Taming fNIRS-based BCI Input for Better Calibration and Broader Use

LIANG WANG* and ZHE HUANG*, Tufts University, USA
ZIYU ZHOU, Tufts University, USA
DEVON MCKEON, Tufts University, USA
GILES BLANEY, Tufts University, USA
MICHAEL C. HUGHES† and ROBERT J.K. JACOB†, Tufts University, USA

Brain-computer interfaces (BCI) are an emerging technology with many potential applications. Functional near-infrared spectroscopy (fNIRS) can provide a convenient and unobtrusive real time input for BCI. fNIRS is especially promising as a signal that could be used to automatically classify a user's current cognitive workload. However, the data needed to train such a classifier is currently not widely available, difficult to collect, and difficult to interpret due to noise and cross-subject variation. A further challenge is the need for significant user-specific calibration. To address these issues, we introduce a new dataset gathered from 15 subjects and a new multi-stage supervised machine learning pipeline. Our approach learns from both observed data and augmented data derived from multiple subjects in its early stages, and then fine-tunes predictions to an individual subject in its last stage. We show promising gains in accuracy in a standard "n-back" cognitive workload classification task compared to baselines that use only subject-specific data or only group-level data, even when our approach is given much less subject-specific data. Even though these experiments analyzed the data retrospectively, we carefully removed anything from our process that could not have been done in real time, because our process is targeted at future real-time operation. This paper contributes a new dataset, a new multi-stage training pipeline, results showing significant improvement compared to alternative pipelines, and discussion of the implications for user interface design. Our complete dataset and software are publicly available at https://tufts-hci-lab.github.io/code_and_datasets/. We hope these results make fNIRS-based interactive brain input easier for a wide range of future researchers and designers to explore.

CCS Concepts: • **Human-centered computing** → **Interactive systems and tools**.

Additional Key Words and Phrases: BCI, Brain-Computer Interface, neural networks, data augmentation, fNIRS, n-back task, machine learning, cognitive workload, near-infrared spectroscopy, implicit interfaces

ACM Reference Format:
Liang Wang, Zhe Huang, Ziyu Zhou, Devon McKeon, Giles Blaney, Michael C. Hughes, and Robert J.K. Jacob. 2021. Taming fNIRS-based BCI Input for Better Calibration and Broader Use. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21), October 10–14, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 26 pages. https://doi.org/10.1145/3472749.3474743

1 INTRODUCTION

Functional near-infrared spectroscopy (fNIRS) shows increasing promise to enable effective brain-computer interfaces (BCI) for a wide range of users. fNIRS is non-invasive, unobtrusive, and even easier to set up than conventional electroencephalograph (EEG) devices. However, a significant barrier to wider use is that fNIRS signals are difficult to

*Both authors contributed equally to this research.
†Both authors jointly supervised this work.

Manuscript submitted to ACM

analyze and have heretofore required extensive per-user calibration effort. Previous works have shown examples in which good fNIRS signals can be used as part of an interactive interface [2, 57, 58, 66, 67]. These efforts have usually tailored tasks to each specific experiment, pursued separate training datasets and classifiers for each subject, and used more traditional classifiers. While these works have obtained objectively measurable task performance improvements and thus offer compelling "proofs of concept," fNIRS remains difficult for a wider range of researchers to adopt. In this paper we attempt to advance the tools, infrastructure, and datasets that will facilitate wider use of fNIRS input.

A key challenge to developing subject-specific classifiers is that the amount of available training data from any single subject is usually limited. Small training sets are an impediment to good machine learning performance. The nature of fNIRS data especially limits the number of samples we can collect from a single subject. Unlike EEG, the signal that fNIRS measures has an inherently slow response rate (on the order of several seconds). This is not due to the equipment or technology, but rather the physiology of the body itself. A typical experimental session lasts for about an hour, and in general it is reasonable to think that typical users would not sit for a session that lasts for many hours. To overcome this limited data challenge, an intuitive idea is to augment the model by including data from other subjects performing the same task using the same measurement device. However, the human brain has considerable variability. Simply aggregating data from a larger pool of subjects has not worked well in our past experience.

In this work, we have developed a new multi-stage machine learning pipeline for training cognitive workload classifiers specific to a chosen "target" subject of interest that can leverage data from many subjects. Our new machine learning pipeline is composed of three phases designed to address the problems of limited data and cross-subject variability. The first phase uses *data augmentation* applied to data from a large pool of subjects (other than the target) to "pretrain" a deep learning classifier. The second phase trains further on the *observed* data from the same large pool of other subjects, producing the initial neural network parameters for the third phase. The third phase is then trained on data from the target subject. The result is a model that benefits from the information from other subjects, but is ultimately specialized to the target subject. Intuitively, this matches the notion that, while human brains are highly variable, they are also similar in some fundamental ways, and we ought to be able to benefit from that property.

We have also developed (and publish herewith) a new general purpose dataset of the fNIRS recordings from 15 users performing a standard psychological task. To our knowledge there are currently few large datasets available for this purpose. We have investigated several [8, 30] and particularly the 26-subject dataset from TU-Berlin [54], which is one of the best available datasets, but found we needed additional data collected with different sensors and a more controlled environment to properly assess the generalization of our pipeline. We show that our proposed pipeline delivers strong performance on both our 15 subject dataset and the external TU-Berlin dataset of 26 subjects [54].

Contributions. The contributions of our work can be summarized as follows:

1. We *release a new multi-subject fNIRS dataset* at https://tufts-hci-lab.github.io/code_and_datasets/ along with demographic and contextual information, cognitive task performance records, subjective workload, experiment log reports, and post-experiment interviews;

2. We propose a *3-phase machine learning-based pipeline* to improve subject-specific classification of n-back cognitive workload from short windows of fNIRS time-series data, reducing the required amount of subject-specific training data by leveraging data from other subjects;

3. We *open source our software* at https://tufts-hci-lab.github.io/code_and_datasets/ to provide an easily-accessible system/tool for other researchers, including code for data collection and machine learning.

Ultimately, our dataset, code, and methodology can enable future researchers to flexibly explore new methods and use cases for BCI workload classification and improve generalization to new subjects. We intend to expand our dataset to more subjects in future work, in hopes that additional data from more subjects will improve our technique even further and advance the state-of-the-art for fNIRS-based cognitive workload classification.

2 RELATED WORK

Generally, cognitive workload ("working memory workload") refers to the quantity of working memory resources used when performing cognitive tasks. Tasks with more cognitive demand (*i.e.* working memory tasks of higher difficulty) induce higher levels of cognitive workload [32]. The n-back task is a standard method for inducing working memory cognitive workload. The task presents a subject with a series of stimuli and requires them to compare the current stimulus to the stimulus shown n steps previously [43]. Extensive previous work suggests that as n increases, heightened workload can be quantified through lower rates of accuracy among users [47]. The range of n values can simulate differences in workload similarly to real tasks of varying difficulty [19], or can be used as a means to induce different amounts of workload to be measured and compared [50]. A meta-analysis of the n-back task conducted by von Janczewski and Nikolai, et al. [64] confirmed the n-back task's effect on cognitive workload. We used the n-back task for our study, because it is a well-established experimental task and can thus provide a grounded evaluation of the effect of our machine learning approach, rather then creating an untested new task.

Self-reports of cognitive workload. Self-reports are widely accepted collaborating evidence of cognitive workload. Many researchers have developed methods of measuring perceived workload. Hart and Sandra G created the NASA-TLX [25] which has six dimensions: 1. mental demand; 2. physical demand; 3. performance; 4.effort; 5. frustration. A. Tattersall and P. S. Foord [61] developed a simpler scale to collect subjects' subjective cognitive workload. However, John Sweller and Ayres, et al. [60] pointed out that self-reports should not be treated as direct measurements of cognitive workload.

Physiological measures of cognitive workload. There are several kinds of physiological data that can be used to reflect cognitive workload, such as EEG[9, 44, 55], GSR[53], and eye tracking[15, 42]. Physiological measures are advantageous in that they can monitor subjects' concurrent cognitive workload. They are used to build adaptive real-time Brain-Computer Interfaces (BCI) [7, 22].

fNIRS as a measurement of cognitive workload. Work by Afergan et al. [2] shows that we can use fNIRS signals to detect task difficulty in real-time and construct an interface that improves user performance through dynamic difficulty adjustment. A system developed by Yuksel et al. [67], can dynamically increase the levels of difficulty in a musical learning task based on subjects' cognitive workload.

Open-access fNIRS datasets for cognitive workload tasks. We have investigated all 30 papers that match a keyword search for fNIRS and were published within the last five years in the ACM digital library. As noted above, the best open-access cognitive workload fNIRS dataset we found is by Shin et al. [54]. Our contribution of a new and more rigorously controlled dataset (as we described in Section 3) thus fills a critical need for the research community.

Previous machine learning classifiers of cognitive workload from n-back fNIRS data. Multiple previous studies have developed machine learning classifiers to identify cognitive states given fNIRS data. Some prior studies focus on average activation patterns for different types of workload [5, 20, 29, 49]. Herff et al. [28] developed subject-specific models using hand-engineered features from sliding windows and reported 78% accuracy on 1-back vs. 3-back tasks across

10 subjects. Aghajani et al. [4] also developed subject-specific models using hand-engineered features from sliding windows, reporting 74.8% accuracy on a 0-back vs. 2-back task across 15 subjects. However, neither of these studies looked at deep learning methodologies to learn feature representations or considered cross-subject transfer of models.

Previous deep learning classifiers of cognitive workload from fNIRS data. Deep learning is a subfield of machine learning that uses artificial neural networks to develop flexible learned representations from complex and high-dimensional input signals, often vaguely inspired by the structure of human brain. Deep learning has been gaining research attention in the past decade due to its successes in many areas such as image classification, speech recognition and machine translation [39].

In recent years, several research efforts on fNIRS data have begun to investigate deep learning methods. Early work by Hennrich et al. [27] applied feed-forward neural networks to classify which cognitive tasks were performed, yielding prediction quality comparable to conventional (non-neural) classifiers. Saadati et al. [51] explored the use of *convolutional neural networks* (CNNs) for mental workload classification from fNIRS data. They found significant improvement compared to conventional methods and regular neural networks without convolutions. Benerradi et al. [8] found that convolutional neural networks required training datasets so large that they could not build subject-specific "personalized" models, only subject-independent "generalized" models. In the generalized setting, their CNNs classified high vs. low mental workload with similar accuracy (72.77% vs. 71.27%) as more conventional support vector machine (SVM) methods. Recently, Kwon et al. [35] used CNNs to build subject-independent fNIRS models. Their CNN architecture used an evolving normalization-activation layer (EvoNorm) [40] in place of the more common Batch Normalization layer (BatchNorm) [31]. They report average classification accuracy of over 70% on distinguishing mental arithmetic versus idle state, outperforming alternative pipelines including another deep-learning model, *EEGNet* [38].

Previous machine learning methods for cross-subject BCI. In general, BCI systems require tedious calibration procedures that adapt to subject-specific data, which has been a major obstacle to wider applications of BCI. Previous studies have explored ways to learn from multiple subjects in order to improve subsequent generalization to a target subject. Kwon et al. [36] applied ML classifiers to EEG data for motor imagery (MI) BCI classification. They found that a subject-independent model (trained on data from many other subjects) can outperform models trained only on a separate session of target subject training data. This motivates our pursuit of learning feature representations from many subjects.

Recently, a study by Lyu et al. [41] considers the problem of workload classification from fNIRS data that specifically adapts across subjects. They train models using data from one subject and generalize to a completely different subject, using methods from optimal transport. They achieve 55% accuracy in a balanced four-class n-back task compared to 44% for a baseline convolutional neural net. Their task of interest allows no subject-specific training ("calibration"). In contrast, our study can learn representations from many subjects while allowing later subject-specific calibration, which we show can improve performance.

Kostas and Rudzicz [34] considered EEG from multiple subjects, using MixUp for data augmentation. They focused on a deep neural network architecture "TIDNet" to classify EEG data. MixUp is important to their work as a regularization method to avoid the degenerate case where their large network might simply "memorize" the training data. Their proposed network, TIDNet, performs well with the help of Euclidean Alignment (a technique to try to standardize the measured signals across multiple subjects) and MixUp.

Han and Jeong [24] approached BCI from a *domain generalization* (DG) perspective, considering EEG data from 15 subjects, where each subject contributed two sessions on separate days. Their goal was to generalize to a new

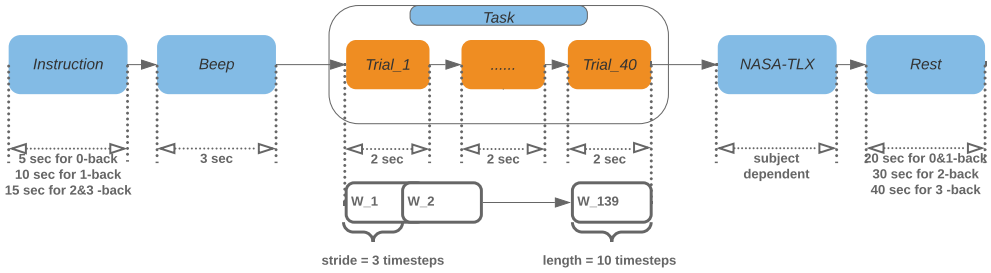


Fig. 1. For each task, subjects underwent pre-task, task, and post-task stages. The duration of instruction and rest period varied among different n-back tasks. Our open source dataset contains data from the whole process. In this study, only the task periods (trials 1-40) were analyzed and used to generate the sliding window data.

session. They conclude that using subject-specific data from Day 1's session can deteriorate Day 2 performance due to inter-session variability. We will later show that within-session fine-tuning improves performance. Han and Jeong [24] also experimented with expanding their limited training data via synthetic samples from MixUp. However, they found MixUp does not improve classifier performance across sessions. In contrast, our later results show improvements from synthetic data via a carefully-designed MixUp procedure.

3 EXPERIMENTAL METHODOLOGY

3.1 Experiment design

3.1.1 IRB and COVID-19 protocol. Our entire data collection was performed in early 2021 during the COVID-19 pandemic. All procedures were approved by our institution's IRB and COVID safety review committee. Subjects were compensated $20 USD. The experimenters used personal protective equipment, disinfected the fNIRS probe for each subject, and waited 72 hours between subjects to allow proper ventilation and sanitation, as approved by the review committees.

3.1.2 Pre-experiment. We collected demographic information from the subject. The system then played a short introductory video, showing an example of a user completing n-back tasks, with voice-over and caption explanations. To minimize interruptions, the video instructed the subject to remain seated, not to talk, and to refrain from adjusting the fNIRS sensors for the duration of the experiment. After the video, the operator placed the sensors on the subject.

3.1.3 Experimental tasks. We utilized the n-back task because, as previously discussed, it is a well-established method for inducing working memory mental workload. Our subjects completed 16 n-back tasks, where the value of n within each task was either 0, 1, 2, or 3. The order of tasks was predetermined and the same for every subject. Organized into 4 sequences of 4 tasks each, the order (0→1→2→3→1→2→3→0→2→3→0→1→3→0→1→2) resembled a Latin square (see Appendix E.1); each value of n occurred once in each of the 4 positions within the sequence. Each task (depicted in Figure 1) was administered as follows:

Pre-task. Before each task, the system displayed a graphic depicting how to identify targets for the current n. The start of the task was indicated with several beeps.

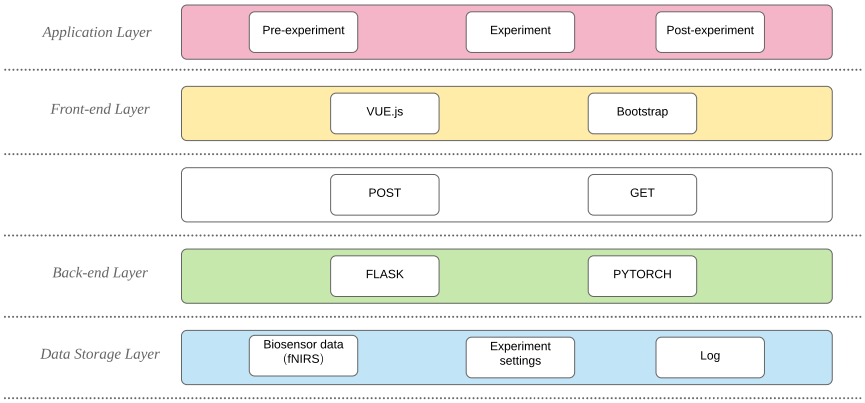


Fig. 2. Framework implementation

Task. Each task consisted of 40 trials for the same chosen value of n. During each trial, a digit between 0 and 9 was shown on the screen for 1.5 seconds and then hidden for 0.5 seconds before the next trial began. The subject needed to press the left arrow key if the digit on the screen was a target (i.e. the same as the digit flashed n trials previously) and the right arrow key otherwise.

Post-task. After each n-back task, the interface prompted the subject to report their subjective workload according to the NASA Task Load Index (TLX) [25]. Then, the interface instructed the subject to close their eyes and rest. The duration of the rest depended on n, with the assumption that harder tasks need longer rests. For 0 and 1-back tasks, the subject was given 20 seconds of rest. For 2-back tasks, the subject was given 30 seconds, and for 3-back, 40 seconds.

3.1.4 Post-experiment. After the subject completed the entire experiment, we conducted a short interview about the subject's experience.

3.2 Framework implementation

To collect high quality fNIRS data, we designed and implemented a fully automated, modularized, multi-modal framework that is capable of analyzing offline and online data.

3.2.1 Automation. In previous studies, we observed that many subjects did not fully understand the experimental tasks and might to ask for help from the operator. This might cause interruption or pollution of fNIRS data. Our system accounts for this by providing every subject with the same information via a detailed introduction and tutorial video. Additionally, the interface provides the subject with a graphic before every n-back task as a reminder of the instructions for the current value of n.

Furthermore, in previous studies, we observed that having the operator on duty manually performing operations during the experiment can result in errors. With our automated system, we eliminate the chance of these errors since the operator is not responsible for any operations during the experiment.

3.2.2 Modularization. We used extensive modularization throughout all parts of our framework to ensure that we can easily add devices and update parameters.

We split the whole experiment process into a series of components such as the pre-experiment (survey, general introduction, device calibration), experiment (sessions, tasks, in-task introduction, trials, post-task surveys, and rest periods), and post-experiment survey. Each component has a corresponding modularized VUE template, making it flexible to re-organize the front-end software and adapt to new experiment procedures. Other researchers could make use of our system with their own experiment design. The front end is also parameterized, making it easy to update the contents inside the components (audios, videos and graphics).

The back-end consists of multiple individual modules, such as data receivers to receive and store data, data pre-processing modules to pre-process the data according to specific device features, the machine learning module to analyze the BCI data, and the prediction generation server to provide real-time output.

3.2.3 Multi-modal. The framework has standardized receiver protocols and supports multi-modal inputs (fNIRS, GSR device, portable EEG device, etc.). More devices can easily be added into the experiment to gather additional biophysiological data, enabling multi-modal machine learning methods.

3.2.4 Offline and online. The framework supports both offline and online data analysis. We can use the system to collect data, run offline data analysis, find the best model, and then save it as a pre-trained model to reduce training time cost on new subjects. Then, we can deploy the model in real-time and generate predictions based on the subject's concurrent biophysiological state.

3.3 Data collection

Over 3 months in early 2021, 27 healthy individuals (age range 20 ± 2.3, 18 to 28 years old) have participated in the study. We deemed 15 of these subjects' data eligible for this work based on their behavior during the experiment session, as explained below.

Demographic and contextual information. Through the testing interface, we collected demographic data (gender, sex, age, ethnicity, handedness, vision, native language) from the subject. We also collected contextual data (previous head injuries, sleep habits, caffeine intake, drug use, prior experience with biological sensors or n-back tasks).

fNIRS measurements. To measure changes in the physiologically relevant chromophore concentrations, oxyhemoglobin ($[HbO]$) and deoxyhemoglobin ($[HbR]$), frequency-domain near-infrared spectroscopy (FD-NIRS) was used. Given that the goal of these measurements was to measure functional activation, the measurement will be referred to as fNIRS. FD-fNIRS was implemented at a modulation frequency of 110 MHz and wavelengths of 690 nm and 830 nm (ISS Imagent, Champaign, IL USA). The output of the FD-fNIRS measurement was the alternating-current (AC) intensity (I) and phase (ϕ) for each source-detector pair.

As shown in Figure 4, two optical probes were placed on each subject's forehead, one for the left hemisphere and the other for the right hemisphere. The probes were secured to the subject's head using a hook and loop headband which passed through the center of the probe. Light was delivered to the probe via 400 μm diameter multi-mode fibers and collected by 5 mm diameter fiber bundles. These fibers were held in-place by a flexible plastic mesh and were encapsulated in black silicone [12].

Each optical probe had optode geometry implemented via the dual-slope (DS) method [13]. A schematic of one of these DS probes is shown in Figure 3. Each probe consisted of two source positions (each with two wavelengths) and

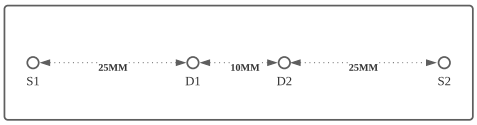


Fig. 3. Structure of a single dual-slope (DS) functional near-infrared spectroscopy (fNIRS) probe. S1 and S2 are two light sources, each emitting light of two wavelengths (830 nm and 690 nm). D1 and D2 are two detectors. Each probe consists of two source-detector pairs at 25 mm and two at 35 mm.

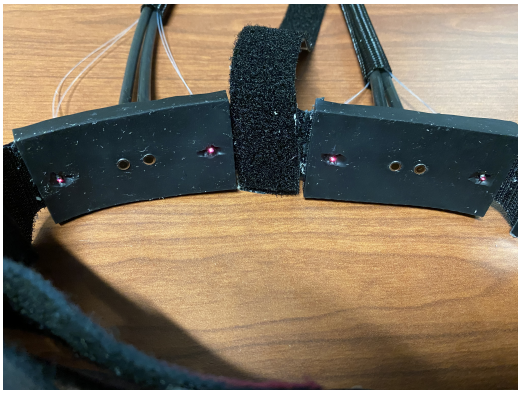


Fig. 4. fNIRS headband

two detectors. For each DS probe, data from all combinations of sources and detectors were collected, resulting in a total of four single-distance (SD) measurements of I and ϕ (source-detector distances (ρ): two of 25 mm and two of 35 mm).

Cognitive task performance. We measured the subject's performance at the n-back task based on the accuracy of the subject's response for each digit. During our pilot study, subjects were asked to press the space bar for targets. However, we found that when subjects were unsure whether a digit was a target, particularly as the value of n increased, they tended to skip it. Their non-response was recorded identically to an intentional response for a non-target number. In our updated experiment design, we incorporated both arrow keys in order to differentiate between intentional responses for non-targets and non-responses; pressing the correct arrow key for a digit was considered a correct response, while pressing the wrong arrow key or pressing no arrow key were considered incorrect responses.

Subjective workload. We used the NASA-TLX as a measure of subjective workload. After each n-back task, subjects rated each dimension of workload in the NASA-TLX on a 21 point scale.

Experiment log. The operator on duty logged any issues that happened during the experiment. This is an essential step which has been neglected by previous public biosensor and fNIRS datasets. fNIRS sensors are very sensitive to the environment and can be polluted by many factors. We require the operator to log issues such as interference from a subject's hair and light leaking (often occurring when the shape of a subject's forehead does not match the curve of the headband). We also ask the operator to report any DC intensity detector over-voltage warnings. The higher DC intensity we set, the better data we are able to collect; however, we want to avoid over-voltage and saturation, which may cause the fNIRS system to shut down automatically.

Post-experiment interview. We asked 11 open-ended questions in a recorded interview. The questions targeted the subject's physical comfort, emotions, and experience with the experimental tasks, testing interface, and hardware.

Data exclusions. Of the 27 total subjects, 15 met all eligibility criteria for this work. The remaining 12 subjects' data were excluded. Three subjects' data were excluded because their performance at digit recognition during the n-back tasks was anomalous (two were too low, one was too high), indicating that their cognitive workload was different than intended. Five subjects' data were excluded because they received a different range of fNIRS DC intensity settings than other subjects. Four subjects' data were excluded because of abnormal oxygen dynamics which were always high

regardless of task difficulty levels. This was generally due to the subjects' hair blocking the light sources or causing light leakage. Data for all included and excluded subjects are in the publicly released dataset.

It is important to note that the exclusion criteria did not depend on any machine learning results, but only on factors that were observed before analyzing the data. We have refined our experimental procedure for subsequent data collection to reduce the light leakage problem. We will continue to screen for anomalously very good or very bad performance on the n-back task.

Open source dataset release. All 27 subjects' data are available at https://tufts-hci-lab.github.io/code_and_datasets. The 15 subjects meeting all eligibility criteria are clearly marked. Released data include demographics, fNIRS measurements, n-back task performance, subjective workload, experimental logs, and post-experiment interviews.

3.4 Data pre-processing

Offline (non-realtime) pre-processing techniques, such as sparse optimization [41, 48] or wavelet methods [1, 17, 33, 63], are often applied to remove artifacts from fNIRS data. However, we did not use these because our focus is exclusively on real-time methods. We applied the Dual-Slope (DS) method to pre-process the raw fNIRS data. DS analyses have been presented previously [13] and are described in detail in Appendix B. This method is advantageous (compared to Single-Distance (SD) methods typically used in fNIRS) because of its insensitivity to optical coupling and drifts, reduction of motion artifacts, and reduced sensitivity to superficial tissue (*i.e.* preferential sensitivity to the brain). This can be applied for any probe that satisfies the DS geometrical requirements [12].

After this pre-processing, each of the 16 n-back tasks for each subject consisted of a multivariate time-series of 8 measurements, each recorded at a sampling rate of 5.2084 Hz. The 8 measurements represent all possible combinations of sensor locations (left, right), chromophore concentrations (*HbO*, *HbR*), and measured signal properties (phase, intensity).

We chose not to normalize the pre-processed data. Even though normalization can often provide better offline classification results, many methods for time series normalization cannot easily be applied in online (real-time) analysis, which is our ultimate goal.

4 MACHINE LEARNING METHODOLOGY

4.1 Classification task and data preparation

Next, we sought to set up a supervised learning task. Our goal is to develop a prediction system that takes a $\sim$ 2-second duration multivariate fNIRS signal as input, and produces a probabilistic prediction of the user's current cognitive intensity category within that brief segment of time. As a proxy for intensity, we used the current value of n in the assigned n-back task. We chose this 2-second duration because we believe near-real-time workload classification every 2 seconds would produce useful and responsive interfaces. Furthermore, we suggest that any finer resolution would be difficult due to the inherent lag time between external stimuli and the oxygen dynamics in the brain measured by fNIRS.

For each subject (indexed by i), we assembled a labeled dataset representing W short-duration "windows". In each window (indexed by w), we observed a time-series of T fNIRS measurements $x^{(i)}_{w,1:T}$ and a corresponding workload intensity category label $y^{(i)}_w \in \{0, 1, 2, 3\}$. The goal of our classifier is to predict the value of workload label $y^{(i)}_w$ given the window's fNIRS measurements $x^{(i)}_{w,1:T}$. Both the fNIRS measurements x and the cognitive workload labels y are easily collected in our experiment because we know which n-back task the user was performing at any given time. In this study, we considered two possible classification tasks: a binary classification task where we want to distinguish

between 0-back versus 2-back (ignoring other values of n), and a four-class classification task that distinguishes between 0-back, 1-back, 2-back, and 3-back.

We define each window to be exactly $T = 10$ timesteps in duration (at 5.2 Hz, each window is a ~2-sec. long). Thus, each window's measurements $x^{(i)}_{w,1:T}$ represent exactly T =10 timesteps of the multivariate fNIRS signal. At each timestep (indexed by t) within the window, we observe a vector $x^{(i)}_{w,t} \in \mathbb{R}^8$. To generate the per-window feature vectors, we take in a subject's raw time-series data and segment it with a sliding window of size 10 timesteps and a stride of 3 timesteps, as shown in Figure 1. Each subject contributed exactly $W = 2224$ windows, evenly sampled between all possible values of n.

Splitting data. For each subject i, we divide all available windows into a labeled training set and labeled test set that have no temporal overlap. We use a 1:1 train-test split: the first half of subject i's data in chronological order becomes subject i's training set, and the remaining data becomes subject i's test set. There are W_{tr} windows in the training set $D^{(i)}_{tr} = \{\mathbf{x}^{(i)}_{1:W_{tr}}, \mathbf{y}^{(i)}_{1:W_{tr}}\}$, and W_{te} windows in that subject's test set $D^{(i)}_{te} = \{\mathbf{x}^{(i)}_{(W_{tr}+1):(W_{tr}+W_{te})}, \mathbf{y}^{(i)}_{(W_{tr}+1):(W_{tr}+W_{te})}\}$.

Some methods can learn from other subjects too. For these methods, when training on subject i, we also make available a large labeled dataset of data from the 14 other subjects (indexed by $j \neq i$). For each other subject, we include all W available windows in dataset $D^{(j)} = \{\mathbf{x}^{(j)}_{1:W}, \mathbf{y}^{(j)}_{1:W}\}$. To be clear, the test set is not touched until final evaluation. In our later experiments reported in Tables 1 and 2, note that we compared using 100% of a target subject's available training data to using only a fraction of that subject's training data (50 %, 0%), in order to assess the model's ability to reduce individual calibration time.

Train/test splits for binary classification. In binary classification, our goal is to determine for a specific subject whether a given window of fNIRS measurements was obtained during a 0-back or 2-back task, which represent low and high workloads respectively. Both the train and test set have the same number of windows ($W_{tr} = 556$, $W_{te} = 556$), evenly balanced between $n = 0$ and $n = 2$ labels. Only windows collected during 0-back and 2-back tasks are included.

Train/test splits for four-class classification. In four-class classification, we classify each window as either 0-back, 1-back, 2-back or 3-back. Again, train and test sets are balanced with the same size ($W_{tr} = 1112$, $W_{te} = 1112$).

4.2 3-phase training approach

Motivated by the widespread success of deep learning in many prediction tasks involving time series, we wish to pursue a deep learning approach for producing a predicted label probability vector given an observed window of fNIRS data $x^{(i)}_{w,1:T}$. For this study, we use a convolutional neural net (CNN) [21], with 1 convolutional layer followed by 2 fully connected layers (fully described in Sec. C.1). This approach builds on a growing literature which suggests that CNNs are successful for BCI applications [23, 30, 37, 52, 65].

However, while common wisdom suggests that deep neural networks (and especially CNNs) can deliver superior performance over manually-engineered feature representations of fNIRS data, gains from deep learning often only appear when the number of labeled examples available for training is quite large. After carefully processing our dataset to perform cognitive workload classification, only about 5-10 minutes of labeled training data are available for each subject. We thus face a challenge of limited labeled data. While several previous works have pursued CNNs for fNIRS, our contribution is a new 3-phase approach to train CNNs that can overcome limited available labeled data to deliver improved heldout accuracy.

Our proposed 3-phase approach (illustrated in Figure 5) involves two key ideas. First, we can pretrain a CNN classifier on a large labeled set gathered from multiple subjects, and then fine-tune this classifier on a labeled set specific to

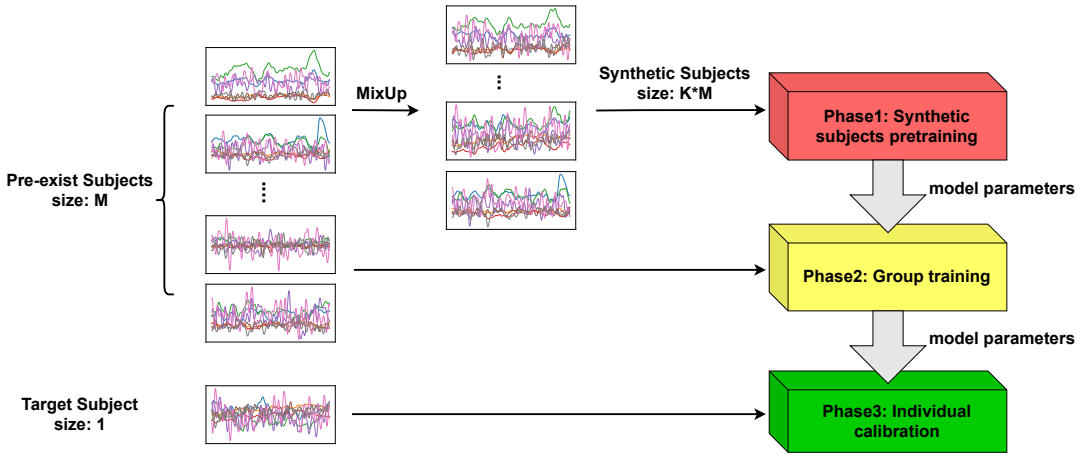


Fig. 5. Illustration of our proposed 3-phase approach to training CNNs to classify cognitive workload from short windows of fNIRS time-series data. After each phase, the resulting CNN parameters are passed on to initialize the next phase.

the target subject. Second, we can boost performance even more by pretraining on an even larger artificial labeled set assembled via MixUp data augmentation.

Our goal is to train a personalized classifier for subject i. We have access to two training sets: the target subject's training dataset $D_{tr}^{(i)}$, as well as the aggregation of full datasets from the 14 other subjects $\{D^{(j)}\}_{j\neq i}$. We make use of each dataset in different phases. All phases use the same CNN network architecture and are related to each other by passing values of the CNN parameters. The first phase is initialized randomly. Then, each subsequent phase is initialized to the learned parameters produced by the previous phase. In this way, each phase provides a "warm start" to its successor.

Each phase can be summarized as follows:

(1) **Phase 1: Train on augmented data from non-target subjects.** In the first phase, we use the 14 other subjects $\{D^{(j)}\}_{j\neq i}$ as source data. We apply a data augmentation technique known as MixUp [68] to these source examples, to create a larger synthetic dataset with the effective size of up to 112 subjects from our collection. We then train our CNN on this augmented dataset. The goal of this first phase is to learn coarsely useful representations of human brain signals. This augmentation is fully-described in Sec. 4.3.

(2) **Phase 2: Train on observed data from non-target subjects.** In the second phase, we train our CNN on the observed data from the 14 other subjects $\{D^{(j)}\}_{j\neq i}$. The goal for this second phase is to leverage other subjects' data to further improve learned representations. Like phase 1, we expect improved representations from this stage because human brains share some commonalities.

(3) **Phase 3: Personalize on training data from target subject.** In the final phase, we train our CNN on the target subject's training set $D_{tr}^{(i)}$. This stage is often called *fine-tuning* because while previous stages learn from many subjects, this phase specializes exclusively to the specific subject i of interest. Fine-tuning allows the model to adapt to specific characteristics of subject i.

We deliberately separate phase1 and phase2, so that each phase has a clear and disentagled purpose. Our later results will show that our 3-phase approach leads to a significant performance improvement compared to common methods

that would use *only* the data specific to the target subject. We emphasize that in our experiments, to make the best use of available data, we specialize all 3 phases to each subject (indexed by i). That is, the source pool of "other" subjects used in Phase 1 and Phase 2 is distinct for each value of i. In a deployment scenario, we could use all available previous subjects as a common pool for the first two phases. This would require only training the last phase for a new subject.

4.3 Data augmentation for Phase 1

MixUp [68] is a data augmentation technique that constructs new *augmented* labeled examples x', y' from a pair of existing labeled examples x_a, y_a and x_b, y_b. The new samples are created by *linearly interpolating* between both the feature vectors as well as the one-hot label indicator vectors. The method has been successfully used in many supervised learning and semi-supervised learning tasks [6, 10, 14, 16, 18, 26, 62, 69]. MixUp and many other data augmentation techniques have been mostly applied in the image classification domain, with only a few emerging attempts in the BCI domain [24, 34]. In this study, we show that MixUp can be used on fNIRS data to create artificial samples that noticeably boost model performance.

To produce a synthetic fNIRS dataset in phase 1 when our target subject has index i, we repeatedly sample two different subject indices a and b uniformly at random from the available subjects represented in our phase 1 training data $\{D^{(j)}\}_{j \neq i}$. We then visit each window w available from subject a in temporal order, and mix that window with the *corresponding* window from subject b, to produce a new synthetic window with features $x'_{1:T}$ and label indicator y' using sampled interpolation weight $\lambda \in [0, 1]$:

$$x'_{1:T} = \lambda x^{(a)}_{w,1:T} + (1-\lambda) x^{(b)}_{w,1:T}, \; y' = \lambda y^{(a)}_{w} + (1-\lambda) y^{(b)}_{w}, \qquad \lambda = \max(\lambda', 1-\lambda'), \quad \lambda' \sim \text{Beta}(\alpha, \alpha). \tag{1}$$

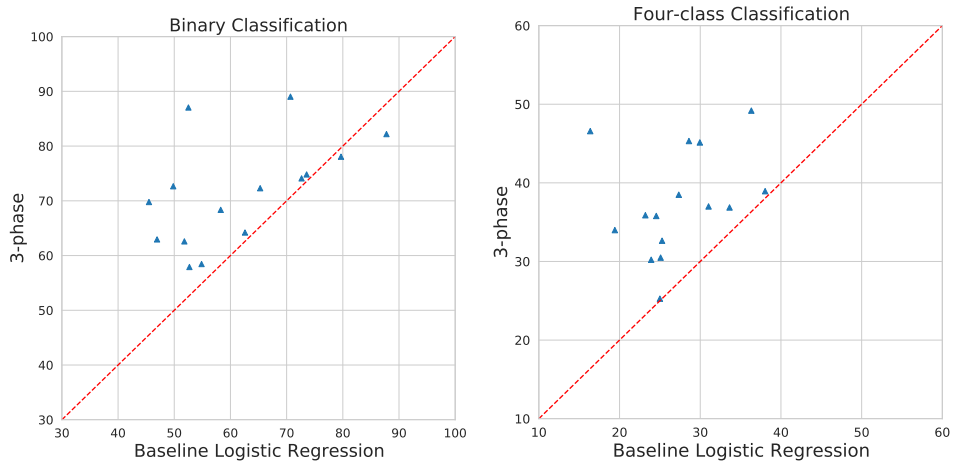
Hyperparameter $\alpha > 0$ controls how distinct the new sample will be from its sources, with larger values producing x' values more likely to be further from the sources. Because we chose the same window in temporal order for both subjects, the source labels will be the same ($y^{(a)} = y^{(b)}$) and thus the synthetic label y' will be the same.

We emphasize that we are careful to use the same window from both chosen subjects, while a standard MixUp implementation might sample different windows for subject a and subject b. We suspect that using the same window will produce more "realistic" samples, because it keeps the labels the same and will preserve global trends in how subject's response change over the course of data collection.

4.4 Baselines and experimental protocol

We consider several possible baseline methods to compare to our proposed 3-phase pipeline:

- **Subject-Specific CNN.** This baseline CNN uses the same architecture as our 3-phase approach, but is trained on only the target subject's data. Effectively, this is only "Phase 3" in our approach, omitting the first two phases.
- **Last-2-Phases CNN.** This baseline omits the first phase (MixUp augmentation). It starts by pre-training on the data from all other subjects (as in Phase 2), and then fine-tunes on the training set for subject i (as in Phase 3). Comparisons to this baseline let us directly assess how much MixUp helps.
- **Subject-Specific Logistic Regression (LR) and Random Forest (RF) Classifiers.** These two baselines use simpler non-neural classifiers which consume a hand-engineered feature vector meant to summarize the observed multivariate time-series window of fNIRS measurements $x^{(i)}_{w,1:T}$, as in [4, 28]. Comparisons to these two baselines let us assess how much using a CNN deep learning approach helps.



(a) Two-class workload classification task (b) Four-class workload classification task

Fig. 6. Test accuracy comparison across 15 possible target subjects for binary classification (*left*, 0-back vs. 2-back) and four-class classification (*right*, 0- vs. 1- vs. 2- vs. 3-back). Each dot reports one subject's accuracy using our 3-phase method (y-axis) and the logistic regression (LR) baseline (x-axis).

All considered prediction methods require careful hyperparameter search to avoid overfitting and achieve good generalization performance. For each method, we perform a search over several candidate hyperparameter values (intended to cover a range of settings including both under-fitting and over-fitting). For each method and each target subject i, we select the hyperparameter setting that achieves the best validation accuracy, averaged across 5-folds of cross-validation. When dividing the target subject's training set $D_{tr}^{(i)}$ into 5 folds, we make sure the folds are *chronologically distinct* to avoid overlapping windows that would cause information leakage between the training and heldout data in one train/test split. We ensure that each fold's data comes exclusively from a different, non-overlapping time interval in the original data collection sequence. For details on the hyperparameter search for each method, see Appendix C.

After selecting a preferred hyperparameter configuration, we have 5 separate CNN models corresponding to the chosen hyperparameter values for subject i (one for each of the cross-validation folds). To make predictions on new test data, we *average* the probabilistic predictions produced by these 5 models.

5 RESULTS

We now present and interpret our experimental results, comparing the proposed 3-phase paradigm for training subject-specific classifiers to other approaches. Here, we focus on performance averaged across all 15 subjects in our leave-one-subject-half-out design (each target subject's first half of data is available for training, the rest for test). For subject-specific performance information, see Appendix F.

Binary classification results. Our first experiments examine the binary cognitive workload classification task: 0-back vs. 2-back (low versus high work load, respectively). Table 1 compares test set accuracy (averaged across subjects) of several possible ML pipelines. We can see that our 3-phase approach improves binary classification performance over all subject-specific baselines (3-phase achieves 71.6% accuracy vs. 62.8% for CNN, 57.7% for random forests, and 61.6%

Labeled Training Data Used				Acc. on Test Set	
From Other Subjects	From Target Subj.	Phases	Model	(avg. over 15 subj.)	
0	5.0 min. (100% train set)	3 (subj.-specific only)	LR	61.64	(55.53, 68.05)
0	5.0 min. (100% train set)	3 (subj.-specific only)	RF	57.70	(52.84, 63.01)
0	5.0 min. (100% train set)	3 (subj.-specific only)	CNN	62.75	(58.60, 67.24)
14 subj., 5 min. each	5.0 min. (100% train set)	2 + 3	CNN	67.00	(62.61, 71.73)
14 subj., 5 min. each	5.0 min. (100% train set)	1 + 2 + 3	CNN	**71.63**	(66.97, 76.46)
14 subj., 5 min. each	2.5 min. (50% train set)	1 + 2 + 3	CNN	65.04	(60.52, 69.69)
14 subj., 5 min. each	0 min. (0% train set)	1 + 2	CNN	55.61	(50.98, 60.54)

Table 1. Results from binary classification of cognitive workload (0-back vs 2-back) given 2-second segments of fNIRS measurements. We report the average accuracy across 15 subjects. Values in parentheses indicate the 2.5th and 97.5th percentiles from 5000 bootstrap samples. A system that guesses at random would have 50% accuracy. LR = logistic regression, RF = random forest, CNN = convolutional neural network.

Labeled Training Data Used				Acc. on Test Set	
From Other Subjects	From Target Subj.	Phases	Model	(avg. over 15 subj.)	
0	10 min. (100% train set)	3 (subj.-specific only)	LR	27.18	(24.23, 30.10)
0	10 min. (100% train set)	3 (subj.-specific only)	RF	26.65	(23.29, 30.15)
0	10 min. (100% train set)	3 (subj.-specific only)	CNN	27.90	(26.26, 29.63)
14 subj., 10 min. each	10 min. (100% train set)	2 + 3	CNN	32.96	(30.48, 35.48)
14 subj., 10 min. each	10 min. (100% train set)	1 + 2 + 3	CNN	**37.46**	(34.26, 40.80)
14 subj., 10 min. each	5 min. (50% train set)	1 + 2 + 3	CNN	35.40	(32.70, 38.03)
14 subj., 10 min. each	0 min. (0% train set)	1 + 2	CNN	26.52	(24.84, 28.43)

Table 2. Results from four-class classification of cognitive workload (0-back vs 1-back vs 2-back vs 3-back) given 2-second windows of fNIRS measurements. We report the average accuracy across 15 subjects, plus the 2.5th and 97.5th percentiles of 5000 bootstrap samples of test set performance. A system that guesses at random would have 25% accuracy.

for logistic regression). Fig 6(a) visualizes the accuracy gains for each of the 15 subjects from our 3-phase pipeline over the logistic regression baseline.

Four-class classification results. Table 2 compares test set accuracy (averaged across subjects) on the more challenging task of four-class workload classification. Again, we can see that our 3-phase approach significantly improves four-class classification performance over subject specific baselines (our 3-phase achieves 37.46% compared to the *near-chance performance* of 27.90% for subject-specific CNN, 26.65% for RF, and 27.18% for LR). Fig 6(b) visualizes accuracy gains for each of the 15 subjects from our 3-phase pipeline over the logistic regression (LR) baseline.

External dataset binary classification results. As an additional validation of our method, we apply our classification pipeline to the open-access TU-Berlin fNIRS dataset containing 26 subjects [54]. We pursue binary workload classification (0-back vs. 2-back). Our 3-phase approach improves baseline subject-specific test accuracy from 60.80 % to 70.69%. Appendix D.1 details our experimental protocol and results.

The paragraphs below summarize the major conclusions we draw from these experiments.

Pretraining on other subjects – as in Phase 2 – improves performance over only using subject-specific data. In both Table 1 (binary classification) and Table 2 (four-class classification), we see consistent gains from our Phase 2,

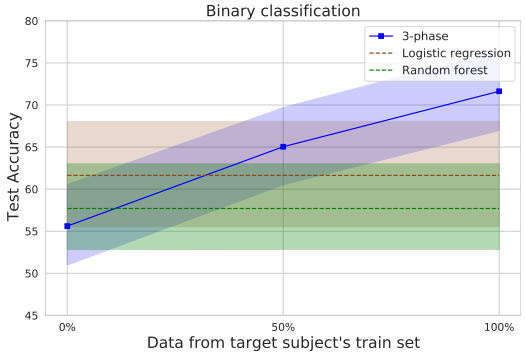


Fig. 7. **Binary Classification Accuracy as Subj.-Specific Training Data Grows**: Y-axis indicates test set accuracy averaged across 15 possible target subjects. X-axis indicates the percentage of available training data used. Shaded regions show the 2.5th and 97.5th percentiles from 5000 bootstrap samples.

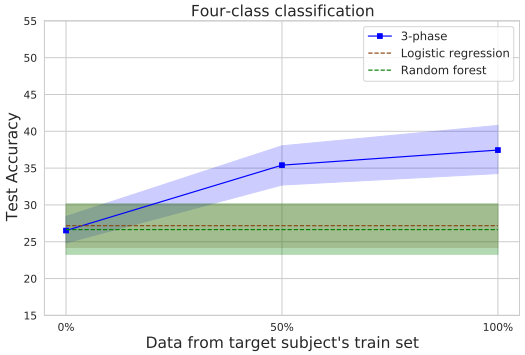


Fig. 8. **Four-class Classification Accuracy as Subj.-Specific Training Data Grows**: Y-axis indicates test set accuracy averaged across 15 possible target subjects. X-axis gives the percentage of training data used. Shaded regions show the 2.5th and 97.5th percentiles of 5000 bootstrap samples.

which uses observed data from other subjects. Our binary classifier improves from 62.75% to 67.00% by adding this phase; four-class classification improves from 27.90% to 32.96%.

Pretraining on synthetic data first – as in Phase 1 – improves performance even further. This is noticeable in both Table 1 and Table 2. For binary classification, after adding the MixUp phase we see accuracy improve from 67.00% (row "2 + 3") to 71.63% (row "1 + 2 + 3"). For four-class classification, we see improvements from 32.96% (row "2 + 3") to 37.46% (row "1 + 2 + 3").

Our 3-phase approach can be effective even with far less subject-specific data. Table 1 (binary classification) shows that even with only 2.5 minutes of subject-specific training data, our 3-phase approach can outperform subject-specific methods given *double* that amount of training data *from the same session*. Figure 7 and Figure 8 further show this visually, plotting average accuracy as a function of the amount of subject-specific data our method is given. This demonstrates our method's effectiveness in reducing individual calibration effort, long known to be a challenge in BCI applications.

Subject-specific calibration is needed. We have tried building general-purpose "subject-independent" models, which use the first two phases of our pipeline but use no training data for the target subject (see rows marked "1 + 2" in Table 1 and Table 2). We find performance is rarely better than random guessing: 55.61% accuracy for binary classification, 26.52% accuracy for four-class classification.

6 SUPPLEMENTARY RESULTS

We used the demographic and contextual information (see Appendix A.3), experiment log (see Appendix A.4), and post-experiment interview (see Appendix A.5) to determine which subjects met our eligibility criteria and to make note of any factors during the experiment that may have affected the signal quality or subjects' task performance. We used the performance results (see Appendix A.1) and subjective workload results (see Appendix A.2) to confirm that the different levels of n-back tasks induced different amounts of mental workload. All of these data are available in our open-access release and can be used for future studies.

7 CONCLUSION

We have presented new tools and a new dataset intended to allow future researchers and designers to create and explore fNIRS-based BCI applications more easily. Some examples of the *implications for design* of this work are found in such previous experimental fNIRS-based systems as: Brainput for robot automation [56], air traffic control [2], music learning [67], and bubble cursor usage [3]. We provide a new fNIRS dataset collected using rigorous procedures from subjects performing a standard n-back cognitive task and show how it can be used to improve performance for future systems. We developed a new machine learning approach to process and utilize fNIRS data. We show from our experiments that our proposed 3-phase machine learning pipeline significantly improves n-back task classification performance over several established baselines. Moreover, even with a reduced amount of per-user training data, our approach still outperforms baseline models trained with all available target-subject-specific training data, showing the potential of reducing individual calibration effort when deploying BCI applications in the future. We hope that our new dataset, machine learning method, and tools will remove barriers that have prevented a wider range of researchers from using fNIRS-based BCI and facilitate the development of a new generation of powerful and easy to use brain-computer interfaces.

ACKNOWLEDGMENTS

We thank our colleagues Shuren Wang, Chengmei Zhu, Matt Russell, Tomoki Shibata, David Guy Brizan, Beste Filiz Yuksel, Cong Liu, Ruijie Jiang, Leli Zhou, Tianzi Zhou, Sergio Fantini, Angelo Sassaroli, Leon Deligiannidis, Jones Yu, and Alex Olwal for their help and collaboration; and Google Inc. for support of this research. This research supported in part by the U.S. National Science Foundation under award HDR-1934553.

REFERENCES

[1] Berdakh Abibullaev and Jinung An. 2012. Classification of frontal cortex haemodynamic responses during cognitive tasks using wavelet transforms and machine learning algorithms. *Medical engineering & physics* 34, 10 (2012), 1394–1410.

[2] Daniel Afergan, Evan M Peck, Erin T Solovey, Andrew Jenkins, Samuel W Hincks, Eli T Brown, Remco Chang, and Robert JK Jacob. 2014. Dynamic difficulty using brain metrics of workload. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3797–3806.

[3] Daniel Afergan, Tomoki Shibata, Samuel W Hincks, Evan M Peck, Beste F Yuksel, Remco Chang, and Robert JK Jacob. 2014. Brain-based target expansion. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 583–593.

[4] Haleh Aghajani, Marc Garbey, and Ahmet Omurtag. 2017. Measuring mental workload with EEG+ fNIRS. *Frontiers in human neuroscience* 11 (2017), 359.

[5] Kai Keng Ang, Cuntai Guan, Kerry Lee, Jie Qi Lee, Shoko Nioka, and Britton Chance. 2010. Application of rough set-based neuro-fuzzy system in nirs-based bci for assessing numerical cognition in classroom. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–7.

[6] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. 2019. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*. PMLR, 312–321.

[7] Mahnaz Arvaneh, Cuntai Guan, Kai Keng Ang, and Chai Quek. 2011. Optimizing the channel selection and classification accuracy in EEG-based BCI. *IEEE Transactions on Biomedical Engineering* 58, 6 (2011), 1865–1873.

[8] Johann Benerradi, Horia A. Maior, Adrian Marinescu, Jeremie Clos, and Max L. Wilson. 2019. Exploring machine learning approaches for classifying mental workload using fNIRS data from HCI tasks. In *Proceedings of the Halfway to the Future Symposium 2019*. 1–11.

[9] Chris Berka, Daniel J Levendowski, Michelle N Lumicao, Alan Yau, Gene Davis, Vladimir T Zivkovic, Richard E Olmstead, Patrice D Tremoulet, and Patrick L Craven. 2007. EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks. *Aviation, space, and environmental medicine* 78, 5 (2007), B231–B244.

[10] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*. 5049–5059.

[11] I. J. Bigio and Sergio Fantini. 2016. *Quantitative Biomedical Optics*. Cambridge University Press, Cambridge, UK.

[12] Giles Blaney, Angelo Sassaroli, and Sergio Fantini. 2020. Design of a source–detector array for dual-slope diffuse optical imaging. *Review of Scientific Instruments* 91, 9 (2020), 093702. https://doi.org/10.1063/5.0015512

[13] Giles Blaney, Angelo Sassaroli, Thao Pham, Cristianne Fernandez, and Sergio Fantini. 2020. Phase dual-slopes in frequency-domain near-infrared spectroscopy for enhanced sensitivity to brain tissue: First applications to human subjects. *Journal of Biophotonics* 13, 1 (jan 2020). https://doi.org/10.1002/jbio.201960018

[14] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).

[15] Ricardo Buettner. 2013. Cognitive workload of humans using artificial intelligence systems: towards objective measurement applying eye-tracking technology. In *Annual conference on artificial intelligence*. Springer, 37–48.

[16] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. 2018. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718* (2018).

[17] Frédéric Dehais, Alban Dupres, Gianluca Di Flumeri, Kevin Verdiere, Gianluca Borghini, Fabio Babiloni, and Raphalle Roy. 2018. Monitoring pilot's cognitive fatigue with engagement features in simulated and actual flight conditions using an hybrid fNIRS-EEG passive BCI. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 544–549.

[18] Zach Eaton-Rosen, Felix Bragman, Sebastien Ourselin, and M Jorge Cardoso. 2018. Improving data augmentation for medical image segmentation. (2018).

[19] Lex Fridman, Bryan Reimer, Bruce Mehler, and William T. Freeman. 2018. Cognitive Load Estimation in the Wild. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–9. https://doi.org/10.1145/3173574.3174226

[20] Audrey Girouard, Erin Treacy Solovey, Leanne M Hirshfield, Krysta Chauncey, Angelo Sassaroli, Sergio Fantini, and Robert JK Jacob. 2009. Distinguishing difficulty levels with non-invasive brain activity measurements. In *IFIP Conference on Human-Computer Interaction*. Springer, 440–452.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[22] Christoph Guger, Alois Schlogl, Christa Neuper, Dirk Walterspacher, Thomas Strein, and Gert Pfurtscheller. 2001. Rapid prototyping of an EEG-based brain-computer interface (BCI). *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 9, 1 (2001), 49–58.

[23] Mehdi Hajinoroozi, Zijing Mao, Tzyy-Ping Jung, Chin-Teng Lin, and Yufei Huang. 2016. EEG-based prediction of driver's cognitive performance by deep convolutional neural network. *Signal Processing: Image Communication* 47 (2016), 549–555.

[24] Dong-Kyun Han and Ji-Hoon Jeong. 2020. Domain Generalization for Session-Independent Brain-Computer Interface. *arXiv preprint arXiv:2012.03533* (2020).

[25] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage publications Sage CA: Los Angeles, CA, 904–908.

[26] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. 2019. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 558–567.

[27] Johannes Hennrich, Christian Herff, Dominic Heger, and Tanja Schultz. 2015. Investigating deep learning for fNIRS based BCI. In *2015 37th Annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2844–2847.

[28] Christian Herff, Dominic Heger, Ole Fortmann, Johannes Hennrich, Felix Putze, and Tanja Schultz. 2014. Mental workload during n-back task—quantified in the prefrontal cortex using fNIRS. *Frontiers in human neuroscience* 7 (2014), 935.

[29] Christian Herff, Felix Putze, Dominic Heger, Cuntai Guan, and Tanja Schultz. 2012. Speaking mode recognition from functional near infrared spectroscopy. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 1715–1718.

[30] Thi Kieu Khanh Ho, Jeonghwan Gwak, Chang Min Park, and Jong-In Song. 2019. Discrimination of mental workload levels from multi-channel fNIRS using deep leaning-based approaches. *IEEE Access* 7 (2019), 24392–24403.

[31] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR, 448–456.

[32] Kosar Khaksari, Emma Condy, John Millerhagen, Afrouz Anderson, Hadis Dashtestsni, and Amir Gandjbakhche. 2019. Effects of Performance and Task Duration on Mental Workload during Working Memory Task. *Photonics* 6 (08 2019), 94. https://doi.org/10.3390/photonics6030094

[33] Bonkon Koo, Hanh Vu, Hwan-Gon Lee, Hyung-Cheul Shin, and Seungjin Choi. 2016. Motor imagery detection with wavelet analysis for NIRS-based BCI. In *2016 4th International Winter Conference on Brain-Computer Interface (BCI)*. IEEE, 1–4.

[34] Demetres Kostas and Frank Rudzicz. 2020. Thinker invariance: enabling deep neural networks for BCI across more people. *Journal of Neural Engineering* 17, 5 (2020), 056008.

[35] Jinuk Kwon and Chang-Hwan Im. 2021. Subject-Independent Functional Near-Infrared Spectroscopy-Based Brain–Computer Interfaces Based on Convolutional Neural Networks. *Frontiers in human neuroscience* 15 (2021), 121.

[36] O-Yeon Kwon, Min-Ho Lee, Cuntai Guan, and Seong-Whan Lee. 2019. Subject-independent brain–computer interfaces based on deep convolutional neural networks. *IEEE transactions on neural networks and learning systems* 31, 10 (2019), 3839–3852.

[37] VJ Lawhern, AJ Solon, NR Waytowich, SM Gordon, CP Hung, and BJ Lance. 2016. EEGNet: a compact convolutional network for EEG-based brain-computer interfaces. arXiv. *arXiv preprint arXiv:1611.08024* (2016).

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

[39] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[40] Hanxiao Liu, Andrew Brock, Karen Simonyan, and Quoc V Le. 2020. Evolving normalization-activation layers. *arXiv preprint arXiv:2004.02967* (2020).

[41] Boyang Lyu, Thao Pham, Giles Blaney, Zachary Haga, Angelo Sassaroli, Sergio Fantini, and Shuchin Aeron. 2021. Domain adaptation for robust workload level alignment between sessions and subjects using fNIRS. *Journal of Biomedical Optics* 26, 2 (2021), 022908.

[42] Sandra P Marshall. 2002. The index of cognitive activity: Measuring cognitive workload. In *Proceedings of the IEEE 7th conference on Human Factors and Power Plants*. IEEE, 7–7.

[43] Kimberly Meidenbauer, Kyoung Whan Choe, Carlos Cardenas-Iniguez, Theodore Huppert, and Marc Berman. 2021. Load-Dependent Relationships between Frontal fNIRS Activity and Performance: A Data-Driven PLS Approach. *NeuroImage* 230 (01 2021), 117795. https://doi.org/10.1016/j.neuroimage.2021.117795

[44] Atsuo Murata. 2005. An attempt to evaluate mental workload using wavelet transform of EEG. *Human Factors* 47, 3 (2005), 498–508.

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[47] Mirka Pesonen, Heikki Hämäläinen, and Christina M. Krause. 2007. Brain oscillatory 4-30 Hz responses during a visual n-back memory task with varying memory load. *Brain research* 1138 (04 2007), 171–7. https://doi.org/10.1016/j.brainres.2006.12.076

[48] Thao Thanh Pham, Thang Duc Nguyen, and Toi Van Vo. 2015. Sparse fNIRS feature estimation via unsupervised learning for mental workload classification. In *International Workshop on Neural Networks*. Springer, 283–292.

[49] Sarah D Power, Tiago H Falk, and Tom Chau. 2010. Classification of prefrontal activity due to mental arithmetic and music imagery using hidden Markov models and frequency domain near-infrared spectroscopy. *Journal of neural engineering* 7, 2 (2010), 026002.

[50] Bryan Reimer and Bruce Mehler. 2013. The Effects of a Production Level "Voice-Command" Interface on Driver Behavior: Summary Findings on Reported Workload, Physiology, Visual Attention, and Driving Performance. (11 2013).

[51] Marjan Saadati, Jill Nelson, and Hasan Ayaz. 2019. Mental workload classification from spatial representation of fnirs recordings using convolutional neural networks. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[52] RT Schirrmeister, JT Springenberg, and T Ball. 2018. Deep learning with convolutional neural networks for brain mapping and decoding of movement-related information from the human EEG. *arXiv preprint arXiv:1703.05051* (2018).

[53] Yu Shi, Natalie Ruiz, Ronnie Taib, Eric Choi, and Fang Chen. 2007. Galvanic skin response (GSR) as an index of cognitive load. In *CHI'07 extended abstracts on Human factors in computing systems*. 2651–2656.

[54] Jaeyoung Shin, Alexander Von Lühmann, Do-Won Kim, Jan Mehnert, Han-Jeong Hwang, and Klaus-Robert Müller. 2018. Simultaneous acquisition of EEG and NIRS during cognitive tasks for an open access dataset. *Scientific data* 5, 1 (2018), 1–16.

[55] Winnie KY So, Savio WH Wong, Joseph N Mak, and Rosa HM Chan. 2017. An evaluation of mental workload with frontal EEG. *PloS one* 12, 4 (2017), e0174949.

[56] Erin Solovey, Paul Schermerhorn, Matthias Scheutz, Angelo Sassaroli, Sergio Fantini, and Robert Jacob. 2012. Brainput: enhancing interactive systems with streaming fnirs brain input. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 2193–2202.

[57] Erin Treacy Solovey and Robert JK Jacob. 2011. Meaningful Human-Computer Interaction Using fNIRS Brain Sensing. In *ACM CHI Conf on Human Factors in ComputingSystems*.

[58] Erin Treacy Solovey, Francine Lalooses, Krysta Chauncey, Douglas Weaver, Margarita Parasi, Matthias Scheutz, Angelo Sassaroli, Sergio Fantini, Paul Schermerhorn, Audrey Girouard, et al. 2011. Sensing cognitive multitasking for a brain-based adaptive user interface. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 383–392.

[59] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[60] John Sweller, Paul Ayres, and Slava Kalyuga. 2011. Measuring cognitive load. In *Cognitive load theory*. Springer, 71–85.

[61] A. Tattersall and P. S. Foord. 1996. An experimental evaluation of instantaneous self-assessment as a measure of workload. *Ergonomics* 39 5 (1996), 740–8.

[62] Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *arXiv preprint arXiv:1905.11001* (2019).

[63] Kevin J Verdière, Raphaëlle N Roy, and Frédéric Dehais. 2018. Detecting pilot's engagement using fNIRS connectivity features in an automated vs. manual landing scenario. *Frontiers in human neuroscience* 12 (2018), 6.

[64] Nikolai von Janczewski, Jennifer Wittmann, Arnd Engeln, Martin Baumann, and Lutz Krauß. 2021. A meta-analysis of the n-back task while driving and its effects on cognitive workload. *Transportation research part F: traffic psychology and behaviour* 76 (2021), 269–285.

[65] Huijuan Yang, Siavash Sakhavi, Kai Keng Ang, and Cuntai Guan. 2015. On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and*

Biology Society (EMBC). IEEE, 2620–2623.

[66] Beste F Yuksel, Daniel Afergan, Evan M Peck, Garth Griffin, Lane Harrison, Nick WB Chen, Remco Chang, and Robert JK Jacob. 2015. Braahms: a novel adaptive musical interface based on users' cognitive state.. In *NIME*. 136–139.

[67] Beste F Yuksel, Kurt B Oleson, Lane Harrison, Evan M Peck, Daniel Afergan, Remco Chang, and Robert JK Jacob. 2016. Learn piano with BACh: An adaptive learning interface that adjusts task difficulty based on brain state. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 5372–5384.

[68] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[69] Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. 2019. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321* (2019).

Appendices

A SUPPLEMENTARY RESULTS

A.1 Performance

We calculated the average and standard deviation of accuracy of all four types of n-back tasks for both subject groups. The results show a negative correlation between n (0, 1, 2, 3) and accuracy, which confirms that the n-back tasks were an effective way to induce different difficulty levels under the control environment. For detailed results, see Table A.1.

Paradigm	Measurement	Qualified Subjects	All Subjects
0-back	Avg	0.971	0.968
1-back	Avg	0.958	0.962
2-back	Avg	0.873	0.886
3-back	Avg	0.791	0.797
0-back	SD	0.129	0.136
1-back	SD	0.045	0.042
2-back	SD	0.095	0.088
3-back	SD	0.076	0.108

Table A.1. Task specific mean and standard deviation of n-back task performance for both subject group

A.2 Subjective workload

The mental demand dimension of the NASA-TLX can be interpreted as a measurement of perceived mental workload. We observed that for both groups, the n-back tasks with larger n values were rated significantly more mentally demanding. For the eligible subjects, the mean values of reported mental demand for 3-back, 2-back, 1-back and 0-back tasks are 12.63, 7.73, 3.68 and 0.58 respectively. A subsequent t-test also shows significant difference between 0-backs and 2-backs ($t = -15.43, p < 0.001$) in a binary classification scenario.

Table A.2 and A.3 show the detailed average and standard deviation for both subject groups on each n-back task. For detailed t-test results between 0-backs and 2-backs on four components of NASA-TLX for both subject groups, see Table A.4.

Paradigm	Measurement	Mental Demand	Performance (the larger the worse)	Effort	Frustration
0-back	Avg	0.58	0.92	1.08	1.06
1-back	Avg	3.68	3.58	4.17	2.70
2-back	Avg	7.73	8.62	8.52	5.65
3-back	Avg	12.63	14.00	12.40	8.80
0-back	SD	1.41	2.84	2.56	2.74
1-back	SD	3.18	3.51	3.18	3.12
2-back	SD	3.27	3.82	3.46	3.80
3-back	SD	3.83	3.31	3.35	5.22

Table A.2. Task specific mean and standard deviation of NASA TLX for qualified subjects

Paradigm	Measurement	Mental Demand	Performance (the larger the worse)	Effort	Frustration
0-back	Avg	0.88	1.53	1.27	1.61
1-back	Avg	4.15	4.06	4.57	3.38
2-back	Avg	8.36	8.42	8.88	6.39
3-back	Avg	13.16	13.32	12.88	9.65
0-back	SD	1.56	3.64	2.28	2.77
1-back	SD	3.07	3.86	3.15	3.15
2-back	SD	3.69	3.88	3.75	4.07
3-back	SD	3.72	3.99	3.51	5.40

Table A.3. Task specific mean and standard deviation of NASA TLX for all subjects

t test	Subject Group	Mental Demand	Performance (the larger the worse)	Effort	Frustration
t	Qualified	-15.43	-12.44	-13.25	-7.51
p	Qualified	<0.001	<0.001	<0.001	<0.001
t	All	-19.35	-13.40	-17.95	-10.14
p	All	<0.001	<0.001	<0.001	<0.001

Table A.4. Binary 0-back and 2-back T-test result of NASA-TLX for both subject groups

A.3 Demographics

A total of twenty seven subjects (18 females, mean age of 20, SD of 2.3) participated in the study. A subset of 15 subjects are qualified (13 females, mean age of 19.9, SD of 2.8). All the subjects are right-handed. One eligible subject and 3 ineligible subjects reported recreational drug (marijuana or alcohol) usage in the past week. All the subjects are healthy without any head injury history. For detailed information, see our open access dataset.

A.4 Experiment log

All the conditions mentioned are recorded (see our open access dataset for details).

A.5 Post-experiment interview

All the conditions mentioned are recorded (see our open access dataset for details).

B DATA-PREPROCESSING METHOD

A measurement of SD I ($SDI(\rho,\lambda) = ln(\rho^2 I(\rho,\lambda))$) or SD ϕ ($SD\phi(\rho,\lambda) = \phi(\rho,\lambda)$) will be referred to as SDY for simplicity. The average slope of SDY versus ρ in the DS set (DSY) is calculated:

$$DSY(\lambda) = \frac{1}{2}\left(\frac{SDY_1(\rho_2,\lambda) - SDY_1(\rho_1,\lambda)}{\rho_2 - \rho_1} + \frac{SDY_2(\rho_2,\lambda) - SDY_2(\rho_1,\lambda)}{\rho_2 - \rho_1}\right) \tag{2}$$

where $SDY_i(\rho_j)$ is the i^{th} SD measurement at ρ_j. This $DSY(\lambda)$ is measured during a baseline period to yield $DSY_0(\lambda)$. Changes from this baseline ($\Delta DSY(\lambda,t) = DSY(\lambda,t) - DSY_0(\lambda)$) were used to calculate $\Delta\mu_{a,Y}(\lambda,t)$ at time t:

$$\Delta\mu_{a,Y}(\lambda,t) = -\frac{\Delta DSY(\lambda,t)}{DSF_Y(\lambda)} \tag{3}$$

where $DSF_Y(\lambda)$ is the differential slope factor [13] which depends on the absolute optical properties of the tissue. An absorption coefficient of $\mu_a = 0.01$ 1/mm and a reduced scattering coefficient of $\mu_s^{'} = 1$ 1/mm were assumed for each wavelength. Finally, $\Delta[HbO]_Y(t)$ and $\Delta[HbR]_Y(t)$ were calculated:

$$\begin{bmatrix} \epsilon_{[HbO]}(\lambda_1) & \epsilon_{[HbR]}(\lambda_1) \\ \epsilon_{[HbO]}(\lambda_2) & \epsilon_{[HbR]}(\lambda_2) \end{bmatrix}^{-1} \begin{bmatrix} \Delta\mu_{a,Y}(\lambda_1,t) \\ \Delta\mu_{a,Y}(\lambda_2,t) \end{bmatrix} = \begin{bmatrix} \Delta[HbO]_Y(t) \\ \Delta[HbR]_Y(t) \end{bmatrix} \tag{4}$$

where $\epsilon_C(\lambda_i)$ is the extinction coefficient for chromophore C at wavelength λ_i[11]. All further analyses utilized these DS-derived changes of chromophore concentrations (*i.e.* $[HbO]_Y(t)$ and $[HbR]_Y(t)$) as input.

C MACHINE LEARNING AND DEEP LEARNING SETTINGS

C.1 CNN details.

For all methods that use a CNN, we use the same architecture with 1 convolutional layer followed by 2 fully connected layers. We search 3 values (2, 4, 6) for the filter size of the convolutional layer, 3 values (1, 2, 3) for the filter stride of the convolutional layer, 3 values (10, 20, 40) for the output size of the convolutional layer, and 3 values (10, 20, 40) for the size of the fully connected layers.

To avoid overfitting, we apply dropout [59] after the fully connected layer. To give the subject-specific baseline CNN more advantage, we search three possible dropout values (0.2, 0.5, and 0.7), while for our 3-phase approach, we fix the dropout rate at 0.2.

We train all CNNs using empirical risk minimization with the cross-entropy loss. Parameters are estimated via stochastic gradient descent optimizer with a fixed momentum of 0.9. We search SGD's learning rate in (0.003, 0.01, 0.3). Our CNN implementation uses the PyTorch software framework [45].

C.2 Augmentation details.

For methods that involve the first data augmentation phase, we search 3 values (0.3, 0.75, 0.9) for the alpha value of beta distribution and 3 values (2, 4, 8) that control how many times we expand the dataset, leading to 28, 56 or 112 synthetic subjects in Phase 1.

C.3 Logistic Regression details.

To obtain features for Logistic Regression (LR), we summarize the time-series for each window w and each channel d with four numbers: the mean, the standard deviation and the slope and intercept of a linear regression fit to the measurements over time (minimizing squared error).

To train LR, we perform *penalized* empirical risk minimization using the cross-entropy loss function and an L2-penalty on the weight coefficients. Parameters are estimated via the L-BFGS algorithm. We search 11 possible values for the L2 penalty strength, logarithmically spaced from -5 to 5. All other settings use the defaults in the SciKit-Learn software package [46].

C.4 Random Forest details.

To obtain features for Random Forest (RF) classifier, like we did for LR, we summarize the time-series for each window w and each channel d with four numbers: the mean, the standard deviation and the slope and intercept of a linear regression fit to the measurements over time (minimizing squared error).

We search 3 values (100, 500, 1000) for number of estimators, which controls how many tree estimators are used in the forest, 4 values (1, 5, 10, 20) for the minimum number of samples required to split an internal node. All other settings use the defaults in the SciKit-Learn software package [46].

D VALIDATION OF OUR PIPELINE ON AN EXTERNAL DATASET

Here, we present the results of our cross-subject pipeline on the only other open-access fNIRS cognitive workload dataset we're aware of, a 26 subject dataset that measures fNIRS performance on n-back tasks released by Shin et al. [54].

The dataset contains the fNIRS data of 27 tasks, 9 tasks for each of 0-back, 2-back and 3-back tasks. Each task consists of 765 rows and 32 features. This dataset is not suitable for the Dual-Slope pre-processing algorithm we used for our dataset. We generate slide windows with the window size of 20 timesteps and the window stride of 3 timesteps to make each window cover the data in ~2 seconds, the same as the experiment on our own dataset.

In the binary classification, we use the 10 tasks (5 0-back tasks and 5 2-back tasks) as the training set and test the model on 8 tasks (4 0-backs and 4 2-backs).

Paradigm	Model	Avg. Acc on Target Subj.'s Test Set
Subj.-Specific	CNN	60.80
Last 2-phase	CNN	68.65
3-phase	CNN	70.69

Table D.1. Binary classification result on external dataset

E LATIN SQUARE OF EXPERIMENT TASK SEQUENCE

Below we present the Latin square applied in n-back task experiment design. It is a 4 × 4 array filled with 4 different n-back tasks (0, 1, 2, and 3), each occurring exactly once in each row and exactly once in each column.

Fig. E.1. All subjects completed 16 n-back tasks; each value of n was determined according to this order.

F DETAILED CLASSIFICATION RESULTS BY SUBJECT

Below we show the detailed classification results for each subject for all methods compared. F.1 shows the binary classification results. F.2 shows the four-class classification results.

Labeled Training Data Used								
From Other Subjects	From Target Subj.	Paradigm	Model	Sub1	Sub2	Sub3	Sub4	Sub5
0	5 min. (100% train set)	Subj.-Specific	Logistic Regression	79.68	72.66	46.94	54.86	62.59
0	5 min. (100% train set)	Subj.-Specific	Random Forest	80.04	75.54	47.84	53.06	51.08
0	5 min. (100% train set)	Subj.-Specific	CNN	76.44	66.01	64.93	53.06	58.63
14 subjects, 5 min. each	5 min. (100% train set)	Last 2-phase	CNN	76.08	67.99	51.80	56.47	60.61
14 subjects, 5 min. each	5 min. (100% train set)	3-phase	CNN	78.06	74.10	62.95	58.45	64.21
14 subjects, 5 min. each	2.5 min. (50% train set)	3-phase	CNN	76.62	80.58	54.86	50.90	56.83
14 subjects, 5 min. each	0 min. (0% train set)	First 2-phase	CNN	55.94	50.18	71.58	48.02	72.84
				Sub6	Sub7	Sub8	Sub9	Sub10
0	5 min. (100% train set)	Subj.-Specific	Logistic Regression	49.82	65.29	51.80	52.70	87.77
0	5 min. (100% train set)	Subj.-Specific	Random Forest	54.68	39.75	55.76	51.26	61.69
0	5 min. (100% train set)	Subj.-Specific	CNN	57.37	73.38	58.99	53.96	63.67
14 subjects, 5 min. each	5 min. (100% train set)	Last 2-phase	CNN	65.83	72.84	62.23	56.12	70.50
14 subjects, 5 min. each	5 min. (100% train set)	3-phase	CNN	72.66	72.30	62.59	57.91	82.19
14 subjects, 5 min. each	2.5 min. (50% train set)	3-phase	CNN	66.19	73.38	57.19	59.71	62.41
14 subjects, 5 min. each	0 min. (0% train set)	First 2-phase	CNN	46.40	41.37	50.00	58.09	53.23
				Sub11	Sub12	Sub13	Sub14	Sub15
0	5 min. (100% train set)	Subj.-Specific	Logistic Regression	70.68	58.27	52.52	45.50	73.56
0	5 min. (100% train set)	Subj.-Specific	Random Forest	57.37	55.58	55.00	67.09	59.71
0	5 min. (100% train set)	Subj.-Specific	CNN	79.68	57.55	50.00	57.91	69.60
14 subjects, 5 min. each	5 min. (100% train set)	Last 2-phase	CNN	88.67	66.01	75.36	63.85	70.68
14 subjects, 5 min. each	5 min. (100% train set)	3-phase	CNN	89.03	68.35	87.05	69.78	74.82
14 subjects, 5 min. each	2.5 min. (50% train set)	3-phase	CNN	54.32	66.01	76.98	67.81	71.76
14 subjects, 5 min. each	0 min. (0% train set)	First 2-phase	CNN	50.00	66.01	48.38	53.78	68.35

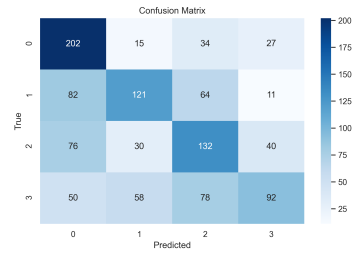
Table F.1. Subject specific performance for Binary classification

Labeled Training Data Used								
From Other Subjects	From Target Subj.	Paradigm	Model	Sub1	Sub2	Sub3	Sub4	Sub5
0	10 min. (100% train set)	Subj.-Specific	Logistic Regression	36.33	38.04	19.42	25.27	23.20
0	10 min. (100% train set)	Subj.-Specific	Random Forest	40.74	37.86	23.47	26.08	26.98
0	10 min. (100% train set)	Subj.-Specific	CNN	33.54	26.89	34.17	26.26	27.79
14 subjects, 10 min. each	10 min. (100% train set)	Last 2-phase	CNN	39.39	36.24	29.68	28.24	33.0.9
14 subjects, 10 min. each	10 min. (100% train set)	3-phase	CNN	49.19	38.94	33.99	32.64	35.88
14 subjects, 10 min. each	5 min. (50% train set)	3-phase	CNN	46.58	41.82	38.49	30.58	37.41
14 subjects, 10 min. each	0 min. (0% train set)	First 2-phase	CNN	26.62	25.63	31.74	27.88	35.52
				Sub6	Sub7	Sub8	Sub9	Sub10
0	10 min. (100% train set)	Subj.-Specific	Logistic Regression	31.03	24.55	25.00	25.09	33.63
0	10 min. (100% train set)	Subj.-Specific	Random Forest	21.40	14.93	29.32	25.81	29.05
0	10 min. (100% train set)	Subj.-Specific	CNN	25.09	22.48	25.09	24.91	28.24
14 subjects, 10 min. each	10 min. (100% train set)	Last 2-phase	CNN	28.06	25.54	25.54	31.56	32.10
14 subjects, 10 min. each	10 min. (100% train set)	3-phase	CNN	37.05	35.79	25.27	30.49	36.87
14 subjects, 10 min. each	5 min. (50% train set)	3-phase	CNN	32.28	39.21	36.87	25.90	35.70
14 subjects, 10 min. each	0 min. (0% train set)	First 2-phase	CNN	24.82	21.22	29.14	23.56	23.65
				Sub11	Sub12	Sub13	Sub14	Sub15
0	10 min. (100% train set)	Subj.-Specific	Logistic Regression	28.60	23.92	16.37	27.34	29.95
0	10 min. (100% train set)	Subj.-Specific	Random Forest	17.36	23.83	31.56	30.22	21.13
0	10 min. (100% train set)	Subj.-Specific	CNN	27.43	26.17	28.68	27.79	33.90
14 subjects, 10 min. each	10 min. (100% train set)	Last 2-phase	CNN	38.85	34.53	32.28	36.15	42.36
14 subjects, 10 min. each	10 min. (100% train set)	3-phase	CNN	45.32	30.22	46.58	38.49	45.14
14 subjects, 10 min. each	5 min. (50% train set)	3-phase	CNN	34.71	26.98	31.03	36.06	37.32
14 subjects, 10 min. each	0 min. (0% train set)	First 2-phase	CNN	28.96	24.10	25.81	26.35	22.84

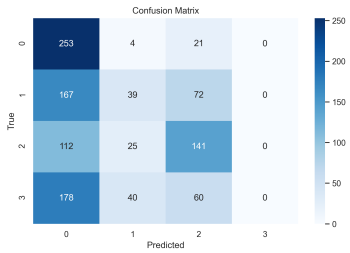
Table F.2. Subject specific performance for Four-Class classification

G CONFUSION MATRIX FOR FOUR-CLASS CLASSIFICATION

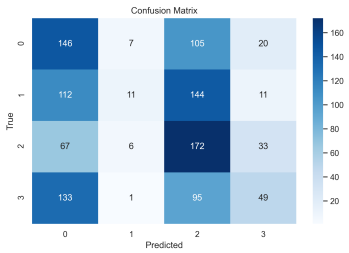
Figure G.1 shows the confusion matrix of each subject for our proposed 3-phase approach on four-class classification (corresponding to the scenario in the 5th row in Figure 2).



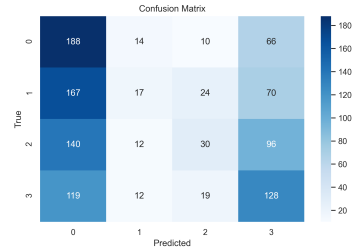
(a) sub_1



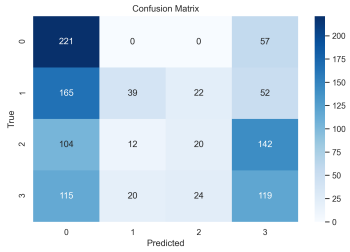
(b) sub_2



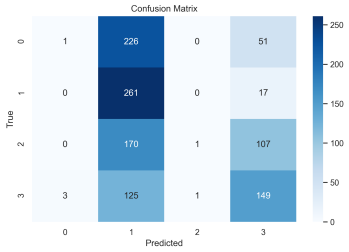
(c) sub_3



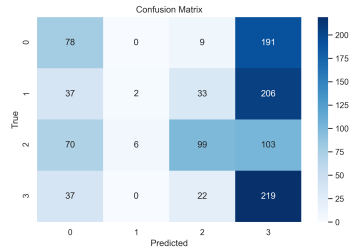
(d) sub_4



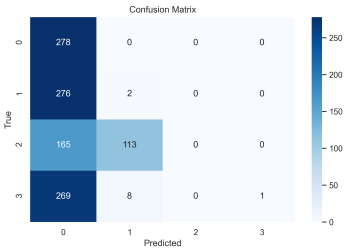
(e) sub_5



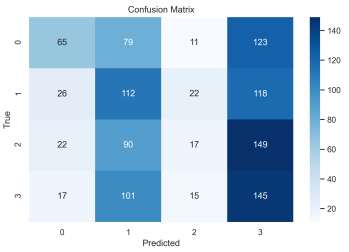
(f) sub_6



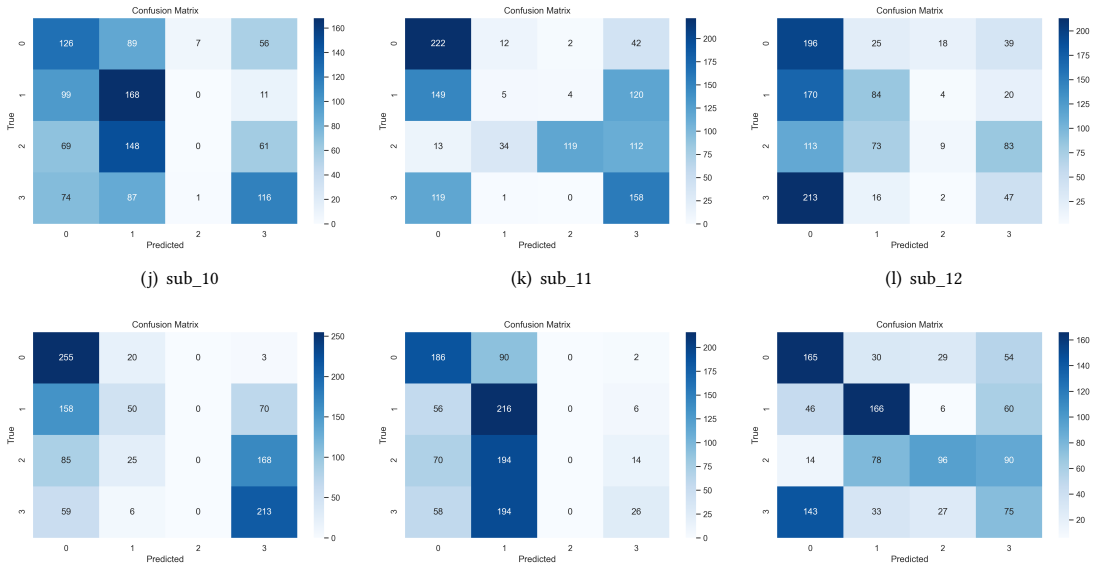
(g) sub_7



(h) sub_8



(i) sub_9



(j) sub_10 (k) sub_11 (l) sub_12

(m) sub_13 (n) sub_14 (o) sub_15

Fig. G.1. Confusion matrices for four-class classification using the 3-phase approach. Each subfigure is the confusion matrix of a subject. The subject's model is trained with 100% of the subject's train data.