Architecture of the EEG-TCNET. Where  $C$  = number of EEG channels,  $T$  = number of timepoints,  $F_1$  = number of temporal filters,  $F_2$  = number of pointwise filters and  $F_T$  = number of filters in TCN module.

considering the second dimension as the depth dimension. The skip connection adds the input to the output feature map, with the check that if the depth of the input and output is different, a  $1 \times 1$  convolution is put in place. See Fig. 2 for an illustration of a residual block and the stacking of two residual blocks together.

By stacking residual blocks, the receptive field size increases exponentially with each residual block, as the dilation in each subsequent block is exponentially larger. The receptive field size (RFS) of the TCN is determined by

$$\text{RFS} = 1 + 2 \cdot (K_T - 1) \cdot (2^L - 1), \quad (1)$$

where  $K_T$  is the kernel size and  $L$  the number of residual blocks.

The TCN described here slightly differs from the one explained in [21], in the residual block in the following ways:

- Batch normalization is used between convolutions instead of weight normalization as batch normalization has been shown to give higher accuracy than weight normalization on various large scale networks [31].
- We use the exponential linear unit (ELU) activation instead of the rectified linear unit (ReLU). This was done since EEG-TCNET showed better performance with a ELU activation function than ReLU.
- Instead of spatial dropout, normal dropout is used. As the TCN is applied after various convolutions the adjacent frames withing feature maps are not strongly correlated, and therefore it is beneficial to drop individual elements instead of entire 1D feature maps to regularize the activations.

### III. METHODOLOGY

In this section, we present the main contribution of the paper. We show how to combine the shallow, yet discriminative feature extraction layers of EEGNet with a TCN, making use of the temporal information present in the features, which would be ignored otherwise. We introduce our model, named EEG-TCNET, and analyze it either with a *fixed* set of hyperparameters for all subjects, or *variable* optimal subject-specific hyperparameters.

#### A. Data Pre-processing

The time frame used for both training and inference on EEG data of the BCI Competition IV-2a is 0.5 seconds before the MI-cue until the end of the MI [9], resulting in a time

series of 4.5 s length, or 1125 samples. The sampling rate is kept at 250 Hz, and no additional bandpass filtering is applied. Optionally, we apply standardization by removing mean and scaling to unit-variance per channel, based on the statistics of the training set.

#### B. EEG-TCNET

Fig. 3 illustrates the architecture of EEG-TCNET with a more detailed description of the layers in Table II. The network is, in part, inspired by the EEGNet architecture [10]. The network starts with a 2D temporal convolution to learn frequency filters, then uses a depthwise convolution to learn frequency-specific spatial filters. The separable convolution learns a temporal summary for each feature map individually and then mixes the feature maps. The output feature map of the separable convolution still contains temporal information; therefore, the addition of a TCN further exploits temporal information.

The first TCN block expands the  $F_2$  feature maps after the separable convolution to  $F_T$  feature maps. Overall, we stack  $L$  residual blocks and select the receptive field size such that  $\text{RFS} \geq 17$ , allowing the TCNs to capture all temporal information available. Finally, the last time steps of each of the  $F_T$  feature maps of the last residual block are read out and fed to a fully-connected layer for classification.

1) *Fixed* EEG-TCNET: A global architecture method involves choosing global hyperparameters for all subjects, then training and testing each subject separately. A cross-validated grid search on the training set over the hyperparameters yields the following optimal architecture that performed best for all subjects:  $F_1 = 8$ ,  $F_2 = 16$ ,  $K_E = 32$ ,  $K_T = 4$ ,  $L = 2$ ,  $F_T = 12$ ,  $p_e = 0.2$ ,  $p_t = 0.3$ , and with data standardization.

2) *Variable* EEG-TCNET: The accuracy of most classifiers on the BCI Competition IV-2a highly varies among individual subjects, e.g., EEGNet achieves accuracies ranging from 54.06%–88.80%. This might originate from the rigid network and training structure, applying the same network as well as optimizing hyperparameters to all subjects. Hence, we propose to find optimal subject-specific network parameters (e.g., kernel size, number of filters, or use of data standardization) and training hyperparameters (e.g., dropout rate) of EEG-TCNET using cross-validated grid search on the training for every individual subject. The test set is not touched for determining the optimal parameters, thus keeping it compatible with the



TABLE II  
EEG-TCNET ARCHITECTURE

Layer	Type	#Filters	Kernel	Output
$\phi^1$	Input			$(1, C, T)$
	Conv2D	$F_1$	$(1, K_E)$	$(F_1, C, T)$
	BatchNorm			
$\phi^2$	DepthwiseConv2D	$F_1 \cdot 2$	$(C, 1)$	$(2 \cdot F_1, 1, T)$
	BatchNorm			
	EluAct			
	AveragePool2D			$(2 \cdot F_1, 1, T//8)$
	Dropout			
$\phi^3$	SeparableConv2D	$F_2$	$(1, 16)$	$(F_2, 1, T//8)$
	BatchNorm			
	EluAct			
	AveragePool2D			$(F_2, 1, T//64)$
	Dropout			
$\phi^4$	TCN	$F_T$	$K_T$	$F_T$
$\phi^5$	Dense			4
	SoftMaxAct			

$C$  = number of EEG channels,  $T$  = number of time samples,  $F_1$  = number of temporal filters,  $F_2$  = number of spatial filters,  $K_E$  = kernel size in first convolution,  $K_T$  = kernel size in TCN module, and  $F_T$  = number of filters in TCN module. For dropout in EEGNet inspired layers we use  $p_e$ , and in the TCN module we use  $p_t$ .

rules of the BCI Competition IV-2a. For comparison, the same procedure is applied to EEGNet.

### C. Training Procedure

Models were trained and tested in a Tensorflow environment on an NVIDIA GTX 1080 Ti GPU. The networks were developed with Keras. We use the same training configuration when training the models proposed in this paper, where categorical cross-entropy loss is used, and the filter kernels are uniformly initialized following the procedure introduced in [32]. The models are trained for 750 epochs with an Adam optimizer at a learning rate of 0.001 and a batch size of 64. These training hyperparameters are determined via cross-validation on the training set.

## IV. EXPERIMENTAL RESULTS

### A. Performance Metrics

We evaluate the models according to the classification accuracy, which is the ratio between correctly classified trials and the total number of trials in the test set. Additionally, we report Cohen's  $\kappa$ -score defined as:

$$\kappa = \frac{p_o - p_e}{1 - p_e},$$

where  $p_o$  stands for the observed agreement ratio (e.g., accuracy) and  $p_e$  for the hypothetical probability of chance agreement or random classification rate.

We also report the number of parameters in each model and the number of MACs for inference. The calculation of MACs for a couple of different convolutional layers can be seen below:

$$\text{Conv2D} = K_1 \cdot K_2 \cdot C_{in} \cdot C_{out} \cdot H_{out} \cdot W_{out},$$

$$\text{Conv1D} = K \cdot C_{in} \cdot C_{out} \cdot W_{out},$$

$$\text{SeparableConv2D} = (K_1 \cdot K_2 + C_{out}) \cdot C_{in} \cdot H_{out} \cdot W_{out},$$

$$\text{DepthWiseConv2D} = K_1 \cdot K_2 \cdot C_{in} \cdot D \cdot H_{out} \cdot W_{out},$$

where  $K$  stands for the kernel size,  $C$  stands for the total number of channels. Then, the  $H$  and  $W$  are the height and width of the tensors, respectively.

Finally, we compare the memory footprint of the models during inference, which is defined here as the size of the two largest consecutive feature maps. For the calculation of the memory size, we assume that both the feature maps and weights of all the networks can be quantized to 8 bits at negligible accuracy loss based on literature [20].

### B. BCI Competition IV-2a

Table III summarizes the accuracy and  $\kappa$ -scores for both fixed and variable EEGNet and EEG-TCNET. Moreover, the table includes also the accuracy and  $\kappa$ -scores of the reproduced fixed Shallow ConvNet [9], and the accuracy of variable DFFN [11], since detailed results were reported in the paper. By first comparing the fixed networks, EEG-TCNET shows high robustness in classifying the nine subjects and achieves 77.35% accuracy and a  $\kappa$ -score of 0.70. This is an increase of 4.95% accuracy compared to EEGNet; moreover, the standard deviation of accuracy scores between subjects is 11.57%, which is significantly lower than the one for EEGNet (13.27%) and Shallow ConvNet (14.54%).

When focusing on the variable networks in Table III, we see that the addition of subject-specific hyperparameters increased the performance of EEG-TCNET by 6.49%, achieving the highest accuracy of 83.84%. Similarly, the introduction of variable hyperparameters in EEGNet improves the accuracy by 6.66%. Variable EEG-TCNET outperforms variable DFFN by 4.13%. Again it is noteworthy that variable EEG-TCNET exhibits the lowest standard deviation between subjects in both accuracy scores and  $\kappa$ -scores than other variable models. This underlines that our variable EEG-TCNET not only improves on already well-performing subjects, but enables higher accuracy for otherwise poorly performing subjects, e.g., Subject 2, Subject 5, or Subject 6.

Table IV summarizes the optimal subject-specific parameters for variable EEG-TCNET and EEGNet. Except for Subject 2, variable EEG-TCNET always makes use of data standardization, whereas variable EEGNet consistently classifies the raw data without standardization. Interestingly, variable EEG-TCNET requires, in general, a smaller number of temporal filters  $F_1$  and filter size  $K_E$  than EEGNet. The temporal filters pose the most restrictive limitations in terms of computational complexity and memory footprint since the temporal convolution requires the vast majority of MACs and memory to store the resulting feature maps [19]. Therefore, variable EEG-TCNET has more potential to be embedded on a resource-limited device.

Table V compares the current SoA networks on the BCI Competition IV-2a in terms of accuracy, the number of trainable parameters, MACs, and memory requirements. For the variable models, we report the maximum number of parameters and memory requirements, as they pose a hard requirement when considering the embedding to an MCU. By first looking at the parameter count and MACs of fixed

TABLE III  
CLASSIFICATION ACCURACY (%) AND  $\kappa$  SCORES ON THE 4-CLASS MI BCI COMPETITION IV-2A DATASET.

	EEGNet* [10]		Fixed Networks		EEG-TCNET		Variable Networks		DFFN [11]
	Accuracy	$\kappa$	Shallow ConvNet* [9]		Accuracy	$\kappa$	EEGNet	EEG-TCNET	
			Accuracy	$\kappa$			Accuracy	Accuracy	Accuracy
Subject 1	84.34	0.79	79.51	0.73	85.77	0.81	86.48	89.32	83.46
Subject 2	54.06	0.39	56.25	0.42	65.02	0.53	61.84	72.44	69.30
Subject 3	87.54	0.83	88.89	0.85	94.51	0.93	93.41	97.44	90.29
Subject 4	63.59	0.51	80.90	0.75	64.91	0.53	73.25	75.87	71.07
Subject 5	67.39	0.57	57.29	0.43	75.36	0.67	76.81	83.69	65.41
Subject 6	54.88	0.39	53.82	0.38	61.40	0.49	59.07	70.69	69.45
Subject 7	88.80	0.85	91.67	0.89	87.36	0.83	90.25	93.14	88.18
Subject 8	76.75	0.69	81.25	0.75	83.76	0.78	87.45	86.71	86.76
Subject 9	74.24	0.65	79.17	0.72	78.03	0.71	82.95	85.23	93.54
Mean	72.40	0.63	74.31	0.66	<b>77.35</b>	<b>0.70</b>	79.06	<b>83.84</b>	79.71
Std. Dev.	13.27	0.18	14.54	0.19	<b>11.57</b>	<b>0.15</b>	12.28	<b>9.20</b>	10.79

\* Reproduced

TABLE IV  
OPTIMAL NETWORK HYPERPARAMETERS FOR EACH SUBJECT OF THE BCI COMPETITION IV-2A DATASET.

Subject	Variable EEG-TCNET									Variable EEGNet								
	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
$K_T$	3	4	4	4	3	4	4	3	3									
$p_t$	0.3	0.2	0.3	0.2	0.2	0.3	0.3	0.3	0.2									
$L$	3	2	2	3	4	3	2	3	4									
$F_T$	15	17	15	17	25	17	20	25	12									
$F_1$	8	8	8	16	16	16	8	16	16	32	32	8	16	32	32	32	32	8
$K_E$	32	64	64	32	64	32	32	64	64	128	128	64	32	32	64	32	32	64
$p_e$	0.2	0.2	0.2	0.1	0.2	0.1	0.1	0.2	0.2	0	0	0.1	0.1	0	0.2	0.2	0.2	0
$S$	True	False	True	True	True	True	True	True	True	False	False	False	False	False	False	False	False	False
Parameters	6144	6793	5815	12171	20526	12171	8184	16526	8176	15620	15620	2628	5252	12548	13572	12548	12548	2628

$K_T$  : Kernel size in TCN module,  $p_t$ : Dropout rate in TCN module,  $L$ : # of residual blocks,  $S$ : Standardize data

$F_T$ : Filters in convolutional layers,  $F_1$ : Temporal filters in EEGNet,  $K_E$ : Kernel size in EEGNet,  $p_e$ : Dropout rate in EEGNet

TABLE V  
METRICS OF CURRENT SOA MODELS OF BCI COMPETITION IV-2A.

	Mean Accuracy	Parameters	Mean MACs	Feature Map [kB]
EEGNet* [10]	72.40	2.63 k	13.1 M	396
Shallow ConvNet* [9]	74.31	47.3 k	63.0 M	1013
FBCSP [8]	73.70	261 k	104 M	50
Riemannian [8]	74.77	50.0 k	-	49
MSFBCNN [27]	75.80	155 k	202 M	5775
EEG-TCNET	<b>77.34</b>	<b>4.27 k</b>	<b>6.8 M</b>	<b>396</b>
CNN++ [12]	81.10	240 k	96.4 M	499
TPCT [13]	88.87	7.78 M	1.73 G	524
Variable EEGNet	79.02	15.6 k	42.6 M	1584
DFFN (variable) [11]	79.71	1.07 M	132 M	650
Variable EEG-TCNET	<b>83.84</b>	<b>20.5 k</b>	<b>12.1 M</b>	<b>792</b>

\* Reproduced

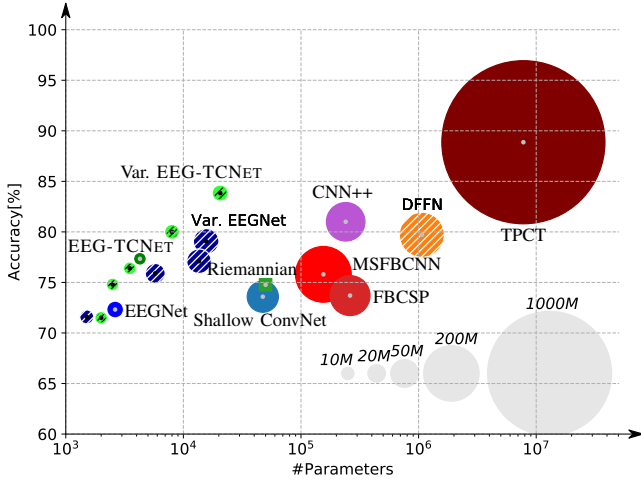
networks, we see that EEG-TCNET requires only 6.8 MACs, which is  $1.9\times$  lower than EEG-Net, while being 4.95% more accurate. The large reduction in complexity comes from the smaller temporal filter size, where fixed EEG-TCNET uses  $K_T = 32$  instead of  $K_T = 64$ . When allowing subject-specific network hyperparameters in the variable EEG-TCNET, the maximum number of parameters increases by  $4.80\times$  and the MACs by  $1.78\times$ , compared to fixed EEG-TCNET. TPCT achieves the highest accuracy of 88.87%. However, this comes at the cost of a  $380\times$  higher number of parameters and  $143\times$  more MACs than variable EEG-TCNET.

Another consideration is the maximum memory requirements of the networks, which—assuming layer-by-layer inference—is the sum of the two largest consecutive feature

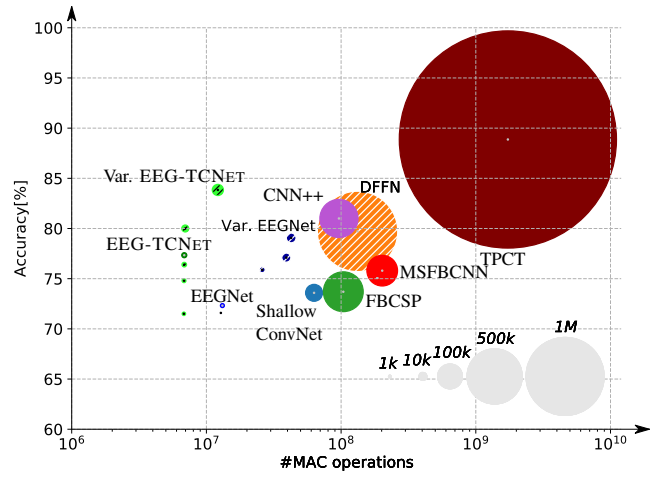
maps. Due to their residual structure, DFFN [11] and MSF-BCNN [27] are calculated differently; DFFN needs to store the first five feature maps in the first dense block, and MSF-BCNN [27] the first five feature maps as later layers depend on all of these feature maps rather than the immediately preceding layer's output. This results in MSFBCNN having the largest feature map size while DFFN manages to keep it relatively low despite the network architecture. Interestingly, TCPT has the fourth-smallest feature map size despite having the largest parameter count and most number of MACs.

TPCT's overall memory footprint is 8.304 MB, and thus far beyond the on-chip memory capacity available in an ARM M7 processor. Its compute effort of 1.73 GMACs would take approximately 50 s/inference— $17\times$  below real-time when requiring a new classification at least every 3 s—where we refer to the throughput of 34.45 MMAC/s of an ARM M7 processor [19]. In comparison, the proposed EEG-TCNet has a memory footprint of 400 kB, and its compute effort of 6.8 MMACs would take approximately 197 ms. EEG-TCNET and variable EEG-TCNET are the best candidates for an embedded implementation; both parameter count and inference cost are kept reasonable while still achieving very high accuracy scores.

Fig. 4 visualizes the trade-off between the accuracy, number of parameters, and MACs of all models. We also include FBCSP and Riemannian [8]; the Riemannian model has a square shape in Fig. 4a and is not included in Fig. 4b as the number of MACs is not clear as its compute workload



(a) Accuracy vs. parameters, size of circles  $\propto$  MACs.



(b) Accuracy vs. MACs, size of circles  $\propto$  number of parameters.

Fig. 4. Classification accuracy on BCI Competition IV-2a vs. (a) parameters and (b) MACs per inference. The circles are proportional to (a) MACs and (b) parameters, with the corresponding legend in grey reported on the bottom right. The grey dots inside the circles highlight their center. The hatched circles show variable subject-specific networks.

is not predominantly based on MAC operations. We further experiment with limiting the number of parameters that the variable models are allowed to have in each network. Specifically, we limit EEG-TCNET to 2k, 2.5k, 3.5k, and 8k parameters and EEGNet to 1.5k, 5.7k, and 13.5k. We recognize that EEG-TCNET achieves Pareto optimality by spanning almost the entire Pareto front in both the parameter and MAC comparison.

### C. Mother of All BCI Benchmarks

We benchmark EEG-TCNET and EEGNet on MOABB by comparing them to three other pipelines included in MOABB [23]. These pipelines are:

- CSP + LDA: where trial covariances were estimated via maximum-likelihood with unregularized CSP. Features were log-variance of the filters belonging to the six most diverging eigenvalues and then classified with LDA.
- TS + optSVM: where trial covariances were estimated via oracle approximating shrinkage, then projected into the Riemannian tangent space to obtain features and classified with a linear SVM with identical grid search.
- AM + optSVM: where features are the log-variance in each channel and then classified with a linear SVM with grid search.

Fig. 5 shows the meta-analysis of the comparison between EEG-TCNET and TS + optSVM, which is the current SoA on MOABB. The meta-effect reports the combined standardized mean differences across all datasets. The standardized mean difference is combined with a weighting given by the square root of the number of subjects. Then, the p-value of a one-tailed Wilcoxon signed-rank test for the hypothesis that EEG-TCNET is more accurate is also reported. While we observe a high variance among different datasets that could give contradictory results if the methods were evaluated on

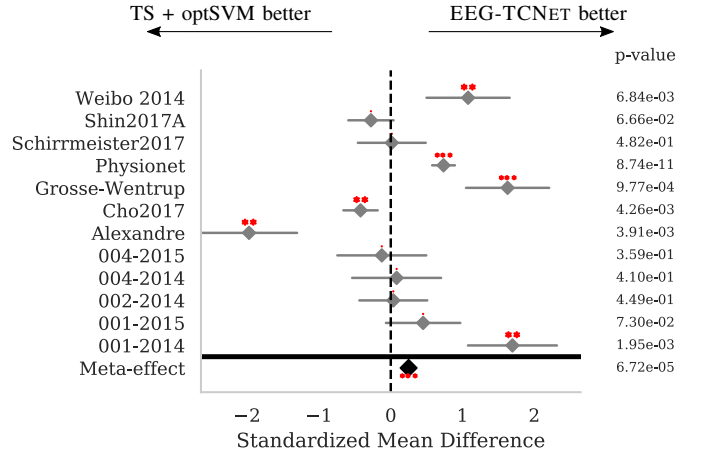


Fig. 5. Meta-analysis comparing tangent space features on optimized SVM (TS + optSVM) against EEG-TCNET on MOABB. The effect sizes shown are standardized mean differences, with p-values corresponding to the one-tailed Wilcoxon signed-rank test for the hypothesis given at the top of the plot and 95% interval denoted by the grey bar. Stars correspond to \*\*\*  $= p < 0.001$ , \*\*  $= p < 0.01$ , \*  $= p < 0.05$ . The meta-effect is shown at the bottom.

one dataset in isolation, the overall trend shows that EEG-TCNET outperforms TS + optSVM with a final meta-effect of 0.25.

Fig. 6 summarizes the comparisons between all methods, showing the meta-effect in case that the method on the vertical axis significantly outperforms the method on the horizontal axis, according to the one-tailed Wilcoxon signed-rank test. EEG-TCNET outperforms all other methods; thus, it is becoming the new SoA on MOABB. This experiment underlines that EEG-TCNET generalizes well outside a single MI dataset, where it was modeled on.

AM + optSVM					
CSP + LDA	0.81 p=8e-16				
EEGNet	1.37 p=2e-20	0.38 p=5e-04			
TS + optSVM	1.29 p=1e-23	0.46 p=4e-04	0.02 p=6e-01		
EEG-TCNET	1.22 p=4e-05	0.54 p=2e-10	0.18 p=4e-04	0.25 p=7e-05	
	AM + optSVM	CSP + LDA	EEGNet	TS + optSVM	EEG-TCNET

Fig. 6. Meta-effect on MOABB with the hypothesis that the method on the vertical axis performs better than the one on the horizontal axis. All p-values are single-sided; in the case the effect goes in the opposite direction of the hypothesis, the values are removed for clarity. The values correspond to the standardized mean difference of the algorithm in the y-axis minus that in the x-axis and the associated p-value.

## V. CONCLUSION

We have proposed EEG-TCNET, a novel model for accurate MI-BMI classification. Thanks to its low memory footprint and limited computational complexity, it can be easily operated on low-power resource-limited devices at the edge. It achieves 77.35% accuracy on the BCI Competition IV-2a dataset, improving the SoA of similarly-sized networks by 4.95%. We then further enhance the model by performing a subject-specific hyperparameter search, which yields an additional 6.49% accuracy increase, achieving a high accuracy of 83.84%. Moreover, it requires a low number of parameters, MACs, and memory usage during inference. Large scale benchmark tests on MOABB confirm that EEG-TCNET generalizes well to other MI datasets, becoming the new SoA on the MOABB framework outperforming the old SoA by a final meta-effect of 0.25.

## REFERENCES

- [1] B. Graimann, B. Allison, and G. Pfurtscheller, "BrainComputer Interfaces: A Gentle Introduction," 2009, pp. 1–27.
- [2] M. Xiong, R. Hotter, D. Nadin, J. Patel, S. Tartakovsky, Y. Wang, H. Patel, C. Axon *et al.*, "A Low-Cost, Semi-Autonomous Wheelchair Controlled by Motor Imagery and Jaw Muscle Activation," in *Proc. IEEE SMC*, 2019, pp. 2180–2185.
- [3] K. A. Condori, E. C. Urquiza, and D. A. Diaz, "Embedded Brain Machine Interface based on motor imagery paradigm to control prosthetic hand," in *Proc. IEEE ANDESCON*, 2016.
- [4] J. H. Cho, J. H. Jeong, K. H. Shim, D. J. Kim, and S. W. Lee, "Classification of Hand Motions within EEG Signals for Non-Invasive BCI-Based Robot Hand Control," *Proc. IEEE SMC*, 2018, pp. 515–518.
- [5] W. Cho, A. Heilinger, R. Ortner, N. Murovec, R. Xu, J. Swift, M. Zehetner, S. Schobesberger *et al.*, "Motor Rehabilitation for Hemiparetic Stroke Patients Using a Brain-Computer Interface Method," in *Proc. IEEE SMC*, 2018, pp. 1001–1005, ISSN: 2577-1655.
- [6] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for EEG-based braincomputer interfaces: a 10 year update," *Journal of Neural Engineering*, vol. 15, no. 3, p. 031005, 2018.
- [7] K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan, "Filter Bank Common Spatial Pattern (FBCSP) in brain-computer interface," *Proc. IJCNN*, pp. 2390–2397, 2008.
- [8] M. Hersche, T. Rellstab, P. D. Schiavone, L. Cavigelli, L. Benini, and A. Rahimi, "Fast and Accurate Multiclass Inference for MI-BCIs Using Large Multiscale Temporal and Spectral Features," in *Proc. IEEE EUSIPCO*, 2018, pp. 1690–1694.
- [9] R. T. Schirrmester, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggersperger, M. Tangermann, F. Hutter, W. Burgard *et al.*, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Human Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.
- [10] 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 braincomputer interfaces," *Journal of Neural Engineering*, vol. 15, no. 5, p. 056013, 2018.
- [11] D. Li, J. Wang, J. Xu, and X. Fang, "Densely Feature Fusion Based on Convolutional Neural Networks for Motor Imagery EEG Classification," *IEEE Access*, vol. 7, pp. 132 720–132 730, 2019.
- [12] Y. Zhao, S. Yao, S. Hu, S. Chang, R. Ganti, M. Srivatsa, S. Li, and T. Abdelzaher, "On the improvement of classifying EEG recordings using neural networks," in *Proc. IEEE Big Data*, 2017, pp. 1709–1711.
- [13] 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, 2020.
- [14] C. Brunner, R. Leeb, G. R. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "BCI competition 2008 - Graz data set A."
- [15] O. Landau, A. Cohen, S. Gordon, and N. Nissim, "Mind your privacy: Privacy leakage through BCI applications using machine learning methods," *Knowledge-Based Systems*, vol. 198, p. 105932, 2020.
- [16] L. Cavigelli and L. Benini, "Cbinfer: Exploiting frame-to-frame locality for faster convolutional network inference on video streams," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.
- [17] R. Andri, L. Cavigelli, D. Rossi, and L. Benini, "Hyperdrive: A multi-chip systolically scalable binary-weight cnn inference engine," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 2, pp. 309–322, 2019.
- [18] X. Wang, M. Magno, L. Cavigelli, and L. Benini, "FANN-on-MCU: An Open-Source Toolkit for Energy-Efficient Neural Network Inference at the Edge of the Internet of Things," *IEEE IoT Journal*, 2020.
- [19] X. Wang, M. Hersche, B. Tmekce, B. Kaya, M. Magno, and L. Benini, "An Accurate EEGNet-based Motor-Imagery Brain-Computer Interface for Low-Power Edge Computing," *arXiv:2004.00077*, 2020.
- [20] T. Schneider, X. Wang, M. Hersche, L. Cavigelli, and L. Benini, "Q-EEGNet: an Energy-Efficient 8-bit Quantized Parallel EEGNet Implementation for Edge Motor-Imagery Brain-Machine Interfaces," *arXiv:2004.11690v1*, 2020.
- [21] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *arXiv:1803.01271*, 2018.
- [22] N. Lu, T. Yin, and X. Jing, "Deep Learning Solutions for Motor Imagery Classification: A Comparison Study," in *Proc. International Winter Conference on BCI*, 2020, ISSN: 2572-7672.
- [23] V. Jayaram and A. Barachant, "MOABB: trustworthy algorithm benchmarking for BCIs," *Journal of Neural Engineering*, vol. 15, no. 6, p. 066011, 2018.
- [24] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring *et al.*, "Review of the BCI Competition IV," *Frontiers in Neuroscience*, vol. 6, 2012.
- [25] Y. Li, Y. Qi, and Y. Wang, "Avoiding subject-specific model selection via highway networks in EEG signals," in *Proc. BIBE*, 2019, pp. 133–137.
- [26] A. Uran, C. van Gemenen, R. van Diepen, R. Chavarriaga, and J. d. R. Milln, "Applying Transfer Learning To Deep Learned Models For EEG Analysis," *arXiv:1907.01332*, 2019.
- [27] H. Wu, Y. Niu, F. Li, Y. Li, B. Fu, G. Shi, and M. Dong, "A Parallel Multiscale Filter Bank Convolutional Neural Networks for Motor Imagery EEG Classification," *Frontiers in Neuroscience*, vol. 13, 2019.
- [28] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *arXiv:1608.06993*, 2018.
- [29] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE CVPR*, 2015, pp. 3431–3440.
- [30] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior *et al.*, "WaveNet: A Generative Model for Raw Audio," *arXiv:1609.03499*, 2016.
- [31] I. Gitman and B. Ginsburg, "Comparison of Batch Normalization and Weight Normalization Algorithms for the Large-scale Image Classification," *arXiv:1709.08145 [cs]*, 2017.
- [32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.