Transfer Learning for Brain-Computer Interfaces in Cyber-Physical Systems

Ekansh Gupta, Mohit Agarwal, Raghupathy Sivakumar
Georgia Institute of Technology, Atlanta, Georgia
Email: egupta8@gatech.edu, me.agmohit@gmail.com, siva@gatech.edu

***Abstract*—Brain-Computer Interfaces (BCI) have emerged as a promising technology for enabling seamless human-machine communication. They bypass the need for explicit effort from a user and thus are apt as interfacing technology for wearables, IoT systems, body-area networks, etc. On account of this, BCIs have seen a steady rise in entertainment, health and wellness, and security applications. However, despite their growing popularity, they suffer from a key problem that prevents them from being a ubiquitous communication modality. BCIs often generalize poorly to new users owing to brain signals exhibiting a lot of variability among individuals. This makes creating standardized BCI models a challenge as they need to be re-trained/calibrated for each new end-user. In this paper, we present a method that not only shrinks this re-calibration overhead to a tiny fraction of the original but also provides a generalization accuracy that approaches that of intra-user models. We use transfer learning to transfer relevant parts of our detection model using divergence calculations in the signal probability space and use few-shot learning to adapt the model to a new user. We demonstrate our method on a dataset collected in our lab and obtain state-of-the-art results.**

***Index Terms*—Cyber-physical systems, Transfer-learning, Brain-Computer Interfaces**

I. Introduction

EEG-based BCI (Brain-Computer Interface) technology has emerged as an attractive modality for humans to communicate with computers and the real world. Over the past decades, humans have used multiple forms of methods to interact with computers, ranging from text input, keypresses, screen taps, and fingerprints to more recently, more spontaneous modalities such as voice inputs, hand gestures, etc. In this context, the human brain, being the architect of all thoughts, is a useful candidate to be tapped into and used for enabling technologies like cyber-physical systems and body area networks. BCI devices have been used extensively in body area network applications ranging from sleep quality detectors [1], wearables for measuring stress [2], hands-free authentication [3], etc. They are also used in cyber-physical systems for human-in-the-loop (HITL) models where a human's implicit reactions are recorded and communicated to a machine for applications like accelerating reinforcement learning agents [4], etc. BCIs are important for body-area networks and other cyber-physical systems as they provide the bridge between the user and the network without having the user explicitly communicate.

Despite the promise, BCI-based systems encounter a two-fold challenge. Since the EEG signals registered by BCI devices are recorded from over the scalp, they exhibit extraneous signals and suffer from poor SNR. This introduces challenges in the usability of devices and wearables as it leads to a decline in their performance in the form of reduced accuracy, improper readings, false alarms, etc. Secondly, there are significant individual differences in brain signals which makes them vary across users, tasks, environments, etc. As a result, it is more difficult to design universal BCI systems that work in an all-encompassing manner without the need for subject-wise calibration or re-training. For instance, a wearable that catalogs a user's mood, which is trained on a specific user's brain signals, may not be accurate for other users due to the variability of brain signals across individuals. This necessitates re-training or re-calibration of wearables with the data of new or unseen users, thereby wasting power and enormous amounts of time and effort for the user. To address the generalization problem, various solutions have been proposed, which range from using spatial filtering [5], and transfer learning using approaches like domain adaptation and optimal transport [6], to more recently, deep learning techniques as well. These approaches either do not provide sufficient transfer learning accuracy or are highly data-intensive, which again requires re-training over new datasets.

In this paper, we address the generalization problem of EEG-based BCIs in a human-in-the-loop (HITL) cyber-physical system. We create a cyber-physical system with HITL and utilize EEG-based BCI to provide input to the system from the human without having them explicitly communicate. In our IRB-approved user study, we have multiple test subjects observe an AI agent navigate a maze in an Atari-based game environment, where the agent takes occasional incorrect actions. We work with a specific brain signal called the Error Potential signal (ErrP), which is observed in a subject's brain after observing an erroneous response committed in a given environment [7]. We aim to improve the generalization detection accuracy of this signal. We use transfer learning to adapt detection models to unseen data using signal space probability, few-shot learning, and confidence interval-based sample augmentation. We show that our method approaches the accuracy of a standalone model while requiring only a fraction of the training overhead needed by such a model. More specifically, our research contributions are as follows:

1) We apply transfer learning to ErrP signals by reconciling two disparate signal distributions (corresponding to two different individuals) in the probability space.
2) We employ few-shot learning to improve the general-

ization accuracy of ErrP signals and we further improve the cross-user accuracy by augmenting our algorithm in an unsupervised manner using samples predicted within specific confidence intervals.

The remainder of the paper is structured as follows. Section II talks about the related work done for ErrP detection and transfer learning. Section III talks about our system setup and the data collection. Section IV shows our method using divergence analysis along with few-shot learning with classifier transfer and unsupervised augmentation and evaluates it. Section V discusses our results and compares them with existing methods, and finally, Section VI concludes the paper.

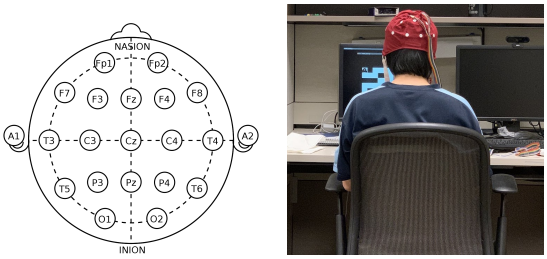
II. Related Work

Historically, signal detection for BCIs like the ErrP has been done using spatial filtering techniques [8]. Common Average Reference filters, Laplacian filters [9], and Common Spatial Patterns (CSP) have also been widely used as supervised spatial filtering techniques for EEG-based BCI signals.

The ErrP signal was discovered in 1991 when Falkenstein et al noticed a negative deflection in the recorded EEG (electroencephalogram) when subjects detected an error being committed by themselves during a choice-reaction task [7]. Error potentials are observed across a wide variety of input modalities and are well-founded in primates as the manifestation of a general-purpose error-control system [10]. [11] used xDAWN-based spatial filtering and tangent space projection to achieve state-of-the-art results for ErrP cross-user generalization. [5] introduced affine transforms, using which, they centered the reference covariance matrix for different users at the identity matrix, and obtained improved results for generalization. There have also been works that utilize combined detection [12] from multiple sources for improving ErrP generalization. Due to the cumbersome nature of collecting BCI data, there is generally a lack of availability of massive datasets pertaining to different brain signals. This has historically precluded deep learning and other data-intensive methods to achieve state-of-the-art performance for BCI compared to their spatial filtering counterparts. Having said that, in the recent past, deep learning models that use shallow architectures as well as fewer parameters have also outperformed spatial filtering approaches. For instance, [13] used generative adversarial networks (GANs) for object classification by using an EEG stream. [14] proposed EEGNet, a lightweight shallow CNN that outperformed previous CNN-based architectures [15] (DeepConvNet, ShallowConvNet) while using fewer parameters for 3 classification tasks. Nevertheless, the model proposed in [11] still achieved the best cross-user ErrP transfer accuracy. In this work, we use the XDAWN spatial filtering approach as a starting point and improve upon it to achieve transfer learning for ErrPs.

III. Experimental Setup and Data Collection

To collect our signals, we use an EEG-based BCI headset. It is a BCI cap with 20 electrodes with the corresponding electrode placement presented in Fig 1(a), with a refresh rate



(a) Electrode map (b) Study subject

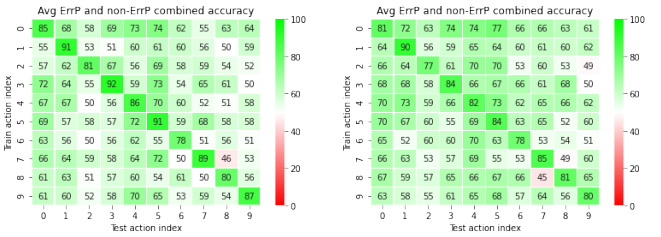
Fig. 1. Experimental framework

of 125Hz. To elicit the ErrP signal in the subjects' brains, we create an openAI-gym-based environment that shows a computer agent trying to navigate a maze. The agent is programmed to make occasional incorrect actions, which is being observed by a human wearing the BCI headset. The wrong moves by the agent elicit an ErrP signal in the user's brain. Every time the agent made a correct move, the signal elicited in the subjects' brains was tagged as a non-ErrP signal and for every wrong move, the corresponding brain signal was tagged as an ErrP signal. We ran the experiment on ten human subjects between the ages of 18-30 (mean age 26.7, 2 female). The total duration of an experiment per subject was about 45 minutes. The computer agent made moves at the interval of 1.5 seconds and the likelihood of making a mistake was chosen to be 0.20. This experiment and all its related protocols were approved by the Institutional Review Board.

To remove high-frequency noise, we pass the signals through a 4th-order Butterworth filter with cutoff frequencies equal to 0.5Hz and 40Hz. We then discard some channels that are not proximal to the area of the brain responsible for ErrP signals. We therefore select 10 electrodes namely Pz, P3, P4, F4, Fz, F3, Cz, C4, C3, and Fp2. We also choose a time window of 1.2 seconds after the onset of the agent move for processing the individual signals. To evaluate our method, we use balanced accuracy which is the average of specificity and sensitivity of a model, as it is more accurate for class imbalances (Non-ErrP signals in our experiment are 4 times as likely as ErrP signals as the probability of an incorrect move is 0.20) and punishes models which are overly selective or sensitive towards a specific class label.

IV. Method and Analysis

We start our analysis for cross-user ErrP transfer learning by using the xDAWN + Riemannian Geometry framework [11], as this method provided the highest cross-user accuracy results [14]. A brief explanation of this model is also provided in [14]. For training, xDAWN spatial filtering is applied on the raw signals to generate average signal templates [16]. Then, these templates are correlated with the individual raw signals to create covariance matrices for each signal sample. Then, electrode selection through backward elimination is performed [17] and finally, the matrices are projected on a Euclidean space and are classified using a linear classifier, like the ElasticNet [18]. The model pipeline is shown in Fig



(a) Cross-user accuracy matrix for xDAWN in %

(b) Cross-user accuracy matrix in % for xDAWN after affine transform and variance normalization

Fig. 2. Transfer learning generalization numbers for xDAWN algorithm before and after affine transform

3(a). As a starting point, we calculate the average per-user classification accuracy of the 10 users' data by performing a 5-fold cross-validation on each user and obtain an average balanced accuracy of 76.6%. For a transfer learning algorithm to be functional and practical, it should be reasonably close to this mark. We then calculate the inter-user classification accuracy pertaining to each user pair. In order to do this, we iteratively train our classifier on individual subjects while using the remaining subjects as test subjects. We present these accuracy results in Fig 2(a), where each cell can be interpreted as the accuracy achieved when a classifier is trained on the subject denoted by the row and tested on the subject denoted by the column. We calculate the average of this matrix and get a total average intra-user transfer accuracy of about 59.3%. The accuracy matrix for this experiment is given in Fig 2(a).

A. Augmenting Affine transforms with Variance normalization

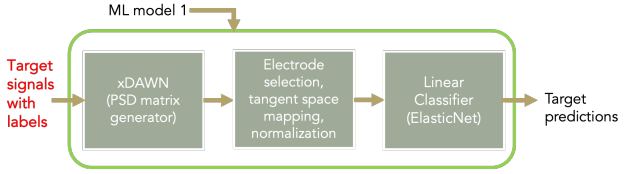
[5] showed that during the classification of PSD matrices across sessions or users, there are "shifts" in the reference points of data across sessions due to unavoidable parameters. This causes the entire dataset to shift with a constant distance which leads to poor generalization. Affine transforms counter this shift by centering the mean matrices of all users on the identity matrix. We augment this transformation by normalizing the variance of these signals. [19] defined the variance for a group of PSD matrices in the Riemann space as the expected value of the squared Riemannian distance from the mean.

$$\sigma^2 = \sum_{i=1}^{N} (\delta^2(P_i, M))/N, \tag{1}$$

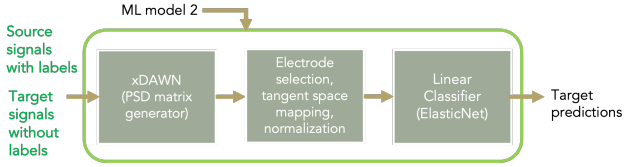
where P_i represents a sample matrix, M is the Riemannian mean of all such P_i's, and $\delta(P_i^{new}, M)$ is the Riemannian distance between P_i and M. To normalize this variance to 1, P_i is shifted such that its distance from M reduces by a σ. This can be achieved by computing P_i^{new} for every P_i which lies between P_i and M such that:

$$\delta(P_i^{new}, M) = \frac{\delta(P_i, M)}{\sigma}. \tag{2}$$

P_i^{new} is the weighted Riemannian mean between P_i and M where P_i is weighted by a factor of $\frac{1}{\sigma}$ and M is weighted by a factor of $\frac{\sigma-1}{\sigma}$. The results after affine transform and variance normalization are shown in Fig 2(b) with the average



(a) Model 1: XDAWN classifier default pipeline for target signals using supervised target samples



(b) Model 2: XDAWN classifier pipeline for target signals using supervised source and unsupervised target samples

Fig. 3. Detection model pipeline and the pipeline with the unsupervised target samples

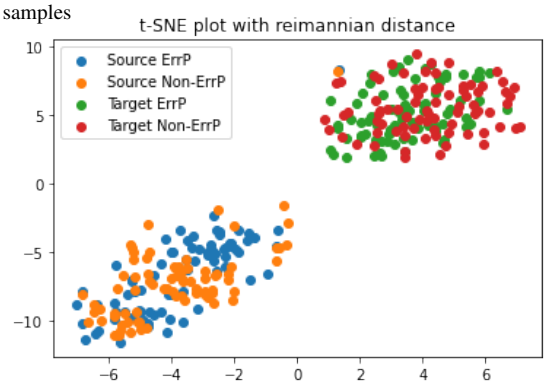
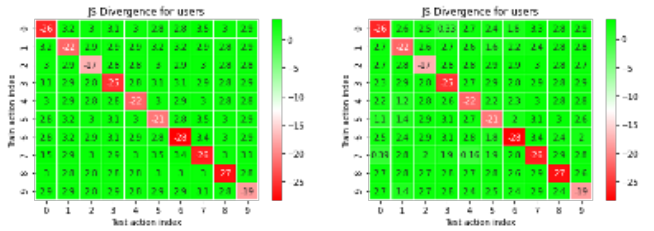


Fig. 4. t-SNE 2d-visualization of source and target distributions in a Riemannian manifold

of the matrix being 62.1%. While the generalization accuracy marginally improves, it is still poor. We use our diverse weak predictors obtained from each training user to boost our overall transfer accuracy using ensemble methods. We employ a simple ensemble method of pooling the classification probabilities from all predictors for a given test user and then using soft-voting to calculate the target label. After applying this ensemble method to our classifier, we see that our accuracy improves to 68.9%. The detailed numbers are shown in Table I. In the next subsection, we investigate the reasons behind the poor generalization across users and the means to address it.



(a) Jensen-Shannon Divergence between source and target samples for model 1 = 2.9

(b) Jensen-Shannon Divergence between source and target samples for model 2 = 2.45

Fig. 5. Jensen-Shannon Divergence for the two models

B. Model transfer for reducing divergence

In this subsection, we lay out our motivation for transferring specific components of our model pipeline from a training user to a test user. The chief source of achieving poor generalization accuracy on ErrP signals is the underlying disparity in the probability distribution the classifier anticipates and the probability distribution of the target data (test data). As we can see in Fig IV-A, the fundamental reason why we achieve poor generalization accuracy is the non-overlapping nature of the source distribution (the distribution that the classifier is trained on) and the target distribution (the distribution that the classifier is tested on). We measure this disparity between the two distributions using the Jensen-Shannon Divergence (JSD). We use JSD instead of Kullback-Leibler Divergence (KLD) as the latter requires the two distributions to be absolutely continuous w.r.t each other, an assumption that is not valid in our dataset. To address this, we use JSD, which is a modification of KLD, that mitigates this lack of overlap by creating a mean distribution as follows:

$$D_{JS}(A,B) = \frac{1}{2}D_{KL}(A||\frac{A+B}{2}) + \frac{1}{2}D_{KL}(B||\frac{A+B}{2}), \tag{3}$$

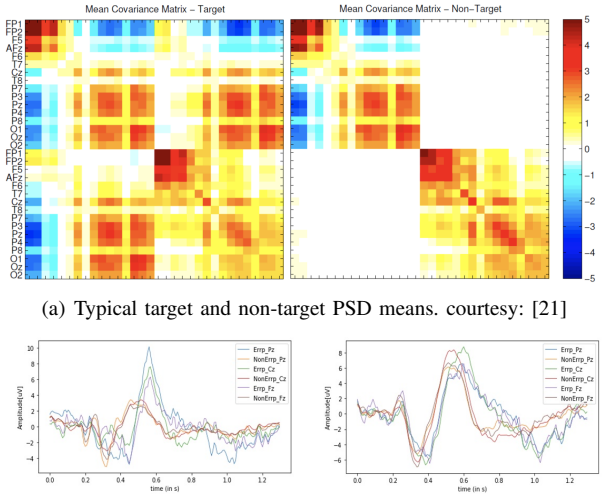
where A and B are our source and target distributions respectively. Since we do not have the probability distributions of our data, we calculate it from our data vectors using a kth nearest neighbor method proposed in [20]. To approximate the KL-divergence between two distributions A and B, we have:

$$D_{KL}(A||B) = \frac{d}{n}\Sigma_{i=1}^{n} log\frac{a_k(x_i)}{b_k(x_i)} + log\frac{m}{n-1}, \tag{4}$$

where $a_k(x_i)$ and $b_k(x_i)$ are the k^{th} nearest neighbors in A and B respectively from the point x_i, d is the total dimension of each vector, and m and n are the number of samples present in distributions A and B respectively. A higher value of JSD means more disparate distributions while a lower value indicates more similar distributions. We calculate JSD between our source and target datasets for two instances of our model. In our first instance, the class templates of the target user are generated by target user class labels (ref Fig 3(a)). In our second instant, the class templates are generated by the labels of the source user. We repeat this experiment for every pair of source (training) and target (test) users and plot the JSDs in a matrix for each instance. As seen in Fig 5, the average JSD for the instance where the target covariance matrices are generated by the source labels is lesser (mean JSD = 2.45) compared to the instance when target covariance matrices are generated by the target labels (mean JSD = 2.9). This suggests that the distribution generated by the source labels is closer to the source distribution (that the classifier is trained on for cross-user generalization) than what is generated using target labels. Thus, for generalization, we transfer the source xDAWN generator to the target model.

C. Classifier transfer with few shot learning

We also note that the xDAWN algorithm is sensitive to the average user signal template and that one user's signal data does not perform well on another user's data given the discrepancy among user templates. We can see this variation across non-target and target prototypes in different channels in Fig 6(b). This motivates us to incorporate some of the target user's signal data for spatial filtering and thereby adapt the model to inter-user variation. However, using the target user's signal data comes at a calibration overhead as generating supervised samples is time-consuming. Using a sufficient quantity of the target user's data will improve accuracy but will extensively burden the user with training overhead. On the other hand, prioritizing no training overhead will provide us with poor transfer learning accuracy. In this trade-off between reducing the training overhead for the user and achieving an accuracy reasonably close to the intra-user detection accuracy, we can find a sweet spot with few-shot learning, where we estimate the template of the target user using a "few" samples of the target user itself. We run our analysis for 3 different values of shots, namely N=5, 10, and 15 to span a varying spectrum of training overhead and cross-user accuracy. Additionally, we can see the mean target and mean non-target matrices as visualized in Fig 6(a) and make a few observations. Since the classifier classifies PSD covariance matrices, it does not explicitly store the signal templates of individual users but rather their covariance with the template waveform. This makes it a better candidate for generalization and transfer compared to signal templates of individual users. In figure 6(a), the two $n \times n$ matrix heatmaps denote the mean target and mean non-target covariance matrices. This is a typical target and non-target mean matrix observed for users and it does not show a lot of variation across users. Accordingly, a classifier transferred in a cross-user setting is expected to generalize well across users. Therefore, we reuse the training classifiers for evaluating this few-shot learning method.



(a) Typical target and non-target PSD means. courtesy: [21]

(b) The variation in average signal prototypes for 2 different users. The blue-green waveforms denote target prototypes and the red-orange waveforms denote the non-target prototypes

Fig. 6. Template differences and motivation for classifier transfer

We conducted the experiments associated with using $N = 5, 10, 15$ shots from each target class for template estimation

Method	Transfer Accuracy (%)	Ensemble accuracy (%)	Average test variance	Average training variance
Naïve TL	62.1	68.9	26.73	30.34
FSL N=5	59.04	61.9	14.35	30.31
FSL N=10	63.34	66.4	14.33	28.31
FSL N=15	65.84	68.4	13.89	30.98

TABLE I
ACCURACY NUMBERS FOR FEW-SHOT LEARNING WITH CLASSIFIER TRANSFER

Method	Transfer Accuracy (%)	Ensemble accuracy (%)	After K-shot supplementation	Average test variance	Average training variance
Naïve TL	60.9	68.9	-	25.9	30.7
FSL N=5	64.5	70.5	73.1	11.9	25.6
FSL N=10	66.4	72.1	74.6	15.2	26.7
FSL N=15	68.2	74.0	75.3	12.4	27.9

TABLE II
ACCURACY NUMBERS FOR FEW-SHOT LEARNING WITH ADAPTIVE CLASSIFIER AND SUPPLEMENTARY UNSUPERVISED SHOTS

while we transferred the classifier as-is from the training data. The results are summarized in Table I. Despite an increase in non-ensemble transfer accuracy compared to the naive transfer learning methodology (implying better cross-user generalization for few-shot learning), the ensemble accuracy is still lower. Looking at the average test variance for few-shot learning, we can infer that the current methodology has almost half the variance than that of the naive transfer learning method, which is because it is affected by a higher bias. Given this high bias, our ensemble classifier does not have the required diversity to make a robust decision and hence the ensemble accuracy of the classifier is poor despite having superior generalization transfer accuracy.

D. Diversifying training samples for reducing classifier bias

The few-shot learning methodology suffers from high bias since we are using very few samples to generate the mean prototype of the target user. This makes our prototype heavily biased toward the initial 5, 10, or 15 samples we choose from the target classes. To get rid of this bias, we need to diversify our starting samples such that prototype calculation is robust to variations in the initial selection of few-shot samples. For this, we include the training user's signal samples in the few-shot prototype estimation step and also include the initial few-shot samples for training the classifier making the classifier "adapt" to a target user's samples. This makes the classifier robust to changes in the initial few-shot sampling, leading to a more powerful classifier. We present the accuracy numbers for this in Table II where we see that not only does the generalization accuracy increase but it is accompanied by an increase in the ensemble accuracy along with a reduction in the average test variance. This signifies that after adapting the classifier with N-shot samples, there is less variance in the prediction from multiple users.

E. Unsupervised Supplementation using confidence intervals

The results suggest that as we increase the number of shots in the few-shot learning paradigm, we can get closer and closer to achieving the same accuracy for cross-user transfer learning as we get for the standalone intra-user machine learning method. However, each additional shot during training implies increased time spent on user calibration. In our experiments, for the 1.5s variant of the maze game, it took about 136 seconds to get 17 ErrP and 67 non-ErrP samples. The average cost of obtaining 1 ErrP sample (which is the limiting step in our experiment) was 8 seconds. So a 10-shot learning paradigm requires about a minute's worth of user time for calibration. If we can increase the number of shots without burdening the user with additional calibration time, we can achieve both, a low calibration time, and a reasonably high detection accuracy. We can realize this by making two observations, namely, i) Even with weak classifiers, we can reasonably assume within a specific margin of error, that we are making some correct predictions without validating them, and ii) Given that our ensemble classifier is robust to variations in the N-shot samples, we can afford to introduce some stochasticity in the correctness of some of the samples. We do this at the end of the first round of classification by selecting K unsupervised samples from both classes with high output probability. Once we select these samples, we assume that their predicted label is their ground truth label and bunch them with the initial N-shot samples used for prototype estimation. Then we repeat the algorithm with our $N + K$ shots and get the ensemble accuracy. We can accomplish this by either batch learning or continuous online learning. In our experiments, we worked with batch learning but it can easily be extended to online learning where the algorithm chooses the next best sample in sequential order and updates itself to become a better predictor.

V. RESULTS AND DISCUSSION

The final results are shown in Table II. We can see that for 10-shot learning (FSL N=10) after K-shot supplementation, we achieve an average cross-user accuracy of 74.6% which approaches the average accuracy of a standalone intra-user model (76.6%) while only requiring 20 total samples from the target user (as opposed to an average of 450 samples needed for the standalone model) while also reducing the overhead on the user by an order of magnitude (as shown in Fig 7).

In our experiments, the optimum value for K was observed to be 10. We obtain this value by using a range of values for K from 0 to 15. We compare this algorithm with EEG-Net for cross-user ErrP detection [14], the ensemble and 1-bagged model of cross-user xDAWN algorithm [11], and the traditional intra-user accuracy that is obtained when training

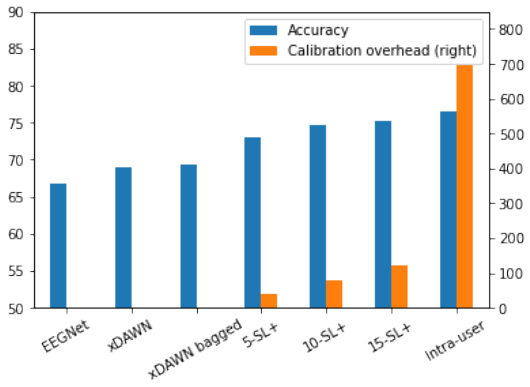


Fig. 7. Comparison with current state-of-the-art models w.r.t accuracy and calibration overhead (measured in required training samples)

and testing on the same user (using the same state-of-the-art xDAWN algorithm). For EEGNet, we used the implementation described in [14] and for every test subject, considered the signals of the remaining 9 subjects as training data. For the 1-bagged model of xDAWN, we bagged the remaining 9 subjects for each test subject and trained the model on them, while for the ensemble model, we used soft-voting for 9 classifiers trained on 9 individual training subjects (the same methodology we use for our few-shot learning classifier). The same parameters for xDAWN as the ones mentioned in [11] were used with the exception of the number of initial electrodes which were 10 in our experiments as opposed to 56 in [11] and [14]. The comparison is shown in Fig 7. We can see 3 N-shot learning approaches (5-SL+, 10-SL+, and 15-SL+ with the '+' denoting unsupervised augmentation) which outperform the existing cross-user transfer learning approaches and approaches the average accuracy for the standalone intra-user model which requires lengthy amounts of time for user calibration (about 12-13 minutes in our experiments) compared to N-shot learning approaches which require about 40, 80, and 120 seconds of training overhead respectively. While EEGNet and xDAWN do not require any calibration they do not provide a high enough accuracy compared to our model. The blue bars showcase the accuracy that is achieved by a method (higher is better) and the orange bar shows the associated user samples required for user calibration as an overhead (lower is better). So in the trade-off between user convenience and higher accuracy, the presented algorithm hits the sweet spot of reducing the calibration time by an order of magnitude while also approaching the standard accuracy of an intra-user classification model.

VI. Conclusion

In this work, we present a few-shot transfer learning approach with unsupervised K-shot supplementation to generalize ErrP brain signals across human subjects which has the advantage of approaching the intra-user ErrP decoding accuracy while using only a small fraction of the samples needed for a standalone model. We provide comparisons of this approach with existing state-of-the-art statistical as well as deep learning transfer learning approaches for ErrP signals.

References

[1] Tao Chen, Haiyun Huang, Jiahui Pan, and Yuanqing Li. An eeg-based brain-computer interface for automatic sleep stage classification. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1988–1991. IEEE, 2018.

[2] Katie Crowley, Aidan Sliney, Ian Pitt, and Dave Murphy. Evaluating a brain-computer interface to categorise human emotional response. In *2010 10th IEEE international conference on advanced learning technologies*, pages 276–278. IEEE, 2010.

[3] E. Gupta, M. Agarwal, and R. Sivakumar. Blink to get in: Biometric authentication for mobile devices using eeg signals. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020.

[4] Duo Xu, Mohit Agarwal, Ekansh Gupta, Faramarz Fekri, and Raghupathy Sivakumar. Accelerating reinforcement learning using eeg-based implicit human feedback. *Neurocomputing*, 460:139–153, 10 2021.

[5] Paolo Zanini, Marco Congedo, Christian Jutten, Salem Said, and Yannick Berthoumieu. Transfer learning: A riemannian geometry framework with applications to brain-computer interfaces. *IEEE Transactions on Biomedical Engineering*, 65:1107–1116, 5 2018.

[6] E Gupta, C. Y. Chen, and R Sivakumar. Hyper-accelerated learning for brain-computer interfaces via partial target-aware optimal transport. In *Proceedings of the 2nd Workshop on Smart Wearable Systems and Applications*, SmartWear '23, 2023.

[7] M. Falkenstein, J. Hohnsbein, J. Hoormann, and L. Blanke. Effects of crossmodal divided attention on late erp components. ii. error processing in choice reaction tasks. *Electroencephalography and Clinical Neurophysiology*, 78:447–455, 6 1991.

[8] Reuderink B, Farquhar J, Poel M, and Nijholt A. A subject-independent brain-computer interface based on smoothed, second-order baselining. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, 2011:4600–4604, 2011.

[9] S Vaid, P Singh, and C Kaur. Eeg signal analysis for bci interface: A review. *International Conference on Advanced Computing and Communication Technologies, ACCT*, 2015-April:143–147, 4 2015.

[10] Clay B. Holroyd and Michael G. H. Coles. The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity. *Psychological review*, 109 4:679–709, 2002.

[11] Alexandre Barachant and Marco Congedo. A plug and play p300 bci using information geometry. *""*, 8 2014.

[12] Ekansh Gupta and Raghupathy Sivakumar. Wisdom of the crowd: Using multi-human few-shot learning to improve cross- user generalization for error potentials in bci systems. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022.

[13] Sunhee Hwang, Kibeom Hong, Guiyoung Son, and Hyeran Byun. Ezsl-gan: Eeg-based zero-shot learning approach using a generative adversarial network. *7th International Winter Conference on Brain-Computer Interface, BCI 2019*, 2 2019.

[14] 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*, 15:056013, 7 2018.

[15] Jost Tobias Springenberg et al Robin Tibor Schirrmeister. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human Brain Mapping*, 38:5391–5420, 11 2017.

[16] B Rivet, A Souloumiac, V Attina, and G Gibert. xdawn algorithm to enhance evoked potentials: application to brain–computer interface. *IEEE Transactions on Biomedical Engineering*, 56(8):2035–2043, 2009.

[17] A Barachant and A Bonnet. Channel selection procedure using riemannian distance for bci applications. In *2011 5th International IEEE/EMBS Conference on Neural Engineering*, pages 348–351. IEEE, 2011.

[18] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of The Royal Statistical Society Series B-statistical Methodology*, Apr 2005.

[19] P. Thomas Fletcher and Sarang Joshi. Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. *Lecture Notes in Computer Science*, 3117:87–98, 2004.

[20] Fernando Pérez-Cruz. Kullback-leibler divergence estimation of continuous distributions. In *2008 IEEE international symposium on information theory*, pages 1666–1670. IEEE, 2008.

[21] Marco Congedo, Alexandre Barachant, and Anton Andreev. A new generation of brain-computer interface based on riemannian geometry. *""*, 10 2013.