Relation Learning Using Temporal Episodes for Motor Imagery Brain-Computer Interfaces

Xiuyu Huang, Shuang Liang, Yuanpeng Zhang, *Member, IEEE*, Nan Zhou, Witold Pedrycz, *Life Fellow, IEEE*, and Kup-Sze Choi, *Member, IEEE*

***Abstract*— For practical motor imagery (MI) brain-computer interface (BCI) applications, generating a reliable model for a target subject with few MI trials is important since the data collection process is labour-intensive and expensive. In this paper, we address this issue by proposing a few-shot learning method called temporal episode relation learning (TERL). TERL models MI with only limited trials from the target subject by the ability to compare MI trials through episode-based training. It can be directly applied to a new user without being re-trained, which is vital to improve user experience and realize real-world MIBCI applications. We develop a new and effective approach where, unlike the original episode learning, the temporal pattern between trials in each episode is encoded during the learning to boost the classification performance. We also perform an online evaluation simulation, in addition to the offline analysis that the previous studies only conduct, to better understand the performance of different approaches in real-world scenario. Extensive experiments are completed on four publicly available MIBCI datasets to evaluate the proposed TERL. Results show that TERL outperforms baseline and recent state-of-the-art methods, demonstrating competitive performance for subject-specific MIBCI where few trials are available from a target subject and a considerable number of trials from other source subjects.**

Manuscript received 8 July 2022; revised 6 September 2022, 9 November 2022, and 1 December 2022; accepted 6 December 2022. Date of publication 9 December 2022; date of current version 1 February 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 82072019, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20201441, and in part by the Hong Kong Research Grants Council under Grant PolyU 152006/19E. *(Corresponding authors: Yuanpeng Zhang; Kup-Sze Choi.)*

Xiuyu Huang and Kup-Sze Choi are with the Center for Smart Health, The Hong Kong Polytechnic University, Hong Kong SAR, China (e-mail: thomasks.choi@polyu.edu.hk).

Shuang Liang is with the Smart Health Big Data Analysis and Location Services Engineering Laboratory of Jiangsu Province, Nanjing University of Posts and Telecommunications, Nanjing 210003, China.

Yuanpeng Zhang is with the Medical Informatics, Nantong University, Nantong 226007, China (e-mail: y.p.zhang@ieee.org).

Nan Zhou is with the School of Electronic Information and Electronic Engineering, Chengdu University, Chengdu 610014, China.

Witold Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2R3, Canada, also with the Systems Research Institute, Polish Academy of Sciences, 00-901 Warsaw, Poland, also with the Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia, and also with the Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Istinye University, 34010 Sariyer/Istanbul, Turkey.

This article has supplementary downloadable material available at https://doi.org/10.1109/TNSRE.2022.3228216, provided by the authors.

Digital Object Identifier 10.1109/TNSRE.2022.3228216

***Index Terms*— Motor imagery, brain-computer interface, temporal encoding, episode training.**

I. Introduction

BRAIN-COMPUTER interface (BCI) enables humans to interact with the world by only relying on their brain activities without any muscular movements [1]. One of the most popular paradigms employed in BCIs is motor imagery (MI). It is a cognitive process where individuals imagine a movement of either one or several parts of their body without performing any actual movement nor activating muscles [2]. MI of different parts of the body elicit different sensorimotor rhythms (SMRs). These SMRs can be captured by brain signal collection techniques [3], such as electroencephalography (EEG), the most accessible and common one used in BCIs. The idea of the EEG-based motor imagery brain-computer interface (MIBCI) is to decode the user's intention by analyzing the EEG signals that contain distinct patterns of SMRs. MIBCI has been successfully deployed in various applications such as external skeleton [4], speller [5], and wheelchair [6].

Researchers have established many attractive approaches and algorithms for MI classification in BCI systems. Conventional methods rely on discriminative hand-crafted features to classify MI. Common spatial pattern (CSP) is one of the most well-known methods [7]. It generates spatial filters that maximize variances between multiple classes in a certain frequency band. Filter-bank common spatial pattern (FBCSP) [8] based on CSP is an elegant linear method which aims to find a set of spatial filters to maximize the differences in variances between MI classes in multiple frequency bands rather than in only one single band. This method is the winner in the BCI Competition IV, achieving 67.75% accuracy in MI classification on dataset IV-2a [9]. However, pre-processing techniques are required to be applied to raw data when using these methods. They can not offer an end-to-end way for the MI classification task.

Alternatively, deep learning (DL) methods provide an end-to-end framework to deal with the MI classification task and achieve a promising performance [10]. They reach the state-of-the-art (SOTA) classification accuracy [10], [11] in multiple public MI benchmarks such as Physionet [12] and those in BCI competitions [9], [13]. The convolutional neural network (CNN) is one of the popular choices in MI signal

processing [10]. One of the main characteristics of CNN is weight sharing, making it quite efficient in terms of memory and complexity. Albeit its low complexity and small memory usage, CNN still maintains an outstanding ability to extract discriminative features. Specifically, CNN layers are able to process patterns at diverse temporal scales with different kernel sizes. Such layers are likely to take full advantage of the information available in temporal series (e.g., MI trials) and generate deep discriminative representations for the MI classification [10]. Schirrmeister et al. [14]. introduced a deep CNN architecture that consists of temporal convolution filters, spatial convolution filters, and convolution pooling blocks. It is a well-known end-to-end CNN structure for MI classification which exhibits encouraging performance. Another popular CNN framework, called EEGNet [15], was proposed by Lawhern et al. in 2018. It is a light-weighted network with only thousands of parameters. It significantly saves the training and calibration time and still delivers competitive performance at the same time. These two architectures are popular backbone networks in the MI classification task regarding DL approaches. Many other interesting approaches have been established on their basis [16], [17], [18].

Most recent SOTA methods usually follow the standard supervised manner to learn the model. This paradigm is straightforward by using the MI input data X to learn a model $f(\cdot)$ to predict probability $\hat{y}$ that X belongs to a certain class. The training is guided by a specific type of loss function, mostly cross-entropy [20], between $\hat{y}$ and ground truth y. The network learns how to classify MI and has been successfully applied to many previous studies based on the offline evaluation [14], [15], [16], [17], [18], [21]. However, it may not be suitable for the real-world MIBCI which is often deployed in an online prediction setting [22], where only limited data can be collected from the target subject for the initial setup. We certainly can use this small dataset to train the model. However, the model will usually suffer from the over-fitting issue and might exhibit poor performance [23]. To alleviate this problem, several supervised transfer learning methods are proposed to leverage a large amount of available data from other subjects in addition to the target data. For example, Zhang et al. [16] applied the fine-tuning technique to different parts of the network. They used the MI data from the target subject to re-train some parameters of the model that have been pre-trained by the MI data from other source subjects. Several researchers instead focused on co-training (i.e., domain adaptation) approaches using data from target and source subjects to concurrently train the model from scratch. For instance, one research group subsequently introduced two different methods [24], [25] to train a model using adversarial learning [26] for the alignment of the deep features between source and target subjects. In addition, a few other studies attempted to add an auxiliary penalization term, e.g., KL divergence [27] or maximum mean discrepancy (MMD) [28] into the objective function to address the issue of the distribution shift between source and target deep features. However, these strategies assume that there are still a reasonable number (i.e. at least 200 trials) of MI trials from the target subject. Thus, they may not be optimal choices for the real-world BCI that usually does not have such a long period of data collection.

Instead of training in a direct supervision way, we can also guide a model to learn an easier task - finding relations (i.e., comparison) between MI classes. The model compares an unseen trial with a small pool of labeled trials. The unseen trial is assigned the class of a labeled trial if they have the closest relation. This is inspired by the relation network (RN) [19] in the few-shot classification scenario. This method originally aims to train a model that can identify the relations between image objects using a large number of labeled data [19]. After the training, the model can output reliable relation scores between objects even if the classes of the testing data are never exposed in training (i.e., the label space of training data is disjoint with testing one, see examples in Fig. 1). Again, the prediction of unseen input is made based on the largest relation score between itself and one of the few labeled data. The training guides the model to learn the relation between input objects rather than detecting their classes. We here utilize the RN for making predictions on MI data with the same class space but from a new target subject (Fig. 1). It is clear that we require a few labeled samples per class available from the target subject in order to calculate the relation scores and classify testing trials. This perfectly adapts to the online prediction setting for real-world BCIs. They usually collect several data from the target subject at the beginning for the model re-training.

Notably, RN utilizes well-designed mini-batches of data, called episodes [23], for training [19]. The episode learning is to imitate the testing environment of the few-shot classification during the training in order to improve the model's generalization [23]. To our best knowledge, only limited previous studies apply episode learning in the EEG-based classification [29], [30], [31], [32]. They apply such a technique to train a few-shot classifier in the tasks of MI classification [29], emotion detection [30], [31], and autism spectrum disorder (ASD) diagnosis [32], respectively. However, two existing gaps in the episode-based training are still to be addressed. First, the temporal information between the trials sampled from the data pool within each episode is ignored in the previous studies. The second gap in the existing literature is that only offline evaluation was performed. The previous results may not be generalizable to the requirement of online prediction in a real-world application. Corresponding to these gaps, this study makes the following contributions.

1) We propose a temporal episode relation learning (TERL), a simple but effective way to extract the temporal information in each training episode and improve the model performance. In this method, the order of trials in episodes, network architecture, and sampling methods to formulate episodes are carefully designed.
2) In addition to the common offline evaluation, we also extensively test different learning methods in an emulating online setting (i.e., online evaluation simulation) to fully understand their effectiveness in the MIBCI.
3) The proposed TERL outperforms the recent SOTA approaches and baselines with 0.5 - 2.9% and 1.2 -

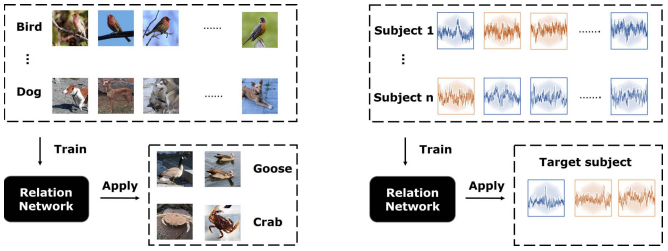

Fig. 1. Tasks of the relation network. Left: original task in the image classification [19]; right: MI classification task, hand MI (blue) and foot MI (orange).

4.0% accuracy improvements in offline evaluation and online evaluation simulation, respectively, across different datasets and MI classification tasks.

The article is organized as follows. Section II explains the proposed method in detail. Section III describes our experiments. Section IV presents the results and covers their corresponding analysis. The conclusion is drawn in Section V.

II. Temporal Episode Relation Learning

This section first introduces the problem to be addressed in the study. Then, we present the learning framework of the proposed TERL, the network architecture employed in this study, and different sampling techniques used to formulate the training episodes. Lastly, we summarize the advantages of our approach in the few-shot MI classification task against the original episode training method [23].

A. Problem Formulation

Although the setting in this study is different from the usual few-shot learning scenario in computer vision [19], [23], we still can inherit some terminologies and definitions used in the previous studies. In the present work, it is assumed that a relatively large pool of MI trials from other source subjects (***training set***) are always available at hand. Only limited labeled MI trials per class are collected from the target subject for the initial setup (***support set***). This fits the usual case encountered in the real-world BCI application. We apply the idea of episode-based learning in this few-shot scenario. The few-shot classifier is trained using the data from the training set. Then, it makes classification predictions to the unseen trials (***testing set***) of the target subject based on their relation scores with the trials in the support set. These three sets have the same label space with C unique classes. If the support set (i.e., initial setup) contains K trials per class, the problem is called C-way K-shot.

The principle idea of the episode-based learning is to imitate the testing interface during the training [23]. In each training iteration, K trials per class (i.e, total number of samples $m = K \times C$) are randomly picked out from the training set as a ***sample set*** (S). A fraction of the remaining samples in the training set is then selected as a ***query set*** (Q). The sample/query samplings are to stimulate the support/testing interface. Q and S are used to train the model and are randomly re-selected in each iteration.

B. Learning Framework

A single trial of EEG MI signal is denoted as (x_i, y_i). $x_i \in \mathbb{R}^{E\times T}$ is the input of the model, where E is the number of EEG electrodes and T is the number of time points. $y_i \in \mathbb{R}^C$ is the corresponding label of C classes. The proposed TERL consists of three parts, i.e., a trial embedding module (f_θ), an episode-based temporal encoding module (g_ϕ), and a relation module (h_γ). The overall design is illustrated in Fig. 2. We separately introduce the training and testing/interface.

1) Training: As mentioned in subsection II. A, a sample set and a query set need to be generated from the trials of source subjects in each training iteration/episode (Fig. 2, top), where samples $(\{x_i\}_i^m)$ in S are fed into the embedding function f_θ. The function generates the feature mappings $\{f_\theta(x_i)\}_i^m$, where m is the number of MI trials in S and is equal to $K \times C$ as stated in subsection II. A. It is noted that $\{x_i\}_i^m$ in our method are sorted in temporal order so as $\{f_\theta(x_i)\}_i^m$. Then, the episode-based temporal encoding module further embeds each $f_\theta(x_i)$ depending on itself and a_i preceding elements, i.e. $g_\phi(f_\theta(x_i), V_i)$, where

$$V_i = \{G(r)\}, r = i - a_i, i + 1 - a_i, \ldots, i - 1. \quad (1)$$

$$G(r) = \begin{cases} f_\theta(x_r), & r > 0; \\ 0, & otherwise. \end{cases} \quad (2)$$

The proposed g_ϕ is inspired by the MI EEG data collection and online prediction, where brain activity at the present moment strongly correlates with previous moments [10]. We call this function as *episode-based temporal encoding*, because it considers the episode as a sequence and captures its temporal patterns. The encoding of the current trial utilizes both the information from its own and proceeding trials. The g_ϕ can be easily implemented by popular DL architectures, such as CNNs and recurrent neural networks (RNNs), in an episode-based way rather than a within-trial-based way in the previous studies [30], [33]. Then, the feature map (M_c) of each class in S is represented by the sum of its corresponding temporal features, defined as

$$M_c = \sum_i^m \delta(y_i = c) \cdot g_\phi(f_\theta(x_i), V_i), \ c = 1, 2, \ldots, C. \quad (3)$$

where $\delta(condition)$ is the indicator function, and it is equal to 1 if the condition is satisfied and equal to 0 otherwise.

Each sample of $\{x_j\}_j^n$ in Q is one-by-one fed into f_θ to get the feature embedding $f_\theta(x_j)$, where n is the number of samples in Q. Note that Q is to imitate the testing set, which is expected to be unknown and may be sorted in any order during the testing interface. Therefore, samples in Q are not feedforwarded into g_ϕ. Each feature map M_c of S and each feature map $f_\theta(x_j)$ of Q are combined with a simple concatenation operator, denoted as $O(M_c, f_\theta(x_j))$. Therefore, we have C concatenated feature maps for each x_j. These feature maps are the input to the relation module h_γ. It outputs a scalar score (r) between 0 to 1, representing the similarity between the representative feature (M_c) of each class in S and each x_j in Q. We always generate C relation scores for each sample x_j in Q.

$$r_{c,j} = h_\gamma(O(M_c, f_\theta(x_j))), \ c = 1, 2, \ldots, C. \quad (4)$$

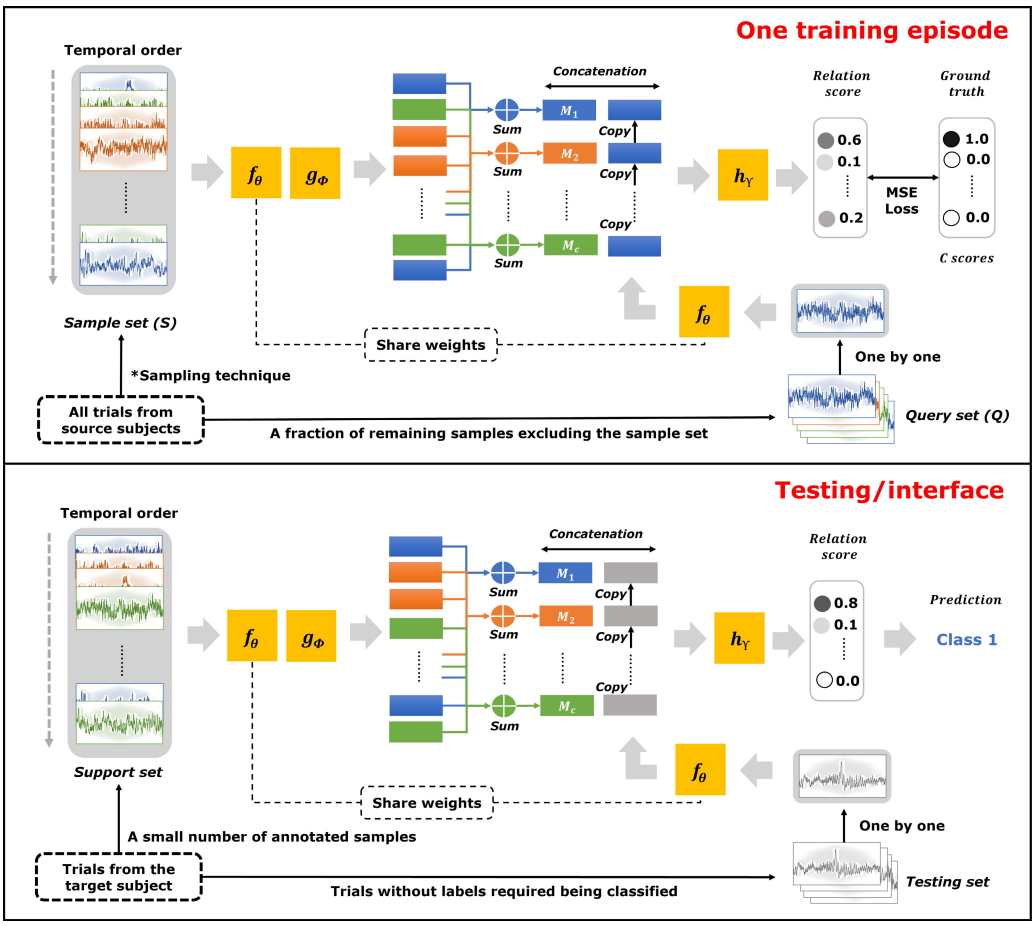

Fig. 2. An example of a single training episode (top) and the testing/interface (bottom) of the proposed framework. Annotated MI trials in different classes are in different colours. One colour, except grey, denotes one class. The trials in grey are required to be classified during the testing/interface. Rectangles denote the feature embedding vectors. M_1, M_2, ..., and M_C (described in subsection II. B) are representative features for C classes, respectively. Different *sampling techniques are applied in the present study to generate the sample set in the training episode, and they are described in detail in subsection II.D. The upper part of the figure only shows one training episode. This process is iteratively executed for many times (e.g., 10000) in training to optimize the f_θ, g_ϕ, and h_γ. These functions are fixed during the testing/interface. The classification of all testing trials of the target subject is based on these optimized functions and the annotated support set.

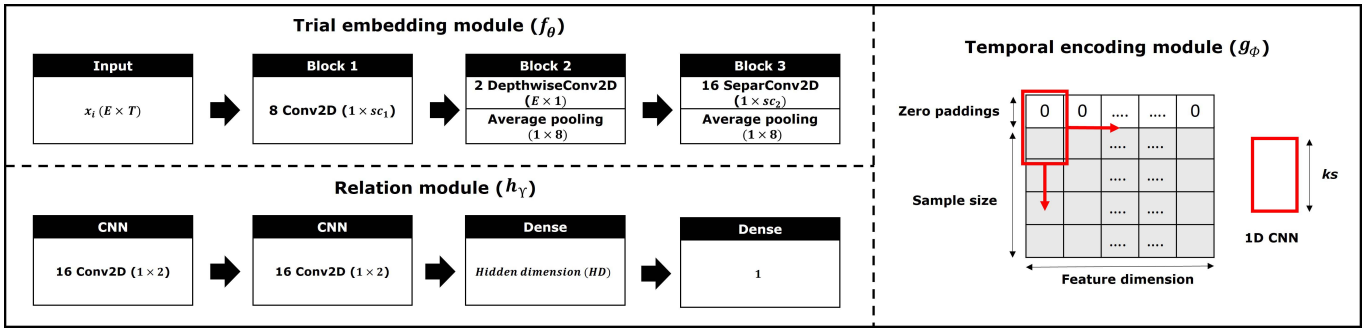

Fig. 3. Network architectures for the trial embedding module (f_θ), temporal encoding module (g_ϕ), and relation module (h_γ).

Since we have n samples in Q, $n \times C$ relation scores are generated for each training episode. The mean square error (MSE) loss is used to train the network. It regresses the relation score $r_{c,j}$ to ground truth: same-class pair equals 1, different-class pair equals 0. Details of the optimization of the loss are as follows.

$$\theta, \phi, \gamma \leftarrow \arg\min_{\theta,\phi,\gamma} \sum_{c}^{C} \sum_{j}^{n} (r_{c,j} - \delta(c = y_i))^2 \qquad (5)$$

The above description and the upper part of Fig. 2 are only for a single training episode/iteration. The number of iterations/episodes depends on our manual setting (e.g., 10000) to reach the coverage of the training to f_θ, g_ϕ, and h_γ.

2) Testing/Interface: After the training, f_θ, g_ϕ, and h_γ are fixed without any retraining before or during the testing/interface process. As shown in the lower part of Fig. 2, m ($K \times C$) MI trials with labels are required to be collected as the support set for the implementation of testing/interface for the

target subject. These m samples are not used to update or train the model. They are only feedforwarded into the network to calculate the representative feature (i.e., M_c) of each class for the target subject. Each testing sample without a label is first encoded by f_θ and then compared with representative features by h_γ. The h_γ outputs a relation score for this testing trial in each class, e.g., 0.9 for hand MI and 0.1 for foot MI in the lower part of Fig. 2. The class with the highest score is assigned as the predicted label for this particular testing trial.

Note that the testing sample is not feedforwarded into g_ϕ, as mentioned in subsection II-B.1. It is also worth noting that all the testing trials of the target subject are compared with the same representative features, given that the support set, f_θ, and g_ϕ are all fixed during the testing/interface process.

C. Network Architecture

EEGNet [15] is one of the most popular architectures for MI classification in the BCI community. For the trial embedding module (f_θ), we follow the same architecture of the EEGNet but abandon its dense layers (supervised classification layers). The embedding function (Fig. 3, top left) contains three blocks to extract deep features.

For g_ϕ in this work, we utilize a simple 1D CNN with a kernel size (ks) along the dimension of the sample size (i.e. m) of S. The stride is $(1, 1)$. The zeros $(ks-1, 0)$ are padded prior to the feature vector of the first trial, as shown in Fig. 3 (right). Note that the 1D CNN computes the weighted sum of its input and slides along the feature dimension. Therefore, it conforms to the setting of the proposed g_ϕ that the encoding of the current trial utilizes both the information from its own and proceeding trials. For example, when 1D CNN (assuming $ks = 2$) encodes the feature vector of the second trial (i.e., the third row) in the right part of Fig. 3, it uses the feature vector of the first trial (the second row) and that of the current trial (the third row) as the input.

We adopt the same architecture of the relation module in [19] for h_γ but modify hyper-parameters of several layers to adapt the data dimensionality of MI trials. More concretely, two CNN layers and two dense layers are used and illustrated in Fig. 3 (bottom left).

D. Sampling Techniques for Training Episodes

The support and testing sets in the testing phase are both coming from the target subject. To better emulate this phase, both S and Q in each episode are sampled from the same source subject (constraint A). In addition, another important assumption in the online prediction setting is that the trials of the testing set always occur after those of the support set. We, therefore, introduce one more constraint (constraint B) that the trials in Q are always collected after all of those in S to adapt to this assumption.

In the offline evaluation, we only apply constraint A to the episode sampling, as constraint B is only for the assumption of the online prediction setting. In the online simulation experiments, we tested two different sampling techniques, i.e., (1) Constraint A only and (2) Constraints A and B.

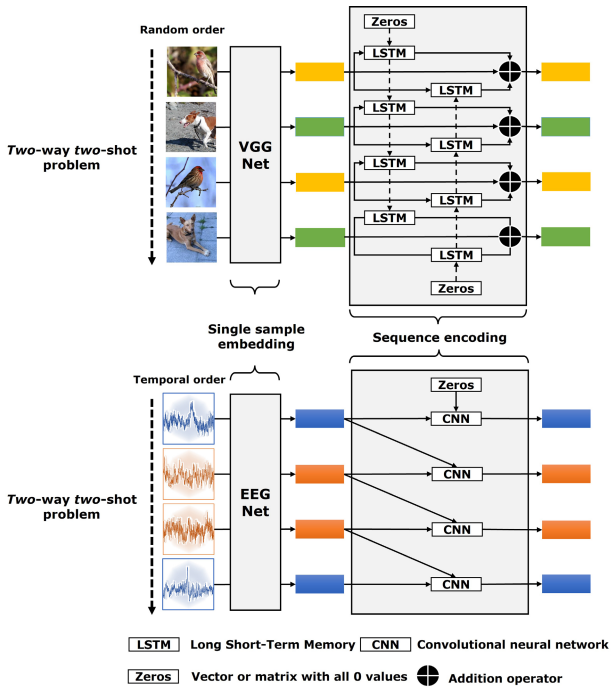

Fig. 4. An illustration of the feature encoding in the support/sample set using the methods in [23] (top, image classification) and the proposed TERL (down, MI classification) for two-way two-shot problems. Rectangles in different colours represent feature vectors in different classes. Both methods contain a single sample feature embedding network and a sequence encoding function. VGGNet [34] and EEGNet [15] are used for the single sample embedding of the approach in [23] and our TERL, respectively, in this illustration. A bidirectional long short-term memory (BiLSTM) is used for the sequence encoding function in [23], taking account of the information from all samples within the support/sample set for the sequence encoding of each sample. The CNN (an implementation example of g_ϕ) is used for sequence encoding in our method and only uses the information from the current trial itself and a single (for the sake of easy illustration; other numbers can also be used depending on the a_i we define) prior trial.

It is unreasonable to apply constraint B only in either offline or online prediction settings, since the temporal order between MI trials makes little sense when they come from different individuals. Therefore, constraint B only is not tested in our study.

E. Advantages of the Proposed Method

The proposed TERL is specifically designed to train a few-shot classifier for the MIBCI. It shows distinct advantages for the MI recognition against the original episode training in [23], initially proposed for the image classification task. First, TERL inherits the main superiority of the original RN [19]. The data-driven nonlinear comparator (i.e., h_γ) is used in our approach instead of relying on the manually selected cosine distance metric in [23].

More notably, the designed TERL is more appropriate than the method proposed in [23] for MIBCI episode training in terms of two aspects below. An intuitive comparison between these two methods for two-way two-shot problems is displayed in Fig. 4.

1) Permutation in support/sample set: The order of the samples in support/sample sets is presumably random

in [23] with no details offered in the study. The dependency of samples between different classes is encoded during the training where the ordering pattern can not be obtained. Alternatively, the MI trials in our framework are sorted in temporal order within each support/sample set to capture temporal patterns, which have been proven highly significant in boosting the MIBCI performance in numerous studies, such as [14], [15], [35], and [36].

2) Embedding scope: The proposed g_ϕ only covers a_i preceding trials in the temporal embedding, i.e., Eq (1), which fits real-world classification setting. On the contrary, the sequence encoding in [23] is based on bidirectional long short-term memory (BiLSTM), where the embedding of the current sample depends on the entire support/sample set. This violates the assumption of temporal pattern that the embedding of the current trial should only have the information from the previous ones.

III. Experiments

We conduct extensive experiments to verify the effectiveness of the proposed method. Source codes for experiments are publicly available.[1]

A. Databases

We use the following four public well-known MIBCI datasets in our experiments. The sequence of the MI tasks is randomized in these datasets.

1) BCI Competition IV-2a (IV-2a) [9]: The dataset was collected from 9 individuals, i.e., A01 to A09, using the device sampling at 250 Hz with 22 EEG channels. A cue-based visual paradigm that consists of four MI classes (left hand (LH), right hand (RH), tongue (TG), and both feet) was deployed in the data collection. Two sessions were conducted for each subject, resulting in a total of 576 MI trials with 144 in each class for each person. These two sessions are treated as a whole set. The temporal order used in the experiments is that the last trial of the first session is collected before the first trial of the second session. The 22-channel EEG signal was epoched at [0, 4] from the starting point of the visual cue until the end of MI. Given the 250 Hz sample rate, each MI trial is a 22×1000 matrix. We conduct experiments of 2- (LH/RH), 3- (LH/RH/TG), and 4-class (all) MI tasks on this dataset. Only the results of 4-class experiments are presented in main text, as this task is the most common one in previous studies. The results of 2- and 3-class tasks are shown in the Supplementary Materials.

2) BCI Competition IV-2b (IV-2b) [9]: This dataset was also recorded from 9 different participants (B01-B09). Only three electrodes, i.e., C3, Cz, and C4, were used to collect the EEG signals at a sample rate of 250 Hz. The dataset includes two MI classes, left hand and right hand, using a visual cue-based paradigm. A total of 720 trials were collected for each subject, with 360 for each class. We use a 4s temporal interval, between the beginning of the visual cue and the end of MI, as a trial in our experiments. Therefore, the data format of one trial is a 3×1000 matrix.

[1] https://github.com/XiuyuHuangsmarthealth/Relation-Learning-Using-Temporal-Episodes-for-Motor-Imagery-Brain-Computer-Interfaces

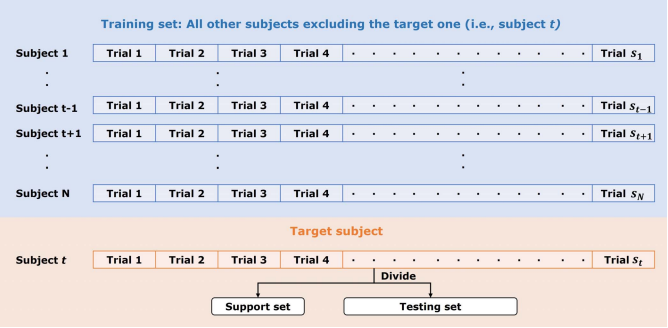

Fig. 5. An illustrative example of splitting the training set and target subject using LOSO scheme. Training set contains all the trials from subjects $1, \ldots, t-1, t+1, \ldots, N$ in the dataset, excluding those from subject t. Subject t is left out as the target subject for model evaluation. N is the number of subjects in the dataset, and t is one of the integers in $[1, N]$. Note that $t-1$ does not exist when $t=1$, neither $t+1$ when $t=N$. s_i denotes the number of trials collected in subject i in the dataset.

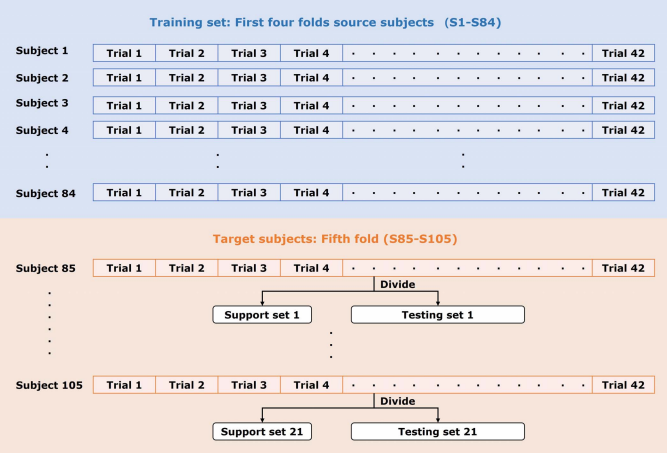

Fig. 6. An illustrative example of 5-fold training and validation on a 2-class task of P-MI. The S85-S105 (fifth split) are the target subjects in this example.

3) BCI Competition III IVa (III-IVa) [13]: This dataset was recorded from five subjects, denoted as *aa*, *al*, *av*, *aw*, and *ay*. Participants perform MI tasks in either right hand or right foot. The total number of trials collected for each subject was 280, with half (i.e., 140) for each class. The EEG signal was collected from 118 channels placed on the scalp according to the international 10-20 system. There are 1000 Hz and 100 Hz versions available for this dataset. We use 100 Hz one in our experiments. According to [37], only 18 channels are selected for our analyzes. Each trial is epoched at [0.5, 2.5], and its dimension is in a size of 18×200.

4) Physionet EEG Motor Movement/Imagery Dataset (P-MI) [12]: The dataset carries EEG recordings from 109 subjects. Data from four subjects are discarded due to the variability in the number of trials, leading to only the signals of 105 subjects (S1-S105) eventually being included in this study. The EEG signals are collected using 64 channels with a sample rate of 160 Hz. Each subject attended three runs for MI of the

left fist (L) against the right fist (R) and three runs for MI of both fists (B) against both feet (F). One run contains 14 trials resulting in 21 trials for each class per subject. A baseline run was also recorded prior to these MI runs. It provides the EEG data in a resting state (RS) where the subjects do not perform any tasks with their eyes open. Referring to the setting in [35] and [38], we conduct the experiments on 2-(L/R), 3- (L/R/RS), and 4-class (L/R/RS/F) MI classification tasks. Consistent with [38], each trial only contains 3 second interval and in a dimension of 64×480.

B. Offline Evaluation

The offline evaluation is based on leave-one-subject-out (LOSO) implementation [39] for IV-2a, IV-2b, and III-IVa. The training data for a target subject consists of the data from all subjects, excluding himself/herself. An illustrative example of splitting the training set and target subject is shown in Fig. 5. In this example, all trials of subjects $1, \ldots, t-1, t+1, \ldots, N$, excluding those of subject t, compose as the training set for model training. N is the number of subjects in the dataset, and t is one of the integers in $[1, N]$. Note that $t-1$ does not exist when $t = 1$, neither $t+1$ when $t = N$. Subject t is left out as the target subject for evaluation. Its trials are divided into a support set and a testing set. It is worth reminding again that the support set is assumed to be annotated and used to calculate the representative features (see subsection II.B). The predictive accuracy of the testing set denotes the model performance. Certainly, all subjects (i.e., from 1 to N) in the dataset are left out once for the evaluation.

The offline evaluation of P-MI inherits the scheme in [35] and [38] using the 5-fold cross-validation. Specifically, 105 subjects (S1-S105) are divided into five splits: S1-S21, S22-S42, S43-S63, S64-S84, and S85-S105. The subjects in each split act as the target ones, while the trials of those in the remaining four splits are used as the training set. The process is recursively implemented five times, with each split tested once. An illustrative example of the 2-class validation for P-MI is displayed in Fig 6. As shown in the figure, the first four splits, S1-S21, S22-S42, S43-S63, and S64-S84, are used as the training set for model training. The fifth split (S85-S105) is used to evaluate the model. Each subject in this split has its own support set and testing set, which are validated on the same model trained by the first four splits. Again, subjects in the splits S1-S21, S22-S42, S43-S63, and S64-S84 are also as the target subjects once for the evaluation. The 3-class and 4-class experiments follow the same protocol as 2-class in the offline evaluation.

In the offline evaluation, it is assumed that all trials are available for every target subject. Therefore, for each target subject, we randomly select m trials (i.e., K trials per class) to formulate the support set. All the remaining trials are used as the testing set; see a two-way $three$-shot example in Fig. 7. The division is repeated five times for each target subject in order to eliminate the random effect of the support/testing division. The mean accuracy of the testing sets across these five times denotes the performance of the model in terms of this target subject.

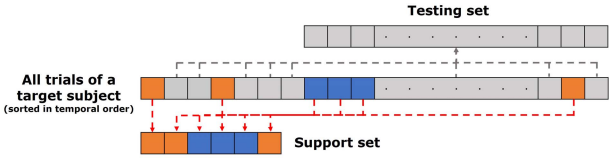

Fig. 7. An example of the division of support and testing sets for a target subject in the offline evaluation for a two-way three-shot problem. Three MI trials in each class are randomly picked out as the support set (with labels). All remaining trials combine as the testing set (to be predicted).

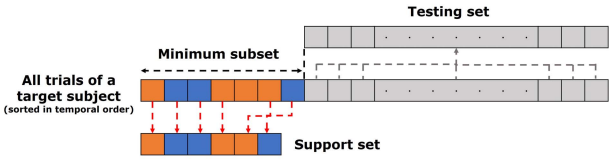

Fig. 8. An example of the division of support and testing sets for a target subject in the online evaluation simulation for a two-way three-shot problem. Three MI trials in each class are randomly picked out as the support set (with labels) from the minimum set at the beginning of data collection. All trials, excluding the minimum set, compose the testing set (to be predicted).

C. Online Evaluation Simulation

The only difference between offline analysis and online evaluation simulation is the support/testing division for the target subject. In the online evaluation simulation, we also randomly select m trials from the target subject to form the support set. However, the selection is only made at the minimum subset at the beginning of the first session to imitate the initial setup of the real-world online BCI system. The minimum subset is defined as the smallest subset containing at least K trials for all classes; see a two-way $three$-shot example in Fig. 8. This implementation is because the data collection is not equally recorded across the entire period. We want to guarantee that the support set can be sampled from the beginning of the first recording session and also contain K trials per class. The remaining trials, excluding the minimum subset, are used as the testing set. Again, the data from other subjects are used as the training set, the same as the offline evaluation. The experiments are repeated five times to decrease the random effect of support set formulation using the minimum subset.

All the RS trials of the P-MI dataset are always recorded at the beginning of the data collection. Therefore, the minimum set at the beginning of the data collection always contains all the RS trials, and no such trials can be assigned to the testing set. In this case, the model performance on 3-class and 4-class tasks using RS-class trials is not validated in the online evaluation simulation. In sum, we only conduct online evaluation simulations on the IV-2a, IV-2b, III-IVa, and 2-class P-MI.

D. Description of Different Approaches

This work compares five baselines and three recent SOTAs with the proposed method. They are described as follows.

1) Baselines:

TABLE I
HYPERPARAMETER SETTING. E AND T ARE THE NUMBERS OF ELECTRODES AND TIME POINTS, RESPECTIVELY, FOR AN MI TRIAL. sc_1 AND sc_2 ARE KERNEL SIZES OF CNN IN BLOCKS 1 AND 3, RESPECTIVELY, IN THE TRIAL EMBEDDING MODULE (f_θ). HD IS THE OUTPUT SIZE OF THE FIRST DENSE LAYER IN THE RELATION MODULE (h_γ)

Hyperparameters	IV-2a	IV-2b	III-IVa	P-MI		
				2-class	3-class	4-class
E	22	3	18	64	64	64
T	1000	1000	200	480	480	480
sc_1	32	32	32	32	32	32
sc_2	16	16	16	16	16	16
HD	32	32	32	32	32	32

a) **Few:** EEGNet trained in a supervised manner by only using the support set.
b) **Source only (SO):** EEGNet trained in a supervised manner by only using all data from other source subjects (i.e., training set).
c) **Combine:** EEGNet trained in a supervised manner by using the data of both support and training sets.
d) **Basic I [29]:** The few-shot model only contains the embedding (f_θ) and relation (h_γ) modules. The generations of the sample and query sets for each episode are completely random from data of all source subjects without constraints A nor B mentioned in subsection II.D.
e) **Basic II [30]:** The few-shot model only contains embedding (f_θ) and relation (h_γ) modules. The sample and query sets are generated with constraint A only. Basic II is an ablation and baseline to the proposed method.

2) SOTAs:

f) **Fine-tuning (FT) [16]:** EEGNet pre-trained in a supervised manner by using all data from source subjects. The last fully-connected layer of the pre-trained model is then fine-tuned by the data of support set.
g) **DDAN [28]:** EEGNet jointly trained by a combined loss of the categorical entropy, maximum mean discrepancy (MMD), and center loss.
h) **DRDA [24]:** This method uses the domain-adversarial training to learn domain-invariant features and the center loss to increase to discriminative power of the model in the target subject.

These three SOTAs are representatives of the most common and effective DL techniques, i.e., fine-tuning, adding an auxiliary penalization in the objective function, and domain-adversarial learning, in the MI supervised transfer learning (TL) and domain adaptation (DA). They were proposed to effectively leverage both source and target data for model training and show a competitive performance in previous studies [16], [24], [28]. In addition, as described in subsection II.D, our approach with two sampling techniques is also tested in this article.

3) Our sampling techniques:

i) **Model A:** The proposed learning framework with all three modules (i.e., f_θ, g_ϕ, and h_γ). The sample and query sets are generated with constraint A only.
j) **Model B:** The proposed learning framework with all three modules (i.e., f_θ, g_ϕ, and h_γ). The sample and query sets are generated with constraints A and B.

Except for Model B, all these methods are implemented in both offline evaluation and online evaluation simulation. Model B is only realized in the online evaluation simulation, because constraint B is meaningless when applied to the offline evaluation. There is no guarantee that the trials of the support set are collected prior to those in the testing set in the offline evaluation. The splitting of the support set and testing set shown in Figs. 7 and 8 are for all the methods in offline evaluation and online evaluation simulation, respectively.

E. Training and Optimization

All experiments are implemented on the Google Colab platform. The computation is accelerated by a Tesla P100 PCIE 16GB GPU provided by the platform. We use an Adam optimizer [40] for network update with a learning rate of 0.001. The number of episodes is 15000 for IV-2a and IV-2b, 2000 for III-IVa (given that the training in III-IVa converges much faster), and 10000, 20000, and 30000 for 2−, 3−, and 4-class P-MI, respectively. The number of K equals 10 for IV-2a, IV-2b, and III-IVa, meaning that the number of m equals 40, 20, and 20, respectively. The value of K is 5 instead for all tasks in P-MI, given that the number of trials in each class is only 21, and we do not want the testing set to be too small. Therefore, m equals 10, 15, and 20 in 2−, 3−, and 4-class tasks. The kernel size (ks) of 1D CNN is set as $m/4$ for IV-2a, IV-2b, and III-IVa, meaning that $ks = 10$, 5, and 5, respectively. ks is set as 5 for all three tasks in P-MI. Other hyperparameters are displayed in Table I. We apply the same configuration for all the subjects within each dataset. The settings described in this subsection are default throughout the article. Several of them are modified accordingly in the following sections to verify the effectiveness of each component of the proposed method.

IV. RESULTS AND ANALYSIS

A. Offline Evaluation

Table II shows the mean classification accuracy and standard deviation of different methods across repetitions in the offline evaluation on the datasets IV-2a, IV-2b, and III-IVa. The last column shows the average classification accuracy and the standard deviation across subjects. Table III displays the mean accuracy across subjects for 2-, 3-, and 4-class P-MI. The best values are highlighted in boldface in both tables. Non-parametric Friedman test and the post-hoc Nemenyi test at level of significance $\alpha = 0.05$ are applied to examinate the statistical difference in performance between methods.

Results show a similar tendency in all four datasets. The models trained only using the support set in a supervised manner have the lowest mean accuracy and large standard deviations across repetitions among all approaches. This result

TABLE II
CLASSIFICATION ACCURACIES (%) OBTAINED IN THE OFFLINE EVALUATION FOR IV-2A, IV-2B, AND III-IVA

Dataset	Method	Subject									Average
		A01	A02	A03	A04	A05	A06	A07	A08	A09	
IV-2a	Few	45.9 ± 1.8	36.5 ± 3.8	47.2 ± 1.9	31.7 ± 7.8	48.1 ± 6.1	37.5 ± 3.2	60.1 ± 5.1	41.3 ± 5.1	51.9 ± 5.0	44.5 ± 8.7
	SO	68.5 ± 1.1	42.7 ± 2.6	77.2 ± 1.6	56.1 ± 1.2	43.5 ± 1.4	57.3 ± 1.5	67.9 ± 0.4	73.7 ± 0.6	61.5 ± 1.3	60.9 ± 12.3
	Combine	69.3 ± 4.4	46.2 ± 2.2	79.8 ± 2.4	56.4 ± 0.9	48.1 ± 6.2	55.9 ± 2.8	70.5 ± 2.4	73.1 ± 2.5	62.2 ± 1.5	62.4 ± 11.6
	Basic I	58.3 ± 1.2	44.3 ± 2.8	72.4 ± 5.3	50.9 ± 3.1	51.0 ± 1.9	55.1 ± 2.4	66.6 ± 2.3	62.6 ± 2.3	58.9 ± 4.8	57.8 ± 8.6
	Basic II	72.2 ± 3.1	47.1 ± 2.3	82.7 ± 0.5	57.3 ± 1.0	44.3 ± 1.2	55.7 ± 2.0	72.2 ± 2.1	73.2 ± 2.0	**70.9 ± 0.6**	64.0 ± 13.2
	FT*	59.7 ± 4.4	43.2 ± 3.2	73.6 ± 2.0	49.6 ± 3.1	**55.8 ± 1.0**	47.8 ± 2.7	66.6 ± 3.3	69.2 ± 2.8	65.3 ± 3.1	59.0 ± 10.5
	DDAN*	71.3 ± 1.3	47.2 ± 2.5	79.8 ± 2.6	57.3 ± 1.7	54.1 ± 1.5	58.9 ± 1.7	74.4 ± 2.2	76.9 ± 2.6	64.8 ± 2.6	65.0 ± 11.3
	DRDA*	72.2 ± 0.7	**49.8 ± 1.6**	81.2 ± 2.9	**59.8 ± 2.3**	50.3 ± 4.2	57.4 ± 0.8	75.7 ± 2.6	74.9 ± 0.4	68.2 ± 5.0	65.5 ± 11.5
	Model A	**74.8 ± 1.5**	48.7 ± 2.0	**84.9 ± 1.1**	58.7 ± 2.4	52.6 ± 2.1	**60.9 ± 2.0**	**75.9 ± 1.8**	**78.3 ± 1.3**	70.4 ± 1.1	**67.2 ± 12.5**
		B01	B02	B03	B04	B05	B06	B07	B08	B09	
IV-2b	Few	55.5 ± 1.8	55.1 ± 3.8	70.0 ± 1.9	76.0 ± 7.8	72.8 ± 6.1	65.6 ± 3.2	67.5 ± 5.1	70.2 ± 5.1	54.8 ± 5	65.3 ± 8.2
	SO	74.6 ± 1.1	68.6 ± 2.6	62.7 ± 1.6	84.9 ± 1.2	84.9 ± 1.4	79.0 ± 1.5	79.5 ± 0.4	71.0 ± 0.6	77.4 ± 1.3	75.8 ± 7.4
	Combine	75.3 ± 0.6	66.9 ± 1.8	61.2 ± 5.3	88.6 ± 1.0	84.6 ± 1.1	81.7 ± 1.2	80.6 ± 0.6	77.3 ± 4.1	76.5 ± 0.9	77.0 ± 8.5
	Basic I	77.4 ± 1.6	70.6 ± 1.3	64.7 ± 1.0	90.7 ± 1.3	85.6 ± 1.1	83.2 ± 0.8	84.5 ± 1.2	82.5 ± 1.1	76.4 ± 1.4	79.5 ± 8.1
	Basic II	75.0 ± 0.8	69.4 ± 0.5	66.0 ± 0.9	91.1 ± 0.9	85.4 ± 1.0	82.3 ± 1.0	84.2 ± 0.4	82.8 ± 0.8	77.3 ± 1.1	79.3 ± 8.1
	FT*	74.4 ± 1.2	67.0 ± 2.0	65.2 ± 1.8	91.1 ± 0.9	82.9 ± 0.4	81.8 ± 1.7	80.5 ± 1.4	79.4 ± 1.3	76.3 ± 0.9	77.6 ± 8.0
	DDAN*	72.6 ± 0.7	65.0 ± 1.4	66.0 ± 2.0	90.6 ± 1.2	81.1 ± 2.9	82.0 ± 3.9	78.6 ± 2.0	81.9 ± 1.3	72.2 ± 1.5	76.7 ± 8.4
	DRDA*	75.0 ± 2.1	68.1 ± 5.0	66.1 ± 3.2	88.4 ± 5.3	79.2 ± 3.9	80.5 ± 1.5	80.7 ± 0.5	81.2 ± 1.9	76.0 ± 0.9	77.2 ± 6.9
	Model A	**77.5 ± 1.0**	**73.7 ± 1.2**	**74.0 ± 0.8**	**93.9 ± 0.1**	**86.7 ± 0.6**	**87.1 ± 0.6**	**86.5 ± 1.0**	**83.4 ± 0.7**	**78.0 ± 0.6**	**82.3 ± 6.9**
		aa		al		av		aw		ay	
III-IVa	Few	67.4 ± 10.3		88.3 ± 1.3		47.6 ± 0.7		65.9 ± 7.4		81.9 ± 2.7	70.2 ± 15.8
	SO	67.7 ± 2.1		86.4 ± 2.5		60.9 ± 2.7		59.6 ± 3.6		76.6 ± 5.4	70.2 ± 11.3
	Combine	71.0 ± 2.4		91.0 ± 1.7		60.4 ± 1.3		69.2 ± 1.3		80.8 ± 2.4	74.5 ± 11.7
	Basic I	71.5 ± 3.1		90.2 ± 1.1		64.9 ± 4.3		68.2 ± 3.2		82.2 ± 1.6	75.4 ± 10.5
	Basic II	73.5 ± 3.8		88.4 ± 5.0		64.8 ± 3.7		**71.2 ± 5.5**		82.8 ± 2.1	76.1 ± 9.4
	FT*	69.1 ± 1.9		89.0 ± 2.1		56.3 ± 1.9		62.8 ± 3.7		72.5 ± 2.9	69.9 ± 12.3
	DDAN*	59.8 ± 2.7		88.0 ± 3.4		57.9 ± 3.1		58.2 ± 2.9		71.2 ± 7.9	67.0 ± 12.9
	DRDA*	76.8 ± 2.4		**92.9 ± 0.7**		56.2 ± 2.9		65.6 ± 1.5		79.0 ± 7.9	74.1 ± 13.9
	Model A	**77.2 ± 2.2**		91.8 ± 0.7		**66.1 ± 1.4**		70.2 ± 2.6		**84.0 ± 1.1**	**77.9 ± 10.4**

The highest accuracies are highlighted in boldface. *Results of FT, DDAN, and DRDA are reproduced according to the description in their papers and open source codes if provided.

TABLE III
CLASSIFICATION ACCURACIES (%) OBTAINED IN THE OFFLINE EVALUATION FOR P-MI

	2-class	3-class	4-class
Few	66.7 ± 13.2	56.5 ± 11.8	44.1 ± 10.8
SO	83.1 ± 12.5	72.3 ± 10.9	65.0 ± 13.0
Combine	81.8 ± 12.3	78.0 ± 11.0	68.7 ± 12.0
Basic I	85.7 ± 11.0	79.0 ± 9.5	67.1 ± 13.1
Basic II	86.9 ± 11.1	81.4 ± 11.0	71.3 ± 14.0
FT*	84.3 ± 11.9	74.0 ± 11.5	67.6 ± 11.6
DDAN*	86.8 ± 10.9	73.6 ± 8.4	67.5 ± 8.2
DRDA*	83.2 ± 11.2	77.5 ± 11.2	68.2 ± 11.1
Model A	**87.4 ± 11.3**	**81.8 ± 11.9**	**74.2 ± 12.0**

The highest accuracies are highlighted in boldface. *Results of FT, DDAN, and DRDA are reproduced according to the description in their papers and open source codes if provided.

matches the description in the Introduction section that training a neural network using such a small size of samples can easily lead to an over-fitting issue and poor performance. Other supervised learning methods leverage the data from other subjects for the training, which usually significantly (all p-values < 0.05, except SO and "Combine" methods on III-IVa) improves the performance compared to the "Few" strategy. It is necessary to draw help from a larger available dataset to increase the model generalization capability when only very few samples from the target subject are accessible.

According to [41], the prediction risk of a classifier on the target domain is lower bound by the sum of domain shift and the prediction risk in the source domain. Theoretically, TL and DA methods can minimize the domain distance, thus improving the model performance in the target domain. However, contradicting our expectation, neither of these methods (i.e., FT, DDAN, and DRDA) achieves a higher average classification accuracy than the "Combine" approach on all four evaluation datasets. In the few-shot setting, there are only very few MI trials from the target subject. These trials can not represent the real distribution of the trials from the target subject. This may be why these methods can not show the superiority in our experiments as reported in their original studies.

Few-shot classifiers mostly have competitive performance with the best supervised learning approaches, except for the Basic I on IV-2a. Models can obtain high generalization ability via the few-shot learning method when the target subject only contains few samples per class. We can also see that Basic II performs significantly better than Basic I on IV-2a and 4-class P-MI (p-value < 0.05). This finding suggests the necessity of sampling each episode from the same source subject, as individuals usually produce MI signals with different characteristics. The proposed TERL shows at least 3.2% (IV-2a), 2.8% (IV-2b), 1.8% (III-IVa), 0.5% (P-MI 2-class), 0.5% (P-MI 3-class), and 2.9% (P-MI 4-class) improvements in the average classification accuracy against other few-shot

TABLE IV
CLASSIFICATION ACCURACIES (%) OBTAINED IN THE ONLINE EVALUATION SIMULATION FOR IV-2A, IV-2B, AND III-IVA

Dataset	Method	Subject									Average
		A01	A02	A03	A04	A05	A06	A07	A08	A09	
IV-2a	Few	40.9 ± 5.9	32.0 ± 1.8	47.7 ± 2.7	39.0 ± 3.3	38.6 ± 2.4	39.6 ± 2.9	54.2 ± 2.3	45.0 ± 2.6	46.5 ± 6.3	42.6 ± 6.5
	SO	65.1 ± 3.1	45.2 ± 3.4	78.7 ± 3.8	52.8 ± 2.4	48.0 ± 5.9	43.9 ± 4.6	67.4 ± 1.4	72.8 ± 0.8	62.9 ± 2.0	59.6 ± 12.6
	Combine	70.0 ± 2.6	44.3 ± 3.8	77.7 ± 1.9	58.2 ± 3.6	45.1 ± 5.2	51.3 ± 2.0	66.0 ± 3.2	76.5 ± 2.6	**65.8 ± 2.7**	61.7 ± 12.6
	Basic I	57.5 ± 1.0	46.6 ± 2.2	70.3 ± 1.9	51.8 ± 1.5	49.5 ± 0.8	50.5 ± 1.7	67.8 ± 2.5	59.6 ± 1.2	58.1 ± 2.4	56.9 ± 8.2
	Basic II	70.5 ± 0.6	46.4 ± 2.7	79.0 ± 1.1	51.9 ± 2.8	46.7 ± 2.4	52.3 ± 0.9	70.4 ± 2.8	72.5 ± 1.6	63.5 ± 1.4	61.5 ± 12.3
	FT*	69.4 ± 1.3	47.0 ± 1.4	80.6 ± 2.0	57.4 ± 1.9	43.5 ± 3.4	52.5 ± 0.8	68.4 ± 2.0	**77.5 ± 1.2**	63.7 ± 3.4	62.2 ± 13.1
	DDAN*	69.5 ± 1.7	42.4 ± 1.0	76.0 ± 1.0	55.7 ± 1.3	51.0 ± 1.8	50.1 ± 0.9	67.6 ± 1.8	73.0 ± 3.4	63.1 ± 4.2	60.9 ± 11.6
	DRDA*	70.8 ± 4.0	44.8 ± 1.7	77.1 ± 1.0	**59.1 ± 3.2**	41.3 ± 4.5	**56.5 ± 2.2**	72.7 ± 1.6	75.1 ± 2.9	62.6 ± 1.1	62.2 ± 13.0
	Model A	**75.0 ± 0.5**	**49.4 ± 1.1**	**83.0 ± 0.8**	58.3 ± 1.4	**52.9 ± 2.6**	52.7 ± 1.3	72.5 ± 2.0	76.1 ± 1.2	65.3 ± 2.3	**65.0 ± 12.2**
	Model B	73.8 ± 1.3	46.7 ± 1.2	80.6 ± 0.7	54.7 ± 1.7	48.6 ± 1.9	55.4 ± 1.4	**73.6 ± 1.5**	73.4 ± 1.8	62.9 ± 1.7	63.3 ± 12.5
		B01	B02	B03	B04	B05	B06	B07	B08	B09	
IV-2b	Few	51.8 ± 2.3	56.7 ± 2.4	44.7 ± 10.8	51.0 ± 2.4	51.3 ± 2.7	50.6 ± 2.6	60.0 ± 3.7	58.4 ± 8.2	56.9 ± 5.3	53.5 ± 4.8
	SO	76.1 ± 1.4	69.5 ± 1.6	62.9 ± 2.9	85.0 ± 2.0	85.2 ± 0.7	80.9 ± 2.1	80.5 ± 1.7	69.6 ± 0.7	77.7 ± 0.5	76.4 ± 7.6
	Combine	76.6 ± 1.3	70.6 ± 1.3	62.4 ± 1.4	87.1 ± 1.1	85.8 ± 1.5	79.3 ± 1.3	80.4 ± 0.7	70.1 ± 1.1	78.9 ± 0.5	76.8 ± 7.9
	Basic I	77.4 ± 1.4	69.5 ± 0.6	63.5 ± 1.8	90.7 ± 0.7	85.7 ± 0.4	83.2 ± 2.0	85.1 ± 0.6	**83.0 ± 0.1**	75.8 ± 0.8	79.3 ± 8.6
	Basic II	75.7 ± 0.4	70.6 ± 1.1	66.5 ± 1.5	92.2 ± 0.6	86.5 ± 0.8	82.7 ± 1.0	84.8 ± 0.4	**83.0 ± 0.2**	76.1 ± 1.0	79.8 ± 8.2
	FT*	74.3 ± 1.5	65.0 ± 3.4	**69.7 ± 3.0**	92.5 ± 1.3	86.1 ± 1.6	**85.4 ± 2.8**	80.9 ± 2.0	80.0 ± 1.1	**79.1 ± 1.8**	79.2 ± 8.5
	DDAN*	72.7 ± 2.2	63.8 ± 0.6	64.6 ± 3.4	85.3 ± 0.7	80.0 ± 2.2	79.4 ± 2.4	79.8 ± 1.2	79.6 ± 2.1	74.1 ± 1.6	75.5 ± 7.4
	DRDA*	74.0 ± 2.2	71.0 ± 0.6	67.8 ± 3.4	87.4 ± 0.7	82.3 ± 2.2	81.0 ± 2.4	85.8 ± 1.2	80.3 ± 2.1	74.1 ± 1.6	78.2 ± 6.8
	Model A	76.6 ± 0.6	**71.2 ± 1.5**	69.2 ± 0.4	**94.5 ± 0.2**	86.0 ± 0.8	85.2 ± 0.8	**86.1 ± 0.5**	82.8 ± 0.4	77.8 ± 0.4	**81.0 ± 8.1**
	Model B	**77.7 ± 0.6**	69.9 ± 0.5	68.2 ± 0.9	93.6 ± 0.3	**86.6 ± 0.4**	83.7 ± 0.9	84.4 ± 0.3	82.2 ± 0.1	76.7 ± 0.8	80.3 ± 8.1
		aa		al		av		aw		ay	
III-IVa	Few	68.6 ± 3.8		88.4 ± 2.4		57.6 ± 3.0		76.0 ± 5.2		72.8 ± 6.9	72.7 ± 11.2
	SO	67.6 ± 2.0		87.9 ± 3.3		50.9 ± 0.7		53.4 ± 0.5		67.9 ± 9.9	65.5 ± 14.8
	Combine	70.0 ± 2.4		91.8 ± 1.9		61.7 ± 1.3		58.9 ± 2.9		82.8 ± 2.9	73.0 ± 14.0
	Basic I	72.5 ± 2.8		89.1 ± 2.7		63.2 ± 1.8		65.7 ± 3.6		81.6 ± 2.1	74.4 ± 10.9
	Basic II	75.7 ± 2.1		90.5 ± 1.7		61.5 ± 2.5		66.5 ± 3.8		83.9 ± 1.3	75.6 ± 12.0
	FT*	65.5 ± 4.5		87.3 ± 1.8		61.5 ± 3.5		56.2 ± 1.5		76.8 ± 4.0	69.5 ± 12.5
	DDAN*	67.8 ± 4.4		88.3 ± 2.2		59.6 ± 3.3		61.9 ± 5.8		75.7 ± 2.0	70.7 ± 11.7
	DRDA*	73.4 ± 4.2		89.7 ± 1.4		57.9 ± 3.6		56.6 ± 2.0		80.7 ± 6.1	71.7 ± 14.4
	Model A	**77.2 ± 0.8**		**92.8 ± 0.8**		**66.9 ± 1.1**		**76.4 ± 1.7**		**84.8 ± 0.7**	**79.6 ± 9.7**
	Model B	75.9 ± 2.4		91.2 ± 1.3		65.1 ± 2.0		74.4 ± 2.9		84.6 ± 1.8	78.2 ± 10.0

The highest accuracies are highlighted in boldface. *Results of FT, DDAN, and DRDA are reproduced according to the description in their papers and open source codes if provided.

TABLE V
CLASSIFICATION ACCURACIES (%) OBTAINED IN THE ONLINE EVALUATION SIMULATION FOR P-MI

Method	2-class
Few	65.3 ± 15.2
SO	80.6 ± 11.9
Combine	82.1 ± 12.7
Basic I	83.5 ± 14.3
Basic II	84.7 ± 14.6
FT*	83.2 ± 12.2
DDAN*	82.6 ± 13.4
DRDA*	84.7 ± 10.9
Model A	**86.3 ± 13.1**
Model B	85.4 ± 14.2

The highest accuracy is highlighted in boldface. *Results of FT, DDAN, and DRDA are reproduced according to the description in their papers and open source codes if provided.

learning methods (i.e., Basic I and II) in the experiments of different evaluated tasks. These results show that the temporal information of the training episodes is important for the MI recognition task. There is valuable latent information between trials of the MI brain activities. It is also worth mentioning that the proposed method performs better than other recent SOTA approaches [35], [38], [42] on 3- and 4-class tasks of the P-MI dataset. To our best knowledge, it achieves new SOTA accuracy for these two tasks. For the 2-class task, our method has a slightly worse (1% in average classification accuracy) performance than the best approach, EEGSym, proposed in [11]. EEGSym is based on a large inception network trained by four large MI datasets. A long-time calibration, consisting of an over 6-hour pre-training and a 12-minute fine-tuning, is required to fit a model for the target user. Alternatively, our method reaches a similar competitive outcome with significantly less calibration time. It only needs a 1-minute pre-training and can be directly applied to the target subject without fine-tuning processes. For the datasets IV-2a, IV-2b, and III-IVa, we use a different training and testing data division from the original one in the BCI competition because of considering the temporal order of the collected trials. Given using a different experiment protocol, it may not be fair to directly compare the performance of our method to other published results [10].

B. Online Evaluation Simulation

Table IV presents the mean classification accuracy of multiple approaches in the online evaluation simulation on the datasets IV-2a, IV-2b, and III-IVa. Table V shows the corresponding results for 2-class P-MI. The best values are highlighted in boldface in both tables. Statistical analysis is

TABLE VI
RETRAINING TIME IN THE INITIAL SETUP FOR TARGET SUBJECT

Methods	Retraining time (seconds)					
	IV-2a	IV-2b	III-IVa	P-MI		
				2-class	3-class	4-class
Few	15.6	12.6	11.6	4.6	6.2	9.2
SO	0	0	0	0	0	0
Combined	570.5	173.5	121.7	55.3	80.6	161.4
FT	34.1	30.3	14.4	2.4	2.9	3.1
DDAN	865.7	699	161.2	44.3	71.6	120.2
DRDA	620.9	158.2	112.7	20.6	26.8	36.5
Few-shots	0	0	0	0	0	0

also performed again using the Friedman test and the post-hoc Nemenyi test at level of significance $\alpha = 0.05$.

Our approaches (Models A and B) perform the best on all four evaluation datasets. Model A has improvements of over 2.8% (IV-2a), 1.2% (IV-2b), 4.0% (III-IVa), and 1.6% (P-MI 2-class) in the average classification accuracy compared to other approaches. Our two sampling schemes do not significantly differ in accuracy (i.e. all p-values > 0.05), but Model A slightly outperforms Model B in all four datasets. This result is against our expectation that the Model B using episodes in which samples in Q are collected after those in S better micic the testing interface and should perform better than Model A. The reason may be since the sampling technique used in Model B significantly reduce the possible combinations of S and Q from source samples, which decreases the generalizability of our model. We also observe that there are accuracy discrepancies in methods (e.g., "Few", DRDA, and Model A; p-values < 0.05) between offline evaluation and online evaluation simulation by comparing the Tables II and IV. Given these discrepancies, it may not be sufficient to perform only the offline analysis to truly understand the effectiveness of an approach on the real-world BCI. Online analysis should also be necessary as several subjects may not produce consistent MI during the usage.

The retraining time in the setup stage for the target subject also has a significant impact on the user experience. The shorter time means a better user experience. We present the retraining time of each method when the data from the target subject is available in Table VI. It is noted that the training of most supervised learning methods, except the SO baseline, can be started only until the data from the target subject are available. The "Few" baseline and FT strategy only utilize a small data set (i.e., support set) for the model training, so they both require short retraining time. Alternatively, the "Combine" baseline, DDAN, and DRDA demand the combination of source and target data to train a model from scratch in the setup stage for the target subject. They need a much more extended period. Although this training time is not excessively long, it will exponentially grow when we adopt a more complex backbone like Deep ConNet proposed in [14]. On the contrary, the few-shot learning classifiers are purely trained by the data from the source subjects without any retraining process after the target data is collected. It can be applied to the target subject in a plug-and-play manner, significantly increasing the user's experience in the real-world MIBCI. Although the SO method does not require retraining either, it does not offer a similar level of accuracy with few-shot classifiers, especially compared with the proposed approach.

C. K-Way Setting

More data available from the target subject usually means a better model performance in the target domain. In the real-world BCI scenario, it is assumed that labeled samples from the target subject are especially limited. Referring to setting in [29] and [30], we investigate how the number of K up to 20 impacts the model performance. The conditions, $K = \{5, 10, 15, 20\}$, are analyzed on the Fine-tuning, DRDA, and Model A in IV-2a, IV-2b, and III-IVa datasets. We also include zero-shot (i.e., $K = 0$) experiments for these three methods in which no target data are available. For the zero-shot problem, ten trials per class from one random source subject are selected as the support set for the implementation of Model A, and the samples of all other remaining source subjects are used as the training set. Fine-tuning and DRDA become the SO method without target data. Non-parametric Friedman test, followed by the post-hoc Nemenyi test, is applied to test the statistical difference in the performance between methods. The significance level is set at $\alpha = 0.05$. We do not show the results of K-way experiments for P-MI datasets here, as only 21 trials per class are collected for each subject. The variation of the results is large even $K = 10$ (i.e., only 11 trials per class for testing) for all methods, which does not offer more insight into how the K impacts models in the target subject. The K-way experiment in this subsection is based on the LOSO evaluation scheme as the offline analysis. To make the analysis more informative, K-way experiments are implemented using three different training seeds, including 0, 1, and 2, for the initialization of weights. All experiments regarding each training seed are conducted once. The K-way experiments of TERL using another validation scheme are displayed in the Supplementary Materials to demonstrate the stability of the proposed method.

Box plots with strip plots for the classification accuracy of three approaches across subjects with different K are shown in Fig. 9. As expected, the performance of both transfer learning (Fine-tuning) and domain adaptation (DRDA) supervised methods gradually improves with an increasing value of K. Our method performs similarly in the few-shot scheme when $K > 0$ in all the datasets. Significant outperformance of the proposed TERL against other approaches is more frequently found when K is small, while it even has slightly worse performance than DRDA when $K = 15$ and 20 in IV-2a, as well as $K = 20$ in III-IVa. Our method gradually loses the advantage in accuracy with a larger value of K compared to the supervised approaches. It is intuitive to understand the reason behind these results. Our method classifies the unknown trials based on their relations with the sum of features of each class (M_c) in the support set. In the prediction perspective, the feature representatives with a sum of 5 trials do not have much difference from a sum of 20 trials, although the sum of a large number of trials may decrease the negative influence of certain noisy trials on the prediction. Alternatively, such supervised

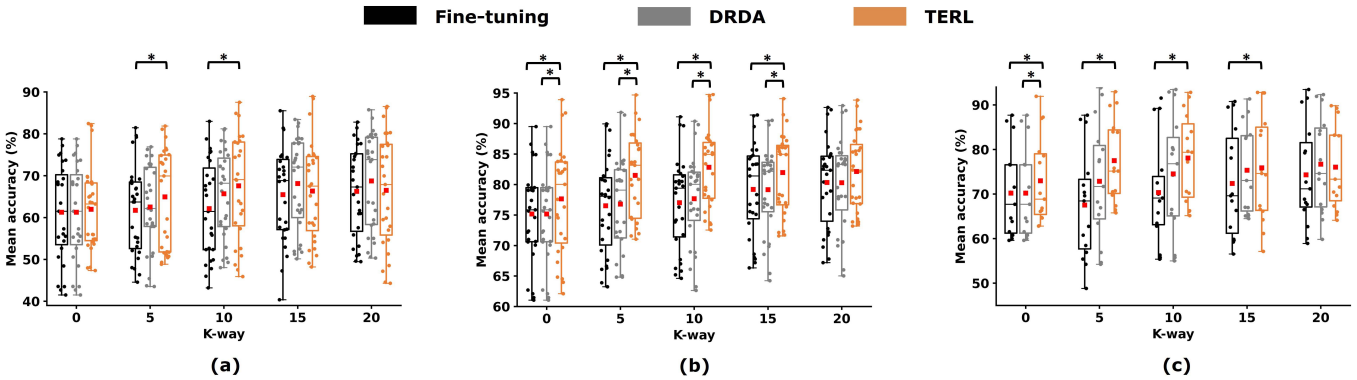

Fig. 9. Box plots with strip plots for the classification accuracy of three approaches with different values of K for IV-2a (a), IV-2b (b), and III-IVa (c). Red squares are the mean values for boxes. The star (*) denotes significant difference (p-value < 0.05) found between two methods by Nemenyi post-hoc test. Each dot represents an accuracy per subject per training seed.

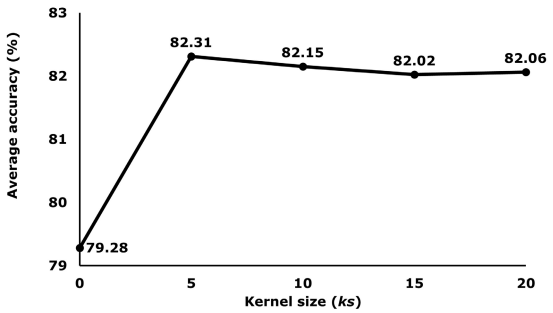

Fig. 10. Classification accuracies in IV-2b obtained with different values of ks in g_ϕ. The condition of $ks = 0$ means that 1D CNN (i.e., g_ϕ) is not used in the episode embedding with only f_θ remained in feature encoding.

methods, i.e., either transfer learning or domain adaptation ones, address the domain shift between source and target subjects. More target trials available usually mean a better sampling distribution to represent the actual distribution of MI trials produced by the target subject. It is more appropriate to choose these supervised methods for BCI development when we have enough MI trials from the target user. However, it is also worthwhile to highlight that our method, as a few-shot classifier, shows an advantage in the condition of having limited target trials. In addition, we can still see the merit of our method when K equals 0. It has a higher mean accuracy than the other two approaches in all three datasets, although no significant significance is observed in some tasks. Learning to compare still outperforms learning to classify when it comes to the subject-independent classifier.

D. Temporal Kernel Size

The kernel size of 1D CNN controls the number of preceding trials in the embedding of the current trial in one episode. We carry out experiments to verify the sensitivity of this significant hyper-parameter. We vary the kernel size in a range of $\{0, 5, 10, 15, 20\}$ to represent different levels of temporal length, i.e., zero, short, medium, long, and all preceding trials. The condition of $ks = 0$ means that 1D CNN (i.e., g_ϕ) is not used in the episode embedding with only f_θ remained in feature encoding. The experiments were performed using the offline setting on the IV-2b dataset.

Fig. 10 presents the averaged accuracy across subjects of our approach with different kernel sizes. It is observed that only using f_θ for feature embedding without considering the temporal information is not a good option. The model achieves the lowest accuracy. We can also see that the performance of our approach remains stable when the kernel size varies between 5 to 20, showing that our method is insensitive to the change in the kernel size. The shortest kernel length achieves the highest accuracy and performs similarly to other levels. The reason behind this result may be due to the considerable temporal shift between trials. The "further previous" trials may not offer extra benefits on the embedding of the current trial to promote the classification performance.

E. Temporal Kernel Weights

We use a 1D CNN (g_ϕ) to encode the temporal information of the sample set in each training episode. The embedding of each trial in one episode is calculated by a weighted sum of itself and preceding trials when using the 1D CNN. In order to visualize the temporal patterns captured by the proposed method, we plot two heat maps (Fig. 11) for the weights of nine 1D CNNs in the models for B01-B09 subjects in two different conditions, respectively. As the kernel size of 1D CNN is 5 for IV-2b, each heat map contains 5 rows. The first condition is that the trials in the sample set are sorted in temporal order, which is the implementation of the proposed method. The second condition is that we do not perform any sorting of the trials in the sample set. The weights are normalized by the $l2$ norm to be within a similar scope.

In the left part of Fig 11, we can clearly see that the weight of the current trial is always the largest among all weights. It is natural for this phenomenon to occur, as the embedding should be encoded with most of its own information. We also make another interesting observation that the weights close to the encoding position (i.e., bottom one) tend to be larger. These 1D CNNs are all randomly initialized and updated by data. It is evident that MI trials within the episode have temporal patterns, and the proposed 1D CNN can capture such

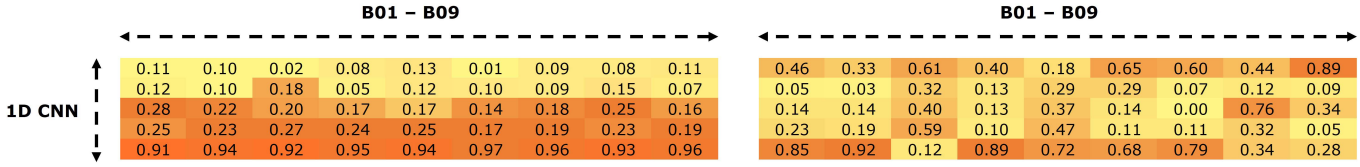

Fig. 11. Heat maps for weights of the 1D CNN (g_ϕ) across B01 to B09 in two different conditions. Left: the sample set sorted in temporal order in training. Right: the sample set sorted in random order in training. The kernel size of 1D CNN is 5 for IV-2b. Thus, both heat maps contain 5 rows.

valuable latent information. Otherwise, the weights should have a random permutation as the right part of the Fig 11. This comparison stresses the necessity of temporal encoding in the episode training for the MI recognition task.

V. Conclusion

This paper introduces a temporal episode learning approach for developing the MIBCI with only a minor initial setup for the target subject. It leverages an episode-based function g_ϕ to encode the temporal information in each episode, significantly different from previous studies that only focus on encoding the information within each trial. The proposed learning framework combined with designated trial sampling techniques can enhance the classification accuracy by 2.9% and 4.0% accuracy in offline evaluation and online evaluation simulation, respectively, compared to the basic few-shot and supervised learning under the same experimental settings. In addition, it is also worthwhile to highlight that our approach does not require retraining in the target subject, which can effectively promote the user's experience. These findings indicate that the proposed TERL is a promising method by offering an accurate prediction and convenient setup for MIBCI development.

We use a simple but effective 1D CNN to encode the temporal information of the episode. It only contains a few parameters (i.e. 5 or 10) to be optimized in training. As mentioned, the function g_ϕ that we introduce can also adapt to other more complex architectures, such as the short-term memory (LSTM) [43] and attention module [44]. They are also worth exploring in future studies under the framework of g_ϕ.

In this study, we only focus on a synchronous setting for the BCI without feedback. In fact, the synchronous BCI [45] with user's feedback (i.e. label) immediately after each trial is also a popular setting in real-world applications. It is mainly performed in motor rehabilitation programs for patients, e.g., post-stroke survivors. It is also meaningful to apply our method to this synchronous setting by predicting each trial based on a few of its nearest previous trials. Another interesting application of g_ϕ in future studies is to improve the performance of the asynchronous MIBCI. This type of MIBCI is the common one applied in a real-world scenario, where users can choose whenever they perform MI tasks. An asynchronous MIBCI system usually has intentional control (IC, performing MI task) and non-control (NC, idle period) states [46]. The g_ϕ can be designed to encode the information of both the current point and the preceding signals before the current time point regardless of whether they are from the same IC or previous ICs but to ignore signals in NCs. This setting enables g_ϕ to encode temporal patterns in asynchronous BCI and potentially boosts its performance.

References

[1] Z. Feng et al., "Design a novel BCI for neurorehabilitation using concurrent LFP and EEG features: A case study," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 5, pp. 1554–1563, Sep. 2022.

[2] L. Qian, Z. Feng, H. Hu, and Y. Sun, "A novel scheme for classification of motor imagery signal using stockwell transform of CSP and CNN model," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 3673–3677.

[3] M. Rashid et al., "Current status, challenges, and possible solutions of EEG-based brain-computer interface: A comprehensive review," *Frontiers Neurorobotics*, vol. 14, p. 25, Jan. 2020, doi: 10.3389/fnbot.2020.00025.

[4] A. A. Frolov et al., "Post-stroke rehabilitation training with a motor-imagery-based brain-computer interface (BCI)-controlled hand exoskeleton: A randomized controlled multicenter trial," *Frontiers Neurosci.*, vol. 11, p. 400, Jul. 2017, doi: 10.3389/fnins.2017.00400.

[5] A. Rezeika, M. Benda, P. Stawicki, F. Gembler, A. Saboor, and I. Volosyak, "Brain–computer interface spellers: A review," *Brain Sci.*, vol. 8, no. 4, p. 57, Apr. 2018.

[6] D. Huang, K. Qian, D. Fei, W. Jia, X. Chen, and O. Bai, "Electroencephalography (EEG)-based brain-computer interface (BCI): A 2-D virtual wheelchair control based on event-related desynchronization/synchronization and state control," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 20, no. 3, pp. 379–388, May 2012.

[7] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K. R. Müller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Signal Process. Mag.*, vol. 25, no. 1, pp. 41–56, Jan. 2008.

[8] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b," *Frontiers Neurosci.*, vol. 6, p. 39, Mar. 2012, doi: 10.3389/fnins.2012.00039.

[9] M. Tangermann et al., "Review of the BCI competition IV," *Frontiers Neurosci.*, vol. 6, p. 55, Jan. 2012, doi: 10.3389/fnins.2012.00055.

[10] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: A systematic review," *J. Neural Eng.*, vol. 16, no. 5, Aug. 2019, Art. no. 051001, doi: 10.1088/1741-2552/ab260c.

[11] S. Perez-Velasco, E. Santamaria-Vazquez, V. Martinez-Cagigal, D. Marcos-Martinez, and R. Hornero, "EEGSym: Overcoming inter-subject variability in motor imagery based BCIs with deep learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 1766–1775, 2022.

[12] A. L. Goldberger et al., "Physiobank, physiotoolkit, and physionet," *Circulat.*, vol. 101, no. 23, pp. e215–e220, 2000, doi: 10.1161/01.CIR.101.23.e215.

[13] B. Blankertz et al., "The BCI competition III: Validating alternative approaches to actual BCI problems," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 153–159, Jun. 2006.

[14] R. T. Schirrmeister et al., "Deep learning with convolutional neural networks for EEG decoding and visualization," Mar. 2017, *arXiv:1703.05051*.

[15] 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, Oct. 2018, Art. no. 056013.

[16] K. Zhang, N. Robinson, S.-W. Lee, and C. Guan, "Adaptive transfer learning for EEG motor imagery classification with deep convolutional neural network," *Neural Netw.*, vol. 136, pp. 1–10, Apr. 2020.

[17] M. Riyad, M. Khalil, and A. Adib, "Incep-EEGNet: A convnet for motor imagery decoding," in *Image and Signal Processing*, A. El Moataz, D. Mammass, A. Mansouri, and F. Nouboud, Eds. Cham, Switzerland: Springer, 2020, pp. 103–111.

[18] 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 *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 2958–2965.

[19] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[20] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4, no. 4. New York, NY, USA: Springer, 2006, p. 738.

[21] X. Huang, N. Zhou, and K.-S. Choi, "A generalizable and discriminative learning method for deep EEG-based motor imagery classification," *Frontiers Neurosci.*, vol. 15, 2021, Art. no. 760979, doi: 10.3389/fnins.2021.760979.

[22] F. Lotte, M. Congedo, A. Lécuyer, L. Fabrice, and B. Arnaldi, "A review of classification algorithms for EEG-based brain–computer interfaces," *J. Neural Eng.*, vol. 4, pp. 1–25, Jan. 2007.

[23] O. Vinyals, C. Blundell, T. Lillicrap, K. kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, vol. 29, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2016. [Online]. Available: https://proceedings.neurips.cc/paper/2016/file/90e1357833654983612fb05e%3ec9148c-Paper.pdf

[24] H. Zhao, Q. Zheng, K. Ma, H. Li, and Y. Zheng, "Deep representation-based domain adaptation for nonstationary EEG classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 535–545, Feb. 2021.

[25] X. Hong et al., "Dynamic joint domain adaptation network for motor imagery classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 556–565, 2021.

[26] I. Goodfellow et al., "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97%b1afccf3-Paper.pdf

[27] A. M. Azab, L. Mihaylova, K. K. Ang, and M. Arvaneh, "Weighted transfer learning for improving motor imagery-based brain–computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 7, pp. 1352–1359, Jul. 2019.

[28] W. Hang et al., "Cross-subject EEG signal recognition using deep domain adaptation network," *IEEE Access*, vol. 7, pp. 128273–128282, 2019.

[29] S. An, S. Kim, P. Chikontwe, and S. H. Park, "Few-shot relation learning with attention for EEG-based motor imagery classification," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, p. 10.

[30] S. Bhosale, R. Chakraborty, and S. K. Kopparapu, "Calibration free meta learning based approach for subject independent EEG emotion recognition," *Biomed. Signal Process. Control*, vol. 72, Feb. 2022, Art. no. 103289. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1746809421008867

[31] R. Ning, C. L. P. Chen, and T. Zhang, "Cross-subject EEG emotion recognition using domain adaptive few-shot learning networks," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2021, pp. 1468–1472.

[32] A. Salekin and N. Russo, "Understanding autism: The power of EEG harnessed by prototypical learning," in *Proc. Workshop Med. Cyber Phys. Syst. Internet Med. Things*, New York, NY, USA, May 2021, pp. 12–16, doi: 10.1145/3446913.3460317.

[33] T.-J. Luo, C.-L. Zhou, and F. Chao, "Exploring spatial-frequency-sequential relationships for motor imagery classification with recurrent neural network," *BMC Bioinf.*, vol. 19, no. 1, pp. 1–18, Dec. 2018.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA, 2015, pp. 1–14.

[35] H. Dose, J. S. Møller, H. K. Iversen, and S. Puthusserypady, "An end-to-end deep learning approach to MI-EEG signal classification for BCIs," *Expert Syst. Appl.*, vol. 114, pp. 532–542, Dec. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417418305359

[36] T. Xu et al., "E-key: An EEG-based biometric authentication and driving fatigue detection system," *IEEE Trans. Affect. Comput.*, early access, Dec. 9, 2021, doi: 10.1109/TAFFC.2021.3133443.

[37] S.-H. Park, D. Lee, and S.-G. Lee, "Filter bank regularized common spatial pattern ensemble for small sample motor imagery classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 2, pp. 498–505, Feb. 2018.

[38] X. Wang, M. Hersche, B. Tomekce, B. Kaya, M. Magno, and L. Benini, "An accurate EEGNet-based motor-imagery Brain–Computer interface for low-power edge computing," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Jun. 2020, pp. 1–6.

[39] O. Y. Kwon, M. H. Lee, C. Guan, and S. W. Lee, "Subject-independent brain-computer interfaces based on deep convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 3839–3852, Oct. 2020.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[41] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218306684

[42] C.-C. Fan, H. Yang, Z.-G. Hou, Z.-L. Ni, S. Chen, and Z. Fang, "Bilinear neural network with 3-D attention for brain decoding of motor imagery movements from the human EEG," *Cognit. Neurodyn.*, vol. 15, no. 1, pp. 181–189, Feb. 2021.

[43] S. Hochreiter and J. J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 80–1735, 1997.

[44] A. Vaswani et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon et al., Eds. Red Hook, NY, USA: Curran Associates, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1%c4a845aa-Paper.pdf

[45] V. Peterson et al., "Transfer learning based on optimal transport for motor imagery brain-computer interfaces," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 2, pp. 807–817, Aug. 2022.

[46] C.-H. Han, K.-R. Müller, and H.-J. Hwang, "Brain-switches for asynchronous brain–computer interfaces: A systematic review," *Electronics*, vol. 9, no. 3, p. 422, Mar. 2020.