Scheduled Knowledge Acquisition on Lightweight Vector Symbolic Architectures for Brain-Computer Interfaces

Yejia Liu[1], Shijin Duan[2], Xiaolin Xu[2], and Shaolei Ren[1]
[1]UC Riverside
[2]Northeastern University
{yliu807@ucr.edu},{duan.s@northeastern.edu},{x.xu@northeastern.edu},{sren@ece.ucr.edu}

ABSTRACT

Brain-Computer interfaces (BCIs) are typically designed to be lightweight and responsive in real-time to provide users timely feedback. Classical feature engineering is computationally efficient but has low accuracy, whereas the recent neural networks (DNNs) improve accuracy but are computationally expensive and incur high latency. As a promising alternative, the low-dimensional computing (LDC) classifier based on vector symbolic architecture (VSA), achieves small model size yet higher accuracy than classical feature engineering methods. However, its accuracy still lags behind that of modern DNNs, making it challenging to process complex brain signals. To improve the accuracy of a small model, knowledge distillation is a popular method. However, maintaining a constant level of distillation between the teacher and student models may not be the best way for a growing student during its progressive learning stages. In this work, we propose a simple scheduled knowledge distillation method based on curriculum data order to enable the student to gradually build knowledge from the teacher model, controlled by an α scheduler. Meanwhile, we employ the LDC/VSA as the student model to enhance the on-device inference efficiency for tiny BCI devices that demand low latency. The empirical results have demonstrated that our approach achieves better tradeoff between accuracy and hardware efficiency compared to other methods.

KEYWORDS

brain-computer interface, vector symbolic architecture, knowledge distillation

ACM Reference Format:
Yejia Liu[1], Shijin Duan[2], Xiaolin Xu[2], and Shaolei Ren[1]. 2024. Scheduled Knowledge Acquisition on Lightweight Vector Symbolic Architectures for Brain-Computer Interfaces. In *Proceedings of tinyML Research Symposium (tinyML Research Symposium'24).* ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION

A brain-computer interface (BCI) allows for direct communication between the human brain and an external device without the need for physical movement [17, 18, 50, 58]. The Electroencephalogram (EEG), as a typically non-invasive neuroimaging technique to measure and record the electrical activity of the brain, has been widely used in the BCI applications [34, 59]. The deep neural networks (DNNs) have shown promising results in extracting spatial-temporal dynamics from EEG signals, and improved classification accuracy compared to the classical feature engineering methods [52, 57, 66]. However, the intensive computation required by DNNs in inference can result in high latency in real-time EEG-based BCIs, which are intended to be lightweight [42, 60, 67]. The latency, for example, can pose a challenge for disabled persons using BCI-controlled prosthetic limbs, making it difficult to perform fine motor tasks such as picking up small objects or typing on a keyboard. Moreover, in some implantable BCI devices, power constraints are even more stringent, with the power needing to stay under 15-40mW to comply with FDA, FCC, and IEEE guidelines [29, 36], which immediately rules out many conventional DNN-based classifiers.

Due to the hardware efficiency with massive processing parallelism, the binary hyperdimensional computing based on the vector symbolic architecture (HDC/VSA) stands out in the edge inference paradigm [27, 31, 32, 44]. In the binary HDC/VSA, objects are encoded into long, binary vectors in a high dimensional space. By exploiting the algebraic properties of a small set of operators, the observed instance vectors belonging to the same class are aggregated, allowing for the creation of a meaningful symbolic class representation in the end, for inference of new objects [27, 41]. Although lightweight, the binary HDC/VSA suffers from low accuracy and large model size as a result of its rudimentary training procedure and high dimensionality of vectors [13, 14, 16]. The recent proposed low-dimensional computing (LDC) [14] alleviates these issues by utilizing a partially binary neural network (BNN) [47] to hash samples into binary codes with dimensionality less than 100. Featuring a systematic training based on backpropagation, the accuracy of LDC is improved with 100× smaller model size, along with faster inference speed, compared to the HDC. Nonetheless, the accuracy gap between LDC and a modern neural network still remains as a challenge, preventing its adoption in the BCIs, which involve complex neural signals as the input.

One popular technique to improve the accuracy of a small model is knowledge distillation (KD), where a larger "teacher" network supervises a smaller "student" model, where the knowledge is distilled by the soft probabilities with a (usually static) hyperparameter α to balance the distillation term and the classification term [22]. However, the distillation is not always advantageous, particularly when there is a large capacity gap between big teachers and small students [11, 68]. Drawing inspiration from curriculum learning [7], [62, 69], in which the student is gradually trained using samples ordered in an easy-to-hard manner. The recent CTKD curriculums the distillation temperature by adversarial training to

tinyML Research Symposium'24, April 2024, San Francisco, CA

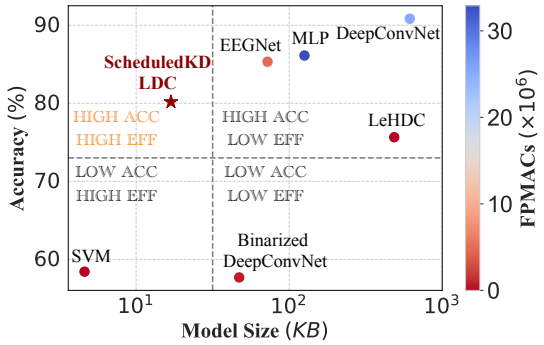

Figure 1: Comparison of accuracy and inference efficiency of various methods on the Motor Imagery dataset. The ScheduledKD-LDC has achieved a good tradeoff between accuracy and efficiency, considering the model size and the number of FPMAC operations.

guide the learning process for the student [35]. Although demonstrated as effective, their training requires meticulous planning and they have not explored how to efficiently transfer knowledge to a much smaller architecture like binary LDC.

In this paper, we propose a simple yet effective approach to control the procedure of knowledge distillation from a complex teacher network to a lightweight LDC for EEG-based BCIs. We refer to the proposed approach as **ScheduledKD-LDC**, which uses an α scheduler that decreases exponentially during the distillation process, with curriculum data order. Intuitively, as the students gradually build up knowledge by learning from the confident predictions on simpler data points from the teacher, they could rely less on the teacher over time. Meanwhile, more independent learning at the later stage enhances the student model to develop its own data representations to generalize to unseen examples. As shown in Figure 1, the ScheduledKD-LDC achieves a good balance between accuracy and efficiency compared to other methods. Our empirical results indicate that it consistently outperforms other methods on the evaluated EEG datasets. As an east-to-use method, we believe the ScheduledKD-LDC is a favorable option to realize efficient edge intelligence for real-time BCIs.

2 PRELIMINARIES

We define the input space as $\mathcal{X}$ and the label space as Y, where $|Y| = C$. Let $f : \mathcal{X} \times \Theta \rightarrow R^C$ be a classifier, parameterized by $\theta \in \Theta$. It outputs a categorical predictive distribution over Y, $\hat{p}(y = i|\mathbf{x}) = \sigma_i(f(\mathbf{x}, \theta))$, where $\sigma_i(\mathbf{z}) := \exp(z_i)/\sum_j \exp(z_j)$ represents the softmax function, and $z := f(\mathbf{x}, \theta)$ is referred to as the logits. For simplicity, we use f^t to denote the teacher model and f^s to represent the student model. Knowledge distillation [22] uses the soft output (logits) of one or multiple large models as the teacher and transfers the knowledge to a small student model, where the student model minimizes a combination of two loss objectives as shown in the Eq.(1),

$$\mathcal{L} := \alpha \mathcal{L}_{KD}(\mathbf{z}^s, \mathbf{z}^t) + (1 - \alpha)\mathcal{L}_{NLL}(\mathbf{z}^s, y), \tag{1}$$

where $\mathcal{L}_{KD}$ is the distillation term to encourage the student model to resemble teacher's responses for data examples, while $\mathcal{L}_{NLL}$ is the normal cross-entropy loss between the logits $\mathbf{z}^s$ and the label y. These two terms are balanced by the α, which is a static hyperparameter in most existing works [9, 15, 63, 65].

Inspired by the significance of arranging information in human learning process, the curriculum training involves regulating the data order to adjust a model's learning trajectory [6, 38, 55]. A scoring function t is used to determine the difficulty of each data example. If $t(\mathbf{x_j}, y_j) > t(\mathbf{x_i}, y_i)$ for two data samples $\mathbf{x_i}$ and $\mathbf{x_j}$, we would say x_j is more difficult than x_i. In this work, we consider the real-valued loss of a reference model that is trained on the same set of data points as our scoring function, $\mathbf{t}(\mathbf{x_i}, y_i) = \ell(f_\theta(\mathbf{x_i}, y_i))$. In essence, presenting the data in an easy-to-hard sequence is known as *curriculum* data order, while we use the term *anti-curriculum* to describe the ranking of data from difficult to easy, and explicitly referring to a random order as *random* in this work.

3 STUDENT: LOW-DIMENSIONAL VECTOR SYMBOLIC ARCHITECTURE

In the **architecture level**, we employ the low-dimensional classifier (LDC) based on the vector symbolic architecture (VSA) as the student model for the BCI applications which demand low latency.

3.1 Hyperdimensional Computing/Vector Symbolic Architecture (HDC/VSA)

The HDC/VSA represents symbolic concepts using high-dimensional distributed vectors that coexist in a shared space, providing context for each other [41, 64]. To achieve efficient hardware implementation, binary-valued hypervectors composed of $\{-1, 1\}^D$ are used, where D is the vector dimension. In addition, the architecture employs a small predefined set of operators on these hypervectors with high processing parallelism, such as Binding $\bigotimes$ [1] and Bundling $\bigoplus$[2], which can be implemented by simple hardware circuits such as AND, OR gates and adders, as explained in [25, 30, 49].

In a standard classification task using the HDC/VSA, instances are *hashed* into long binary vectors. As an example, an image $\mathbf{x}_i$ with features $F = [f_1, f_2, \ldots, f_N]$, and each feature with discrete values $V_{f_i} = [v_1, v_2, \ldots, v_M]$ can be constructed symbolically. For instance, an image in the MNIST [12] dataset consisting of 784 pixels, where each pixel contains 256 shades of gray, can result in $N = 784$ and $M = 256$. A data example x_i is expressed by $\mathbf{x}_i := \text{sgn}(\bigoplus_{j=1}^{N} f_j \bigotimes V_{f_j})$ with a final dimensionality of 256. During training, vectors from the same class are combined to create a symbolic class representation. During inference, a test image undergoes the same hashing process, and then the Hamming distance between the encoded vector and each class representation is calculated to determine the classification result [25, 27, 49].

3.2 Low-dimensional Computing (LDC) Classifier

The HDC/VSA blends the benefits of connectionist distributed representation and structured symbolic representation, where the representations can be composed, probed, and transformed by a set of hardware-efficent math operations [26, 41]. However, the HDC/VSA

[1] e.g., $(v \otimes w) \in \{-1, 1\}^D$ and $(v \otimes w)_i = v_i w_i$.
[2] e.g., $(\oplus_{k=1}^{m} w_k)_d = \sum_{k=1}^{m} (w_k)_d$ for all $d \in \{1, 2, ..., D\}$.

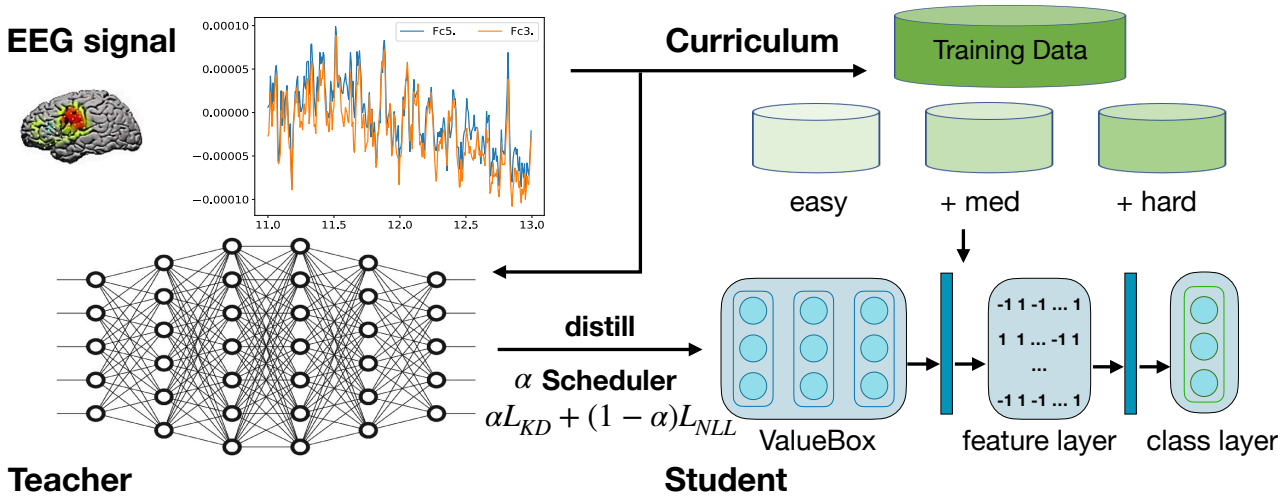

Figure 2: Overview of the ScheduledKD-LDC. We use an α scheduler and curriculum data order to enable the student model to gradually distill knowledge from the complex teacher model. The student model, low-dimensional computing classifier (LDC), is based on the vector symbolic architecture for efficient on-device inference.

architecture has two main issues: the *large model size* brought by the high dimensionality of vectors, where D is typically on the order of $10,000$; and the *low accuracy* due to the simple heuristic hashing process used to generate hypervectors [13, 16]. To address these problems, the low-dimensional computing classifier (LDC) has been proposed [14, 37].

As shown in the Student Model of Figure 2, the LDC uses a partially *binary neural network* in its ValueBox which encodes the input to low-dimensional vectors through deep hashing to generate trainable F and V_{f_i} for a sample. Additionally, the LDC allows for the feature vector dimension D_{f_i} to be a multiple of the value dimension $D_{V_{f_i}}$, enabling dimensionality reduction. The empirical results in [14] have shown that the LDC classifier with $D < 100$ can still achieve comparable or even better accuracy than the HDC.

4 SCHEDULING THE KNOWLEDGE DISTILLATION: SCHEDULEDKD-LDC

It can be challenging for a student model to replicate the predictions $\mathbf{z}^t$ of a teacher model when there is a significant disparity in their model capacities [11, 56, 68]. This presents a particular problem in the context of small brain-computer interface (BCI) devices, where the employed student model like LDC has limited model size and computational complexity considering fast inference. To mitigate this issue, we propose a simple yet effective approach called the ScheduledKD-LDC. It integrates both an α scheduler, which schedules the α in Eq. (1) and a data curriculum to jointly regulate the distillation process. An overview of the ScheduledKD-LDC is provided in the Figure 2.

Data Level Given the limited computation and data representation capability of the lightweight LDC, presenting examples in a curated order can help in building representations step-by-step, starting from simpler concepts and gradually incorporating more intricate ones to capture complex dynamics in BCI datasets. We therefore adopt a *curriculum data ordering* strategy in our approach, where we divide the training data into three pools being {Easy}, {Easy, Medium Hard} and {Easy, Medium Hard, Hardest}. The higher percentage allocation of training data to the easy pool (e.g. [60%, 70%] in the experiments), allows the model to establish a strong initialization by assimilating foundational knowledge from the teacher. Conversely, the hard examples, which are usually more challenging or represent edge cases, receive a smaller percentage allocation (e.g. [10%, 20%]), as the hardest pool.

Learning Procedure However, the curriculum data order alone is not sufficient to address the problem of the student model struggling to learn from more complex teacher models effectively. We therefore also introduce an α scheduler to *manage the distilling level* from the teacher to the student model. Initially, a higher value of α is used to emphasize the influence of the teacher, as the teacher model can provide more accurate and reliable predictions on the easier examples, which allows the student model to build up its knowledge base by learning from the teacher's confident predictions. As the student model becomes more sophisticated, we decrease the value of α to let it learn more by itself. The rationale of using an α scheduler is two-fold: 1) the small student model cannot comprehend the entire knowledge of the much more complex teacher net [11]; 2) some hard examples may challenge even the teacher model, where the wrong prediction can lead to poor performance if using the same level of distillation strength. By allowing the more mature student model to learn from itself, it can develop its own data representations that may generalize better to unseen examples.

When it comes to the choice of the α scheduler, we consider several desirable properties. First is the *gradual transition*. A gradual shift helps prevent sudden disruptions and provides a more stable learning process for the student model. Secondly, *flexibility* is important to cater to different tasks and dataset. For example, some tasks may require longer guidance from the teacher, while others may benefit from faster independence of the student. Thus, the scheduler should allow customization to accommodate these varying needs. We therefore propose using the exponential scheduler due to its smoother change compared to the linear one, as well

Algorithm 1 Scheduled Knowledge Distillation

Input: Training data $\{x_i, y_i\}_{i=1}^{I}$; Total training epoch H; Pretrained teacher model f^t with θ_t; Student model f^s with randomly initialized θ_s; Initial balancing weight α; Difficulty ranking function t; Decay step k; Decay rate γ; Change point P; Order $o \in$ {"curriculum", "random", "anti-curriculum"}.
Output: Trained student model f^s.
Rank data: $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_I) \leftarrow \text{sort}(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_I, s, o)$
while $h < H$ **do**
  if $h \geq P$ and $h \,\%\, k = 0$ **then**
    Exponentially decrease α: $\alpha \leftarrow \alpha \times \gamma^{\lceil \frac{h}{r} \rceil}$
  end if
  Update θ_s based on the curriculum: $\theta_s^h \leftarrow$ train-one-epoch$(\theta_s^{h-1}, \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_I\})$
end while

as its customization capability compared to a static α. Specifically, after the Pth epoch, the value of α is exponentially decreased by $\gamma^{\lceil \frac{h}{r} \rceil}$, where γ is the decay rate, h represents the epoch number and r is the scaling factor. Additionally, our empirical results demonstrate that using the exponential α scheduler yields higher accuracy than directly optimizing α by parameterizing it, as the difference in magnitudes between the values of $\mathcal{L}_{KD}$ and $\mathcal{L}_{NLL}$ heavily biases the training process towards prioritizing one loss over the other, leading to worse performance. It is also worth mentioning that for more challenging data, smaller values of P in general leads to a slightly higher accuracy through our observation. We outline our algorithm in Algorithm 1.

5 EXPERIMENTS

5.1 Experimental Setup

Datasets we have considered three EEG datasets in experiments. The Motor Imagery EEG signals were recorded from participants who were instructed to perform or imagine performing one of five movements: eyes closed, both feet, both fists, left fist, and right fist [54]. The X11 and S4b datasets from the BCI competition IIIb benchmark [46] involves classifying left- and right- hand movements based on EEG signals. The ERN dataset from [39] includes 26 participants completing a P300 speller task.

Evaluation Metrics In the evaluation, we take into account both the efficiency and accuracy. Specifically, we assess the binary multiply-accumulate (BMAC) operations [67], floating point multiply-accumulate (FPMAC) operations [4], and model size required for inference to determine the efficiency of evaluated methods. The BMAC operations can be executed using XNORs and population counts (popcnt) in a highly hardware-efficient manner [67]. Therefore, models dominated by BMACs typically exhibit significantly improved inference speed and reduced model size compared to the ones governed by FPMACs [14, 67].

Baselines The EEGNet is a compact deep neural network specially designed for EEG-based tasks [34], Our experimental DeepConvNet is composed of five convolutional layers, followed by a fully connected layer. In addition, we have binarized the conv layers of these two deep networks for a more comprehensive comparison. These binarized models were respectively named the Binarized-DeepConvNet and the Binarized-EEGNet. Also, we considered the Multi-Layer Perceptron (MLP) and the Support Vector Machines (SVM) in our experiments. The LeHDC is based on the HDC/VSA architecture but trains it with a systematic learning strategy [13]. Besides the standard knowledge distillation method for the LDC (KD-LDC), we also have compared with the CTKD [35], where the curriculum is applied on the distilling temperature by an adversarial manner to control the information transfer.

Implementation Details During training, we set the learning rate for the Motor Imagery dataset to 0.005, decaying by a factor of 0.1 every 50 steps, with a batch size of 1000. For the X11 and S4b datasets, the learning rate remains at 0.005, with a step size of 60 and a batch size of 256. For the LeHDC model, we set the feature dimension D_{f_i} and value dimension $D_{V_{f_i}}$ as 4000 and 4, respectively, while 128 and 4 for the LDC-based models (i.e. LDC, KD-LDC and ScheduledKD-LDC). We opt for DeepConvNet for the Motor Imagery and ERN datasets, and EEGNet for the X11 and S4b benchmark, as the teacher model, respectively. The change point P is 100 in Motor Imagery and ERN, while 75 for X11 and S4b. In curriculum, we set the easiest 65% of data examples as the easy pool, the easiest 80% as the {easy + medium hard} pool, while the easiest 95% samples as the {easy + medium hard + hardest} pool for the Motor Imager and ERN datasets. In X11 and S4b, the percentages are 70%, 90% and 100%.

5.2 Main Results

Table 1 shows the main results of accuracy and efficiency across different methods among the datasets. From the accuracy perspective, the DeepConvNet and MLP have achieved the highest accuracy. However, their accuracy comes at the cost of heavy FPMAC operations, and requires over 40 times larger model size compared to the ScheduledKD-LDC. Regarding the model efficiency, although the Binarized-DeepConvNet and LeHDC are also dominated by efficient BMACs operations, the ScheduledKD-LDC outperforms them by a large margin in terms of accuracy. As for the inference model size, SVM has the smallest number, but its accuracy is ~20% lower than the ScheduledKD-LDC on the evaluated tasks. In addition, the ScheduledKD-LDC consistently outperforms the other knowledge distillation methods like the plain KD LDC and CTKD w/ LDC across all evaluated EEG datasets. In general, ScheduledKD-LDC has better balanced accuracy and efficiency on the EEG datasets compared to other methods. Note that methods like EEGNet show BMACs of 0 as they only involve floating-point computations. Conversely, for methods like LDC, which only entail binary operations in inference, they have 0 FMACs.

5.3 Analysis of the αcheduler

We show the results of using different α setups without data curriculum in the Figure 3. The experiment results have demonstrated that utilizing an α scheduler to regulate the distillation level during the training process is more effective than using a static α. It supports our belief that as the student model gains proficiency in tackling the task, it requires less knowledge distillation from a much more complex teacher model, in order to independently generate data representations using its own understanding based on the already good initialization. Furthermore, as observed from the Figure 3 (a) and (b), the exponential α scheduler generally performs slightly

Table 1: The ScheduledKD-LDC provides a better tradeoff between accuracy and inference efficiency compared to other methods. Note that models dominated by BMACs are typically more hardware-computation efficient than those dominated by FPMACs, as the BAMCs can be implemented in a massively parallel fashion in platforms like FPGA [67].

Dataset	Method	Accuracy (%)	BMACs ($\times 10^6$)	FPMACs ($\times 10^6$)	Model Size (KB)
Motor Imagery	EEGNet [34]	85.33±1.25	0	4.81	72.22
	DeepConvNet [34]	92.83±1.21	0	25.33	613.82
	Binarized-DeepConvNet	57.68±2.21	24.56	0.76	47.15
	SVM (HALO) [29]	58.42±0.45	0	1.02×10^{-3}	4.60
	MLP	86.12±1.44	0	32.90	125.88
	LeHDC	76.02±0.44	4.06	0	527.81
	LDC	77.18±0.89			
	KD-LDC	77.89±0.73			
	CTKD [35] w/ LDC	78.85±0.62	**0.13**	**0**	**16.89**
	ScheduledKD-LDC	**80.17**±0.83			
X11 and S4b	EEGNet [34]	80.04±1.09	0	5.98	105.38
	Binarized-EEGNet	56.14±1.83	5.76	0.21	12.77
	DeepConvNet	83.71±2.07	0	28.99	892.01
	SVM (HALO) [29]	53.55±0.43	0	1.50×10^{-3}	6.01
	LeHDC	68.64±0.22	5.93	0	754.68
	LDC	69.16±1.04			
	KD-LDC	69.68±0.83			
	CTKD [35] w/ LDC	70.22±0.64	**0.19**	**0**	**24.15**
	ScheduledKD-LDC	**71.83**±0.77			
ERN	EEGNet [34]	82.84±1.04	0	4.77	69.78
	Binarized-EEGNet	59.63±1.74	4.59	0.18	5.31
	DeepConvNet	86.67±1.34	0	27.68	632.56
	SVM (HALO) [29]	55.80±0.33	0	1.12×10^{-3}	4.38
	LeHDC	72.63±0.45	4.59	0	597.81
	LDC	73.34±0.87			
	KD-LDC	73.86±0.73			
	CTKD [35] w/ LDC	74.42±0.60	**0.15**	**0**	**19.13**
	ScheduledKD-LDC	**75.57**±0.62			

better than the linear α scheduler across various temperatures. In contrast, using parameterizing α leads to the poorest accuracy. This is primarily due to the significant difference in magnitudes between the values of $\mathcal{L}_{KD}$ and $\mathcal{L}_{NLL}$ throughout the training process. Consequently, optimizing the parameter α tends to heavily favor one loss over the other, leading to suboptimal performance. We attribute the improvements to the smoother and gradual decay, as well as the fine-grained exploration during the early stage of the exponential change, as opposed to the linear one.

Table 2: Efficacy of curriculum training on the Motor Imagery.

	Static α	Linear α	**Expo** α	param α
Curri KD-LDC	**78.04**±0.60	**79.57**±0.86	**80.17**±0.83	**72.83**±0.56
KD-LDC	77.89±0.73	78.59±0.64	78.92±0.57	70.72±0.47
Anti-curri KD-LDC	73.93±0.81	73.84±0.74	74.65±0.70	67.72±0.83
Curri LDC		**77.20**±0.57		
LDC		77.18±0.89		
Anti-curri LDC		73.43±1.20		

5.4 Efficacy of Curriculum Data Order

In the Table 2, we present results of using different data curriculum methods. Firstly, our experiments reveal that curriculum data order helps in the knowledge distillation setting. However, if not under the knowledge distillation setting, the curriculum training does *not* significantly improve accuracy [61], as demonstrated by comparing the results of LDC and Curri LDC. Second, our experiments show that anti-curriculum (as adopted in Anti-curri KD-LDC), which ranks and trains data from difficult to easy, adversely affects accuracy, lowering it by approximately ~4% compared to KD-LDC, which uses random data ordering, on the Motor Imagery dataset. Lastly, combining the curriculum data order (as in Curri KD-LDC) with the exponential α scheduler yields the best performance.

In Table 3 (a), we show the results of using loss-based rankings from the pretrained teacher model versus the pretrained student model. We observe that using the teacher's loss to order data results in significantly worse accuracy compared to using the student's loss. One plausible explanation could be that the teacher and student models have different perceptions of the difficulty of the same

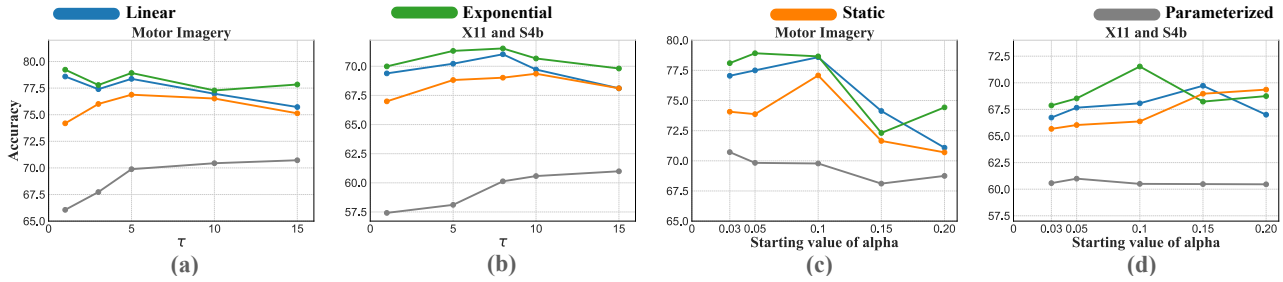

Figure 3: Comparison of different α setups, *without* employing data curriculum in the KD setting on the Motor Imagery, X11 and S4b datasets. (a), (b): With different temperatures τ. (c), (d): With different starting values of α.

Table 3: Data ordering analysis on the Motor Imagery dataset. (a) Accuracy comparison when data ordered by loss of teacher model vs. student model; (b) Loss-based ranking intersection between teacher and student model.

	Order by Teacher Loss	**Order by Student Loss**
ScheduledKD LDC	74.14±0.66	**80.17**±0.83
Curri LDC	70.67±0.58	**77.20**±0.57
Curri LDC w/ KD	74.34±0.61	**78.04**±0.60

(a)

	Overlapped Rank (%)
Hardest 30%	30.06
Hardest 50%	50.25
Hardest 70%	70.72

(b)

data examples. In fact, the Table 3 (b) reveals a low overlap in rankings between teacher and student models, highlighting the need of scheduled learning to bridge the gap and reduce mismatch.

6 RELATED WORKS

In BCIs, real-time operation with minimal signal-acquisition-to-output delay is crucial, making computation complexity and model efficiency a focus in research [1, 5, 8, 28, 33, 40, 53]. The [43] has reduced the calibration cost of brain signals by leveraging previously acquired EEG signals and projecting the new signals into a shared latent space. However, there has been a lack of measurements such as latency or model size to quantify their efficiency improvements. In [60], they have proposed to decode and classify signal without storing them to enhance the response speed of BCI applications, but the models in their work are still computationally-expensive large deep nets. Using classic lightweight feature engineering models such as SVM in BCIs to meet latency or computation constraints can be a simple solution, but their accuracy can be unsatisfactory due to the limited computation capabilities [29, 60, 70].

Motivated by the observation that human brain operates on high dimensional data, the HDC/VSA has emerged [27]. It offers a promising alternative for handling noisy time-series data such as brain signals, as it integrates learning capability and memory functions. Several recent studies have used the HDC/VSA in biosignal tasks for improved inference efficiency [21, 45, 48]. Despite a few attempts to enhance its accuracy, the HDC/VSA still lacks satisfactory performance [13, 16, 23, 24]. The recently proposed LDC offers improved accuracy and efficiency, with a model size 100 times smaller during inference compared to HDC [14, 37]. However, directly applying LDC to EEG datasets, known for strong noise, can yield unsatisfactory accuracy [21].

Knowledge distillation is a common practice to achieve model size compression and maintain accuracy in small architectures [22]. The standard knowledge distillation has been shown to improve the performance of student models in various applications [2, 3, 10]. However, severe prediction distribution mismatch between teacher and student models can occur, especially when there is a large gap between their model capacities [11, 56]. Existing works have proposed using intermediate features to transfer learned representations from the teacher to the student. However, this approach requires high computational cost and storage sizes [19, 20, 51, 68]. In comparison, our proposed method ScheduledKD-LDC introduces no additional intermediate between the teacher and student model. The work closest to us is [35], where their proposed CTKD curriculums the distilling temperature. In contrast, our ScheduledKD-LDC directly adjusts the weights of soft targets to control the influence of the teacher's prediction, where the exponentially decreased α provides the student with more confidence to generate its own data representations as it progresses towards maturity. Our empirical evaluation demonstrates the ScheduledKD-LDC achieves higher accuracy than the CTKD on the EEG benchmarks.

7 DISCUSSION AND FUTURE WORKS

In this work, we propose the ScheduledKD-LDC to regulate the distillation level by α and the order of the training data by curriculum, leading to improved knowledge transfer and accuracy on the evaluated EEG benchmarks. Instead of using classical feature engineering methods or DNNs, we opt for the LDC/VSA classifier for high efficiency with low inference computational cost and smaller memory footprint for tiny BCI devices demanding low latency.

Limitation and Future Works We focus on the EEG-based BCI benchmarks, but it would be worthwhile to explore the use of other brain signals like invasive Electrocorticography (ECoG) and Functional Magnetic Resonance Imaging (fMRI) in future studies. While our proposed ScheduledKD-LDC has demonstrated improved accuracy compared to methods with similar model size (or MACs), it still falls short compared to modern DNNs in terms of accuracy. We therefore aim to explore more ways to bridge the accuracy gap between our method and DNNs. Additionally, we hope to further investigate on-chip power consumption measurements during inference for BCIs on realworld tiny devices.

8 ACKNOWLEDGMENT

Yejia Liu and Shaolei Ren were supported in part by the U.S. NSF grant CNS-2007115. Shjin Duan and Xiaolin Xu were supported in part by the U.S. NSF grant CNS-2326597.

REFERENCES

[1] Swati Aggarwal and Nupur Chugh. Review of machine learning techniques for eeg based brain computer interface. *Archives of Computational Methods in Engineering*, pages 1–20, 2022.

[2] Taichi Asami, Ryo Masumura, Yoshikazu Yamaguchi, Hirokazu Masataki, and Yushi Aono. Domain adaptation of dnn acoustic models using knowledge distillation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5185–5189, 2017.

[3] Ye Bai, Jiangyan Yi, Jianhua Tao, Zhengkun Tian, and Zhengqi Wen. Learn spelling from teachers: Transferring knowledge from language models to sequence-to-sequence speech recognition, 2019.

[4] Mohamed Asan Basiri M and Noor Mahammad Sk. An efficient hardware-based higher radix floating point mac design. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 20(1):1–25, 2014.

[5] Kais Belwafi, Fakhreddine Ghaffari, Ridha Djemal, and Olivier Romain. A hardware/software prototype of eeg-based bci system for home device control. *Journal of Signal Processing Systems*, 89:263–279, 2017.

[6] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery.

[7] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning.

[8] Wooseok Byun, Minkyu Je, and Ji-Hoon Kim. An energy-efficient domain-specific reconfigurable array processor with heterogeneous pes for wearable brain–computer interface socs. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 69(12):4872–4885, 2022.

[9] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. Online knowledge distillation with diverse peers, 2019.

[10] Wei-Chun Chen, Chia-Che Chang, Chien-Yu Lu, and Che-Rung Lee. Knowledge distillation with feature maps for image classification, 2018.

[11] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation, 2019.

[12] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[13] Shijin Duan, Yejia Liu, Shaolei Ren, and Xiaolin Xu. Lehdc: Learning-based hyperdimensional computing classifier, 2022.

[14] Shijin Duan, Xiaolin Xu, and Shaolei Ren. A brain-inspired low-dimensional computing classifier for inference on tiny devices, 2022.

[15] Rasool Fakoor, Jonas Mueller, Nick Erickson, Pratik Chaudhari, and Alexander J. Smola. Fast, accurate, and simple models for tabular data via augmented distillation, 2020.

[16] Lulu Ge and Keshab K. Parhi. Classification using hyperdimensional computing: A review, 2020.

[17] Xiaotong Gu, Zehong Cao, Alireza Jolfaei, Peng Xu, Dongrui Wu, Tzyy-Ping Jung, and Chin-Teng Lin. Eeg-based brain-computer interfaces (bcis): A survey of recent studies on signal sensing technologies and computational intelligence approaches and their applications. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(5):1645–1666, 2021.

[18] Christoph Guger, Brendan Z Allison, and Aysegul Gunduz. *Brain-computer interface research: a state-of-the-art summary 10*. Springer, 2021.

[19] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation, 2019.

[20] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons, 2018.

[21] Michael Hersche, José del R. Millán, Luca Benini, and Abbas Rahimi. Exploring embedding methods in binary hyperdimensional computing: A case study for motor-imagery based brain-computer interfaces, 2018.

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[23] Mohsen Imani, Samuel Bosch, Sohum Datta, Sharadhi Ramakrishna, Sahand Salamat, Jan M Rabaey, and Tajana Rosing. Quanthd: A quantization framework for hyperdimensional computing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(10):2268–2278, 2019.

[24] Mohsen Imani, Samuel Bosch, Mojan Javaheripi, Bita Rouhani, Xinyu Wu, Farinaz Koushanfar, and Tajana Rosing. Semihd: Semi-supervised learning using hyperdimensional computing. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8. IEEE, 2019.

[25] Mohsen Imani, John Messerly, Fan Wu, Wang Pi, and Tajana Rosing. A binary learning framework for hyperdimensional computing. In *2019 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pages 126–131, 2019.

[26] Pierre-Alexandre Kamienny, Guillaume Lample, Sylvain Lamprier, and Marco Virgolin. Deep generative symbolic regression with monte-carlo-tree-search, 2023.

[27] Pentti Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive computation*, 1(2):139–159, 2009.

[28] Young Ho Kang, Dongjae Kim, and Sang Wan Lee. Meta-bci: Perspectives on a role of self-supervised learning in meta brain computer interface. In *2022 10th International Winter Conference on Brain-Computer Interface (BCI)*, pages 1–5. IEEE, 2022.

[29] Ioannis Karageorgos, Karthik Sriram, Ján Veselý, Michael Wu, Marc Powell, David Borton, Rajit Manohar, and Abhishek Bhattacharjee. Hardware-software co-design for brain-computer interfaces. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pages 391–404, 2020.

[30] Geethan Karunaratne, Manuel Le Gallo, Giovanni Cherubini, Luca Benini, Abbas Rahimi, and Abu Sebastian. In-memory hyperdimensional computing, 2020.

[31] Denis Kleyko, Mike Davies, Edward Paxon Frady, Pentti Kanerva, Spencer J Kent, Bruno A Olshausen, Evgeny Osipov, Jan M Rabaey, Dmitri A Rachkovskij, Abbas Rahimi, et al. Vector symbolic architectures as a computing framework for emerging hardware. *Proceedings of the IEEE*, 110(10):1538–1571, 2022.

[32] Denis Kleyko, Dmitri Rachkovskij, Evgeny Osipov, and Abbas Rahimi. A survey on hyperdimensional computing aka vector symbolic architectures, part ii: Applications, cognitive models, and challenges. *ACM Computing Surveys*, 55(9):1–52, 2023.

[33] Onur Erdem Korkmaz, Onder Aydemir, Emin Argun Oral, and Ibrahim Yucel Ozbek. An efficient 3d column-only p300 speller paradigm utilizing few numbers of electrodes and flashings for practical bci implementation. *PloS one*, 17(4):e0265904, 2022.

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

[35] Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. Curriculum temperature for knowledge distillation, 2022.

[36] Yu-Po Lin, Chun-Yi Yeh, Pin-Yang Huang, Zong-Ye Wang, Hsiang-Hui Cheng, Yi-Ting Li, Chi-Fen Chuang, Po-Chiun Huang, Kea-Tiong Tang, Hsi-Pin Ma, Yen-Chung Chang, Shih-Rung Yeh, and Hsin Chen. A battery-less, implantable neuro-electronic interface for studying the mechanisms of deep brain stimulation in rat models. *IEEE Transactions on Biomedical Circuits and Systems*, 10(1):98–112, 2016.

[37] Yejia Liu, Shijin Duan, Xiaolin Xu, and Shaolei Ren. Metaldc: Meta learning of low-dimensional computing classifiers for fast on-device adaption, 2023.

[38] Yejia Liu, Wang Zhu, and Shaolei Ren. Navigating memory construction by global pseudo-task simulation for continual learning, 2022.

[39] Perrin Margaux, Emmanuel Maby, Sébastien Daligault, Olivier Bertrand, and Jérémie Mattout. Objective and subjective evaluation of online error correction during p300-based spelling. *Advances in Human-Computer Interaction*, 2012, 01 2012.

[40] Kishore Medhi, Nazrul Hoque, Sushanta Kabir Dutta, and Md Iftekhar Hussain. An efficient eeg signal classification technique for brain–computer interface using hybrid deep learning. *Biomedical Signal Processing and Control*, 78:104005, 2022.

[41] Anton Mitrokhin, Peter Sutor, Douglas Summers Stay, Cornelia Fermüller, and Yiannis Aloimonos. Symbolic representation and learning with hyperdimensional computing. *Frontiers in Robotics and AI*, 7, 06 2020.

[42] Masaki Nakanishi, Yu-Te Wang, Chun-Shu Wei, Kuan-Jung Chiang, and Tzyy-Ping Jung. Facilitating calibration in high-speed bci spellers via leveraging cross-device shared latent responses. *IEEE Transactions on Biomedical Engineering*, 67(4):1105–1113, 2019.

[43] Masaki Nakanishi, Yu-Te Wang, Chun-Shu Wei, Kuan-Jung Chiang, and Tzyy-Ping Jung. Facilitating calibration in high-speed bci spellers via leveraging cross-device shared latent responses. *IEEE Transactions on Biomedical Engineering*, 67(4):1105–1113, 2020.

[44] Peer Neubert, Stefan Schubert, and Peter Protzel. An introduction to hyperdimensional computing for robotics. *KI-Künstliche Intelligenz*, 33:319–330, 2019.

[45] Yang Ni, Nicholas Lesica, Fan-Gang Zeng, and Mohsen Imani. Neurally-inspired hyperdimensional classification for efficient and robust biosignal processing. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '22, New York, NY, USA, 2022. Association for Computing Machinery.

[46] G. Pfurtscheller and C. Neuper. Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, 89(7):1123–1134, 2001.

[47] Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe. Binary neural networks: A survey. *Pattern Recognition*, 105:107281, 2020.

[48] Abbas Rahimi, Pentti Kanerva, Luca Benini, and Jan M. Rabaey. Efficient biosignal processing using hyperdimensional computing: Network templates for combined learning and classification of exg signals. *Proceedings of the IEEE*, 107(1):123–143, 2019.

[49] Abbas Rahimi, Pentti Kanerva, and Jan M. Rabaey. A robust and energy-efficient classifier using brain-inspired hyperdimensional computing. In *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, ISLPED '16, page 64–69, New York, NY, USA, 2016. Association for Computing Machinery.

[50] Rahul Rai and Akshay V. Deshpande. Fragmentary shape recognition: A bci study. *Computer-Aided Design*, 71:51–64, 2016.

[51] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets, 2015.
[52] Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. Deep learning-based electroencephalography analysis: a systematic review. *Journal of neural engineering*, 16(5):051001, 2019.
[53] Theerat Saichoo, Poonpong Boonbrahm, and Yunyong Punsawad. Investigating user proficiency of motor imagery for eeg-based bci system to control simulated wheelchair. *Sensors*, 22(24):9788, 2022.
[54] G. Schalk, D.J. McFarland, T. Hinterberger, N. Birbaumer, and J.R. Wolpaw. Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on Biomedical Engineering*, 51(6):1034–1043, 2004.
[55] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey, 2021.
[56] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew Gordon Wilson. Does knowledge distillation really work?, 2021.
[57] Akara Supratak, Hao Dong, Chao Wu, and Yike Guo. Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(11):1998–2008, 2017.
[58] Francisco Velasco-Álvarez, Salvador Sancha-Ros, Esther García-Garaluz, Álvaro Fernández-Rodríguez, M. Teresa Medina-Juliá, and Ricardo Ron-Angevin. Uma-bci speller: An easily configurable p300 speller tool for end users. *Computer Methods and Programs in Biomedicine*, 172:127–138, 2019.
[59] Neeraj Wagh, Jionghao Wei, Samarth Rawal, Brent M. Berry, and Yogatheesan Varatharajah. Evaluating latent space robustness and uncertainty of eeg-ml models under realistic distribution shifts, 2022.
[60] Di Wu, Jingjie Li, Zhewen Pan, Younghyun Kim, and Joshua San Miguel. Ubrain: A unary brain computer interface. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*, ISCA '22, page 468–481, New York, NY, USA, 2022. Association for Computing Machinery.
[61] Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. When do curricula work?, 2021.
[62] Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification, 2020.
[63] Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and global knowledge distillation for detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4643–4652, June 2022.
[64] Tao Yu, Yichi Zhang, Zhiru Zhang, and Christopher De Sa. Understanding hyperdimensional computing for parallel single-pass learning, 2023.
[65] Fei Yuan, Linjun Shou, Jian Pei, Wutao Lin, Ming Gong, Yan Fu, and Daxin Jiang. Reinforced multi-teacher selection for knowledge distillation. In *AAAI Conference on Artificial Intelligence*, 2020.
[66] Dalin Zhang, Lina Yao, Xiang Zhang, Sen Wang, Weitong Chen, Robert Boots, and Boualem Benatallah. Cascade and parallel convolutional recurrent neural networks on eeg-based intention recognition for brain computer interface. In *Proceedings of the aaai conference on artificial intelligence*, volume 32, 2018.
[67] Yichi Zhang, Junhao Pan, Xinheng Liu, Hongzheng Chen, Deming Chen, and Zhiru Zhang. Fracbnn: Accurate and fpga-efficient binary neural networks with fractional activations, 2020.
[68] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11953–11962, June 2022.
[69] Qingqing Zhu, Xiuying Chen, Pengfei Wu, JunFei Liu, and Dongyan Zhao. Combining curriculum learning and knowledge distillation for dialogue generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1284–1295, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.
[70] Xuyang Zhu, Peiyang Li, Cunbo Li, Dezhong Yao, Rui Zhang, and Peng Xu. Separated channel convolutional neural network to realize the training free motor imagery bci systems. *Biomedical Signal Processing and Control*, 49:396–403, 2019.