Mental Workload Classification using Style Transfer Mapping on fNIRS data

Irene Chang[1] Michael Hughes[2]

[12]Tufts University

December 2022

Abstract

Brain signals, such as functional near-infrared spectroscopy (fNIRS) and electroencephalogram (EEG) vary significantly from one person to another, thus collect sufficient data and building customized brain-computer interfaces for every user is a necessary, yet laborious thing to do. These variations have given rise to many studies on transfer learning models that can be trained on a sample of individuals to analyze the signals of a new subject. As part of the ongoing expansion of this line of work, this study seeks to evaluate a style transfer mapping (STM) method on a dataset of fNIRS signals in a mental workload intensity level classification task. This technique has previously been showed to produced improved precision on a similar task of recognizing emotions using EEG signals, and we want to examine its effectiveness can be extended to other types of classification tasks on brain signal data. This study inherits the fNIRS dataset and preprocessing pipeline from Huang *et al.* and Wang *et al.* (2021 [1]). We compare the fine-tuned STM model performance to our baseline metrics on the test data when being trained with different numbers of source subjects. We show that the accuracy of STM varies greatly on the target test data, which is likely due to the shift in distributions between the train and test set of the target data. In addition, we also assess the performance of a modified version of the original model and show that the algorithm with reduced complexity can perform just as well.

1 Introduction

1.1 Motivation

BCIs not only are used to assist physically challenged users, but have recently also been pivoted towards gaining an understanding of mental statuses of a wider population while performing daily tasks. However, the limited amount of data and the high variation between individuals' signals are some major obstacles in the development of efficient and scalable BCIs. Huang *et al.* and Wang *et al.* (2021) addressed these problems with the proposal of a standardized evaluation protocol for training and validating the machine learning model. They employed this protocol to compare the performance of four different classifiers (Random Forest, Logistic Regression, Deep Convolutional Network and EEGNet) on mental workload intensity classification task under subject-specific (trained and evaluated on a single subject) and generic (trained on a sample of subjects and evaluated on a new subject) settings. The generic models show a comparatively better accuracy on the test data compared to one that learns directly from limited target subject data, demonstrating potentials in this research direction. Nevertheless, the model still suffers from the cross-subject noises caused by differences in cortical structures and other physiological characteristics. These issues are referred to as covariate shifts in transfer learning literature, in which a model learning from the training data, referred to as source domain, performs poorly on the test set coming from a different domain, i.e features having different distributions from those in the source domain.

With that in mind, we aim to look at the performance of a newly developed domain adaptation technique using the same data modeling pipeline of this study to see if it can mitigate the covariate shift problem in fNIRS dataset. The method is based upon style transfer, an optimization technique that has been researched in the field of deep learning on image data. A style transfer task on image data involves taking a content image and a style reference image and to produce a blended image that looks like the content image, but has the style of the style reference image [2]. Li *et al.* (2020 [3]) developed a multi-source style transfer mapping based on this principle for the transfer learning task of using EEG signals to categorize emotions. In this method, the model attempts to "blend" the signal data of a new subject with the signal data of a source subject. The goal is to reuse the classifier already trained on the source dataset to classify the new data effectively. In doing so, it minimizes the distance between the new and the old subjects, hence decreasing the effect of covariate shift in the data. The algorithm leverages the results of multiple transfers to different source subject to make predictions on a never-before-seen subject. The proposed model proved to be successful on the EEG signal data compared to an ensemble of subject-specific classifiers.

Recognizing the similarities between this study and our study in terms of the data format (brain signal data) and the task (classification of a mental characteristics), we want to examine if the result carries over to our experiments. If successful, this methodology can be deployed to produce real time predictions as the human users are completing a particular task, which subsequently contributes to an area of BCIs regarding the use of fNIRS signals to inform about mental workload intensity.

1.2 Datasets

We utilized the published dataset in the study of Huang *et al.* and Wang *et al.* (2021 [1]). There are 4 labels $n \in \{0, 1, 2, 3\}$ representing 4 mental workload levels in increasing order of intensity. The subjects in the study participated in different tasks that induced different levels of mental workload and their fNIRS signals, consisting of 8 measurements, were collected. Each task happened over a block of time, which contains a predefined number of timesteps, following by a break. During each block, data is recorded in batches of timesteps, referred to as windows. Windows contain overlaps: consecutive windows are collected by sliding over by 3 timesteps until reaching the end of the block. The algorithm doesn't consume sequential data, so for the purpose of this study, we transform the data by taking the average of the measurements within each window. In this way, each window corresponds to an entry in the input dataset.

Hence, the size of the finalized input dataset for each subject (both source and target) is of size `num_windows` $\times 9$. The features of the dataset consists of 8 fNIRS measurements and a label of the corresponding mental workload level.

1.3 Research Hypotheses

Through this study, we aim to test the following hypotheses:

1. **STM tackles the covariate shift problem and produces better predictions on a new subject**: We want to focus on the effect of STM on a never-before-seen subject. STM maps the target data to the space of each source and hence takes advantage of the classifiers built on each source domains. This implies that the target domain classification accuracy relies on the precision of these individual classifiers. We want to look into situations where STM performs well and situation where it does poorly to better understand the overall fitness of this method on our specific task.

2. **The current algorithm can be simplified while still delivering comparable performance**: The transfer learning algorithm is made up of multiple steps that involve different statistical machine learning models. We argue that the support vector machine (SVM) step and the best source selection (BSS) step in the pipeline do not add significant benefits to the outcome of the model and thus may be removed to reduce the complexity of the model.

1.4 Related Work

Recently, there has been a lot of progress on the research into domain adaptation techniques in neuroscience settings using different kinds of brain signals data, contributing to the growing development of BCI. However, models on electroencephalogram (EEG) signals are more extensively studied than fNIRS. Wu *et al.* (2022 [4]) compiled state-of-the-art transfer learning methodologies developed since 2016 for EEG-based tasks, ranging from statistical machine learning to deep neural networks. Our study aims to contribute to the expansion of these successful techniques onto fNIRS-based models.

2 Methods

Summary of the original algorithm:

Li *et al.* (2020 [3]) developed cross-subject emotion classifier under both supervised and semisupervised settings. In this study, we focus on the supervised setting. We refer to the data points in the source as the “destination”, and data points in the target as the “origin”. The optimization function is defined as:

$$\min_{A\in\mathbb{R}^{m\times m}, b\in\mathbb{R}^m} \sum_{i=1}^{n} ||Ao_i + b - d_i||_2^2 + \beta||A - I||_F^2 + \gamma||b||_2^2 \tag{1}$$

where n refers to the number of data points in the set, m refers to the feature dimensionality; o_i's are the origin points and d_i's are the destination points; A, b are parameters of the mapping we are learning to map the patterns of the “origin” to the “destination”. The first component of (1) is the square error. Since this is a supervised transfer setting, the parameter f_i's, which represent the transfer confidence and are attached to the square error as the weighting factor in the original algorithm, all carry the value of 1 and hence not shown above. The second component is one regularization term to prevent A from being too far away from the identity matrix I, and the third component is another regularization term to make sure b is not far from 0. The regularization hyper-parameters β and γ strike a balance between overtransfer (if their values are too small) and nontransfer (if their values are too large), and are chosen during the tuning procedure. $|| \cdot ||_F^2, || \cdot ||_2$ is the Frobenius norm and the L_2-norm of the matrix, respectively.

They first trained a classifier in each of the sources involved in the training and selected the sources that produced the highest classification accuracy on the labeled target set used for training. Then they built an SVM model on each source data and removed the support vectors, which are theoretically close to the decision boundaries between the classes and hence unreliable. They performed k-means clustering within each class of the remaining data points to obtain k prototypes per class, which are then used to compute the style transfer mapping. Formulation (1) is a convex quadratic programming problem with the following closed-form solution.

$$A = QP^{-1}, b = \hat{d} - A\hat{o} \tag{2}$$

$$Q = \sum_{i=1}^{n} d_i o_i^T - \left(\frac{1}{n+\gamma}\right) \hat{d}\hat{o}^T + \beta I \tag{3}$$

$$P = \sum_{i=1}^{n} o_i o_i^T - \left(\frac{1}{n+\gamma}\right) \hat{o}\hat{o}^T + \beta I \tag{4}$$

$$\hat{o} = \sum_{i=1}^{n} o_i, \hat{d} = \sum_{i=1}^{n} d_i \tag{5}$$

To predict a new sample of data of the target subject, they first mapped the data to each of the sources and then used the classifier previously trained in the corresponding source to produce a prediction. All the predictions were then weighted averaged, where the weights were determined by the accuracies of the source classifiers obtained at the first step of the algorithm. We replicate this model for our study, choosing random forest as the source classifiers for our multi-class classification task.

This style transfer mapping algorithm is customized for each new target subject, i.e each model learns from multiple sources to make prediction on a single new subject. This is because the method uses the available labeled target set during training to compute the mappings. Thus, our evaluation method is also made with respect to each target subject separately.

Preprocessing data

We follow the same data preprocessing procedure as laid out in Huang *et al.* and Wang *et al.* (2021 [1]). All data from each source is used for learning the mappings, whereas data from the target subject is divided into 3 splits: the training split, the validation split (for hyperparameter tuning), and the test split (to evaluate the performance of the model on never-before-seen data), such that there is no overlapping window across these splits.

Baseline metrics

In order to evaluate our model, we will compare the results on the target test set against several baseline models:

- The first baseline metric is the accuracy score of a supervised model on the target test set. A random forest model is trained on all the target subject data used for training, and evaluated on the target test set. We want the accuracy of our method to be as close to this accuracy as possible.

- The second metric is the accuracy from a generic random forest model trained on all source subjects' data. This shows us how the model performs as covariate shift is present in our data. Ideally, our method should achieve a better accuracy than this model.

- The third metric is the weighted accuracy of an ensemble model of individual random forest classifiers from all source subjects. We also want our model to exceed this accuracy score.

Hyperparameter tuning

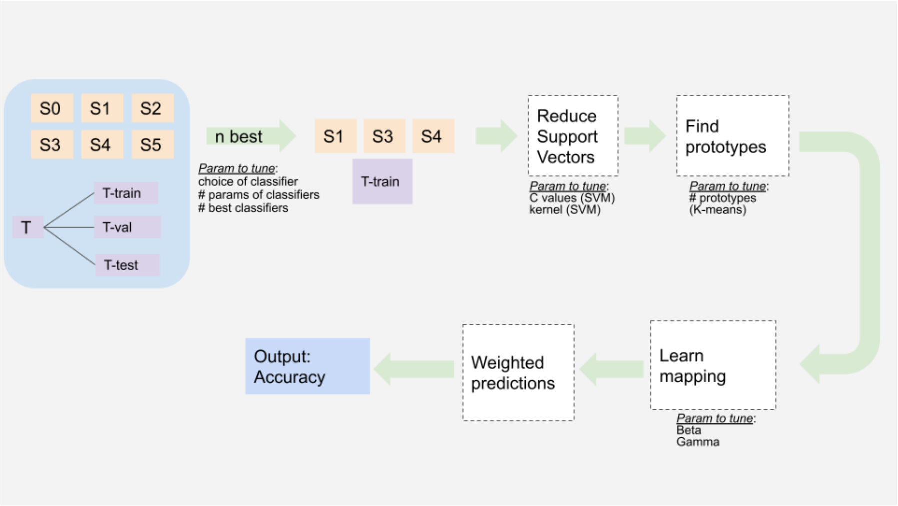

Figure 1: The full training pipeline, as outlined in [3]

The hyperparameters to be tuned consist of the hyperparamters in individual statistical models involved in the pipeline as illustrated in Fig. 1. These include:

- The number of best source classifiers selected for STM learning.
- `C_list` and `kernel_list`: hyperparameters for the SVM step for each domain.
- `k_list`: the number of prototypes in the K-means step for each of the source domains
- β and γ: the regularization terms in learning STM.
- The choice of source classifiers: this includes but is not limited to logistic regression, random forest, a neural network classifier.
- Hyper-parameter tuning for source classifiers: this tuning process can be designed to be on the basis of each source separately, or on the basis of shared parameters across all source classifiers.

The use of SVM and best sources selection

In the original algorithm, the author makes use of SVM (which removes the data points near the classification boundaries of the classes) and BSS (which selects the best sources based on their original classification accuracy scores on the target training set). For the latter, the author argued that the purpose of BSS is to avoid negative transfer, i.e unrelated sources leading to poor performance on target subject (Li, 2020[3]). However, we did not find a conceptual basis that shows choosing only the best sources based on the initial classification results will lead to improved accuracy in the study. For the SVM steps, although the removal of the data points near the boundaries means the transferred target data can be classified with high reliability on the space of each source, it can also cause loss of important information. Since the source classifiers are learned prior to the SVM step, the amount of data left after the removal might not be representative of the patterns learned by the source classifiers, and hence their predictions on the transferred target data might not be accurate. This limitation is also pointed out by the author to be a subject of further experimentation. In the next section, we also provide ad-hoc comparison of STM models with and without these two steps to see whether these steps help us achieve a better accuracy and determine if they should be included in the model.

Models built

Putting everything together, to test for hypothesis 1, for each target, we build a model on all the source data and a fraction of the target data. Following the reasoning above, in this experiment, we eliminate two steps in the original algorithm, the SVM step and best source classifier selection step. We carry out parameter tuning at the same time using a random search approach of all hyper-parameters and select the model with the highest accuracy on the validation set. Finally, the estimator will be applied (without retraining) on the test set of the target data. Test accuracy scores will be summarized and analyzed. This yields a total of `num_target_subjects` models for the STM method.

For each target, we also compute the 3 mentioned baseline metrics to compare against the test accuracies from STM. Thus, we obtain 4 statistics for each target subject, and a total of `num_target_subjects` × 4 models.

Investigating hypotheses 2 comes in 3 parts. First, we will test for the effectiveness of each of the two steps in question separately. For each target subject, control the model to not include BSS. A model using SVM to eliminate near-boundary data points and a model without using SVM are both tuned to get the best set of parameters. These estimators are then applied on the test set and can be compared by looking at the differences in the accuracy scores. Likewise, we can fix the models to not include the SVM, and test the effect of having BSS step on the results. Second, we also build the full model (SVM + BSS) and compare its test accuracy to the model that does not include these two steps. In total, we build 4 models: Base (without SVM or BSS); SVM-only; BSS-only, SVM + BSS.

3 Experiments

To measure the benefits of this algorithm on our 4-class classification task (labels are balanced in training, validation and test sets: 25% accuracy represents a random guess), we experiment the algorithm on a subset of 14 subjects, 10 of which are arbitrarily treated as source subjects, and the other 4 are treated as target subjects. We thus have 4 different test sets. We combine training and fine-tuning the models using random grid search for `n_iter =50`. We then compare the performance of fine-tuned STM on each of the 4 target test sets across varying number of source subjects involved in training, with the minimum of 1 source, and maximum of 10 sources.

In a similar process as in Huang and Wang (2021[1]), to capture the uncertainty, we draw 1000 bootstrap resamples of the test set with replacement, each of size 500. We then calculate the 95% confidence intervals from all samples. The test set accuracy for each target computed above is reported along with the corresponding confidence interval, which tells us how the test set accuracy for a given target subject might vary across test sets drawn from the same empirical distribution (Huang and Wang, 2021[1]).

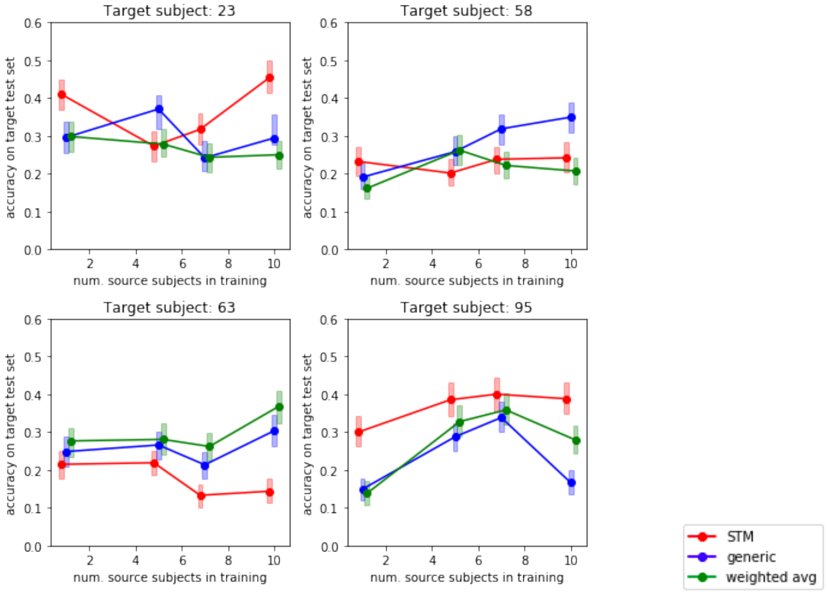

Figure 2: **Classifier accuracy on the target test set as more subjects are added**. For each target subject, the y-axis reports the accuracy of the fine-tuned model in predicting target test set. The dark points represent the test set accuracy and the intervals represent the 2.5-97.5 percentile of accuracy score across 5000 bootstrap resamples of the test set. For each target, the models in examination are trained with 1, 5, 7, 10 source subjects. *generic* models corresponds to our second baseline metric and *weighted avg* corresponds to our third baseline metric

Table 1: Accuracy (%) comparison (higher means better) on target subject 23, 58, 63 and 95 between STM and other baseline models after 1000 bootstrap resamples (mean ± std), examined over different number of source subjects. Bold entries indicate the highest accuracy for the models apart from target supervised. *Target supervised* refers to the first baseline metric, *generic* models corresponds to our second baseline metric and *weighted avg* corresponds to our third baseline metric

	1 source				5 sources			
	sub23	sub58	sub63	sub95	sub23	sub58	sub63	sub95
Generic	0.30 ± 0.02	0.19 ± 0.02	0.25 ± 0.02	0.15 ± 0.02	**0.36 ± 0.02**	**0.26 ± 0.02**	0.27 ± 0.02	0.29 ± 0.02
Weighted Average	0.30 ± 0.02	0.16 ± 0.02	**0.28 ± 0.02**	0.14 ± 0.02	0.28 ± 0.02	**0.26 ± 0.02**	**0.28 ± 0.02**	0.33 ± 0.02
STM	**0.41 ± 0.02**	**0.23 ± 0.02**	0.22 ± 0.02	**0.30 ± 0.02**	0.27 ± 0.02	0.20 ± 0.02	0.22 ± 0.02	**0.38 ± 0.02**
Target Supervised	0.39± 0.02	0.27 ± 0.02	0.30 ± 0.02	0.27 ± 0.02	0.39± 0.02	0.27 ± 0.02	0.30 ± 0.02	0.27 ± 0.02

	7 sources				10 sources			
	sub23	sub58	sub63	sub95	sub23	sub58	sub63	sub95
Generic	0.25 ± 0.02	**0.32 ± 0.02**	0.21 ± 0.02	0.34 ± 0.02	0.31 ± 0.02	**0.350 ± 0.02**	0.30 ± 0.02	0.17 ± 0.02
Weighted Average	0.24 ± 0.02	0.22 ± 0.02	**0.26 ± 0.02**	0.36 ± 0.02	0.25 ± 0.02	0.21 ± 0.02	**0.37 ± 0.02**	0.28 ± 0.02
STM	**0.32 ± 0.02**	0.24 ± 0.02	0.13 ± 0.02	**0.40 ± 0.02**	**0.46 ± 0.02**	0.24 ± 0.02	0.14 ± 0.02	**0.39 ± 0.02**
Target Supervised	0.39± 0.02	0.27 ± 0.02	0.30 ± 0.02	0.27 ± 0.02	0.39± 0.02	0.27 ± 0.02	0.30 ± 0.02	0.27 ± 0.02

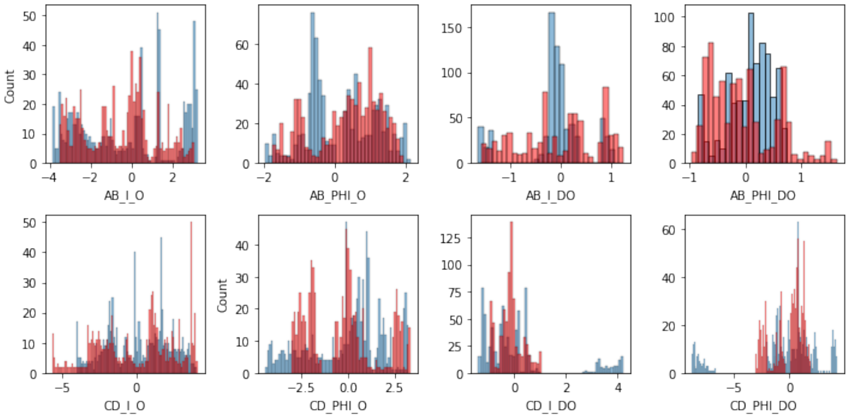

Figure 3: **Marginal distribution of the features in the training and the test set of subject 63**. Each subplot is a histogram of a feature's distribution. Blue bars represent instances in the training set and red bars represent instances in the test set.

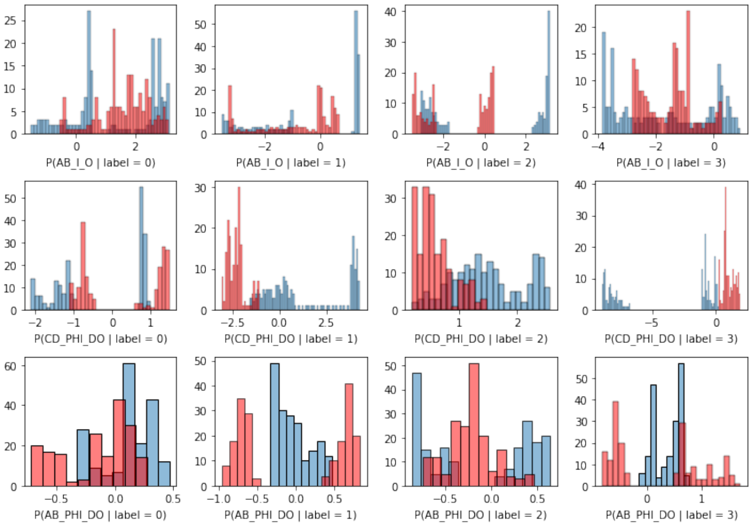

Figure 4: **Class-conditional marginal distributions of each feature of target subject 63**. Training set and test set are depicted in blue and red respectively.

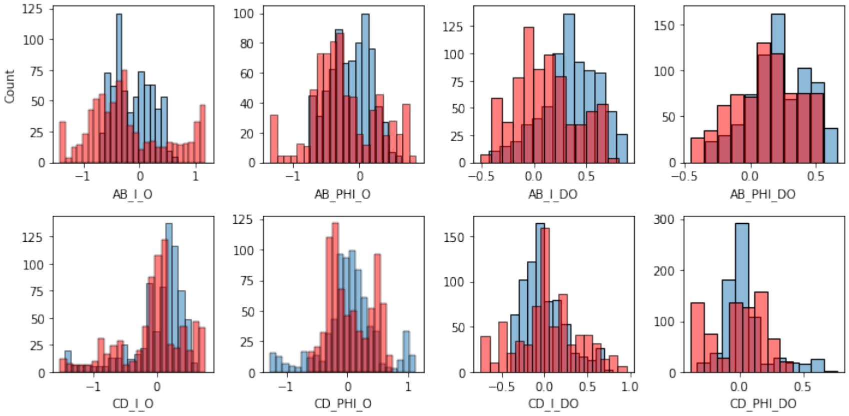

Figure 5: **Marginal distribution of the features in the training and the test set of subject 95**. Each subplot is a histogram of a feature's distribution. Blue bars represent instances in the training set and red bars represent instances in the test set.

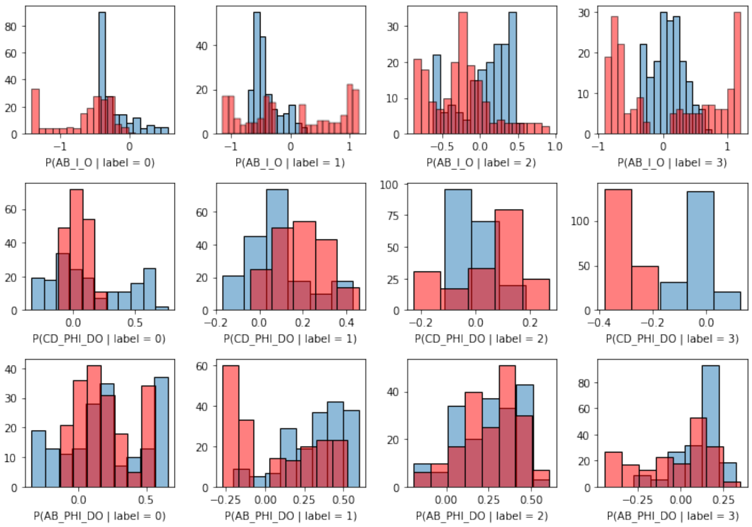

Figure 6: **Class-conditional marginal distributions of each feature of target subject 95**. Training set and test set are depicted in blue and red respectively.

In order to test the effect that the SVM step and BSS step has on the model, we also make use of bootstrap sampling in a similar procedure. We build the four listed models using all 10 source subjects. In SVM-only and SVM+BSS models, grid search also includes tuning the regularization term, C, for each of the sources. In BSS and SVM+BSS model, we select a fixed $n = 4$ best sources. The accuracy scores of the base method (model without SVM and BSS) is subtracted from the accuracy scores of each method separately, and the distribution of these differences across 1000 bootstraps are plotted.

Below we highlight some takeaways from our experiments.

Finding 1: Including SVM and BSS does not improve accuracy. Fig. 7 shows that none of 4 target subjects significantly benefits from the addition of SVM and/or BSS step into the training pipeline. The intervals of the differences either contain 0 (implying the base method and the enhanced methods produce the same accuracy), or are in the negative range (implying that the enhanced methods produce less accurate predictions than the base method). This shows that the assumption that Li *et al.* (2020 [3]) made with regard to negative transfer when we include all the sources does not hold in our case.

Finding 2: Target test set accuracy scores vary across different number of source subjects. In Fig. 2, we show that the performance of STM does not improve as more source subjects are included. Overall, the performance of STM fluctuates when being trained with different numbers of source subjects. However, since we are using a small number of iterations for random grid search (50),

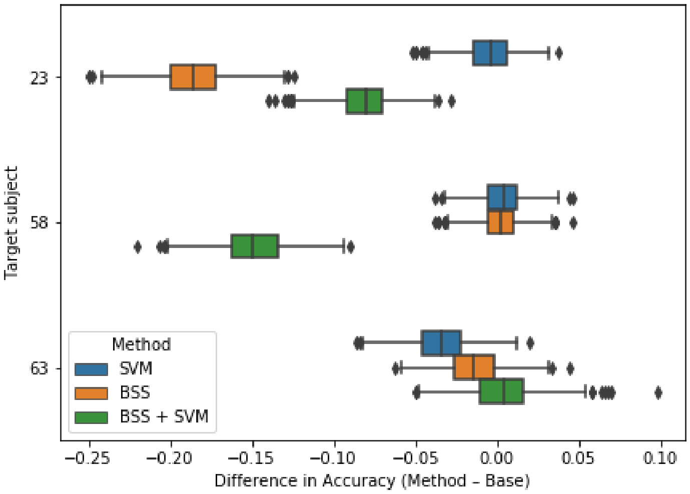

Figure 7: **Compare the accuracy of STM with and without the additional steps**. Tuned models from all methods are tested on 1000 bootstrap resamples of the test set of target subjects 23, 58, 63. The three methods in question are support vector machine (SVM) only, best source selection (BSS) only, and a combination of SVM and BSS.

compared to the total number of possible combinations, there is a possibility that we have not found the optimal model.

Finding 3: For some subjects, STM outperforms the full supervised learning on the target dataset, while others perform worse than the generic models. According to our results in Table 1, almost all STM models on subject 23 and subject 95 make better predictions on the test set, compared to the baseline models, including the supervised models that are trained directly on the target subjects. However, this is not the case for target subject 63 and subject 58, whose performances are worse than the model without domain adaptation. In fact, only subject 95 showed consistently better classification accuracy scores across all trials.

We also observe that the scores of both subject-specific and cross-subject paradigms are generally low (< 50%), with most models only slightly exceeding the accuracy of random guessing on this balanced dataset (25%). These raised a question of the dependence of STM on the overall fitness of statistical classifiers on the dataset of a specific subject, for which we provide a further discussion of this problem in the next section.

4 Discussion

4.1 Feature distributions and fitness of STM

As showed in the previous section, the performance of STM varies a lot across different target subjects. Subject 95 is an example of when STM produces better classification results compared to an ensemble, a generic, and even the target's own supervised model. Subject 63, on the other hand, is the case when using STM negatively impacts the accuracy of the model. In attempting to understand this discrepancy, we take a further look into the distributions of the measurements in the training and test splits of both subjects. Fig. 3, 5 show the marginal distributions of the features and Fig. 4, 6 shows the class-conditioned marginal distributions of the features.

Overall, we see that the marginal distributions of the features between the train and the test splits are better aligned for subject 95 than for subject 63. For subject 63, features such as AB_I_O and AB_I_DO have different modes and variances between the train and test splits, whereas this disparity is smaller in the dataset of subject 95. However, for subject 95, there is also a slight mismatch between the train and test distributions of some features, such as with AB_PHI_O and AB_I_DO.

Class-conditioned marginal distributions tell us how the instances of the same label in a dataset are distributed. Looking at the class-conditional distributions of both subjects, we also see the same trends as the marginal distributions. On the class level, the poor alignment between the test and train splits appear even more obvious. In other words, there appears to be covariate shifts in their feature distributions between the train and test splits of these subjects. This makes sense in the context of our data preprocessing pipeline, since we split the training and testing splits such that they come from different periods of the experiments.

Since STM works on the assumption that instances of the target will come from the same empirical distributions (Li 2020[3]), this can partly explain for the poor performance of this model on our test dataset.

4.2 Future direction

- As more source subjects are included in the model, the number of hyperparameters for tuning grows exponentially. With more computing power, the current algorithm can be tested more rigorous on a wider range of values for each parameter, as well as altering the choice of the source classifiers (with considerations for other low complexity deep learning networks).
- Looking forward, we anticipate that the current STM algorithm can be enhanced to take into account the sensitivity of STM to the intra-variation of the target subject set as analyzed above. Alternatively, other domain adaptation-based methods as summarized in Wu *et al.* (2022) [4] can be experimented on this fNIRS dataset to see which methods excel on this task,

and why they do.

References

[1] Zhe Huang and Liang Wang. *The Tufts fNIRS to Mental Workload Dataset: Toward Brain-Computer Interfaces that Generalize*. 2021. URL: `https://openreview.net/forum?id=QzNHE7QHhut`.

[2] *Neural style transfer*. URL: `https://www.tensorflow.org/tutorials/generative/style_transfer`.

[3] Jinpeng Li et al. "Multisource Transfer Learning for Cross-Subject EEG Emotion Recognition". In: *IEEE Transactions on Cybernetics* 50.7 (2020), pp. 3281–3293. DOI: `10.1109/TCYB.2019.2904052`.

[4] Xu Y Wu D and Lu B.-L. "Transfer learning for EEG-based braincomputer interfaces: A review of progress made since 2016". In: *IEEE Transactions on Cognitive and Develop- mental Systems* 14 (2022), pp. 4–19.