Domain Adaptation and Transfer Learning methods enhance Deep Learning Models used in Inner Speech Based Brain Computer Interfaces

Luciano Ivan Zablocki[1,#], Agustín Nicolás Mendoza[1,#], Nicolás Nieto[2,3,*]

[1] Facultad de Ingeniería y Ciencias Hídricas, FICH, UNL # Equally contributing authors.
[2] Instituto de Matemática Aplicada del Litoral, IMAL, UNL-CONICET.
[3] Instituto de Investigación en Señales, Sistemas e Inteligencia Computacional, sinc(i), UNL-CONICET *nnieto@sinc.unl.edu.ar.

Abstract. Brain-Computer Interfaces are useful devices that can partially restore communication from severely compromised patients. Although advances in deep learning have significantly improved brain pattern recognition, a large amount of data is required for training these deep architectures. In recent years, the inner speech paradigm has drawn much attention, as it can potentially allow natural control of different devices. However, as of the date of this publication, there is only a small amount of data available in this paradigm. In this work we show that it is possible, through transfer learning and domain adaptation methods, to make the most of the scarce data, enhancing the training process of a deep learning architecture used in brain-computer interfaces.

Keywords: Deep Learning · Domain Adaptation · Transfer Learning · Convolutional Neural Network

1 Introduction

Brain-Computer Interfaces (BCIs) are useful tools that allow users to control external devices only by using their cerebral activity [19]. These technologies are fundamental for patients who suffer from strokes, amyotrophic lateral sclerosis, and other accidents that may interrupt the normal communication between the brain and the muscles. By means of a BCI, these patients can, at least, partially restore their capability to communicate and interact with the environment, significantly improving their life quality [16].

Surface electroencephalography (EEG) is widely used for measuring brain activity in a BCI, as it is a standard and noninvasive technique [11]. EEG provides signals with good time resolution, allowing real-time applications, but with a low signal-noise ratio and a high inter and intra-subject variability [12]. BCIs can be commanded by different paradigms. One of the existing BCI paradigms is inner speech, which refers to the mental process of imagining one's voice and allowing a more natural way for controlling different devices, just by thinking of a specific command [12].

Received January 2023 Accepted April 2023; Published May 2023

In the last decades, the rise of deep learning has been of great benefit for the BCIs, mainly with the development of Convolutional Neural Networks (CNNs) as brain pattern classifiers [18,4,10]. However, an important requirement of these deep structures is that they require a big amount of data to train their many parameters. As inner speech is a relatively recent paradigm there is only a limited amount of data able for training deep models.

As mentioned before, the high variability in the EEG data makes the BCIs an instrument that needs to be highly personalized [17]. Several transfer learning and domain adaptation techniques are paving the way to take the most of the already scarce data [15,7,20,5,17], and to allow to use of data from different subjects by dealing with the difference in the data distribution.

In this work, we combine a domain adaptation technique, called Euclidian Alignment [7], which matches the data distribution from different subjects, interpreted as domains in this context. Additionally, we also combine this technique with a transfer learning method allowing us to pre-train a deep structure on several different subjects and then fine-tune the network with the subject of interest. We show that this strategy enhances the training process of a CNN that can be used in an inner speech EEG-based BCI, improving not only the classification performance but also short the training time.

2 Materials and Methods

2.1 Data description

Tasks and participants. The dataset introduced in [13], which contains EEG signals from ten healthy participants, was used for all the experiments. The participants performed several trials in three different paradigms: pronounced speech, inner speech, and visualized condition. In this study, we focus on distinguishing between the signals produced in the different paradigms, and not classifying between the different trial classes made within each paradigm, as proposed in [14].

The EEG data were acquired with a BioSemi ActiveTwo acquisition system of 128+8 channels at 1024 Hz. The data was later down-sampled 4 times and a standard EEG signal processing was applied to each subject individually. The number of available trials slightly varied among subjects. A more detailed description of the acquisition procedures, the number of trials for each subject, and preprocessing can be found in [13].

EEG processing. Each trial has a total duration of 4.5 seconds, but only 2.5 seconds corresponds to the participant executing the requested task. From those 2.5 seconds just 2 were used, to prevent any protocol-evoked EEG potential, as were described in [13].

These trials were later split into non-overlapped smaller parts, using a slide window technique, with the main goal of data augmentation. To avoid slides from the same original trial forming part of the train, test, and validation set

simultaneously, the data was first divided into different sets before applying the sliding window. This way, any performance improvements on the test and validation sets can be attributed to the model's ability to generalize to new data, rather than to the model memorizing information from the test or validation sets, as slides coming from the same original trial have a strong time dependence. Finally, the data were scaled between 0 and 1 using a min-max scaler that finds its parameters in the train set.

2.2 Classification algorithm

Convolutional Neural Networks. These architectures are one of the most used in deep learning, as it allows end-to-end learning, working as both feature extractor and classifier. They were originally introduced in [8] and had a huge impact on artificial intelligence in general and in the BCIs application in particular ever since [18,4,10,6].

The convolutional architecture employed in this study is outlined in Table ??. The kernel size was set as 3x3, with a stride and padding both set to 1. Dropout probabilities of 0.2 and 0.3 were assigned to the first two and last layers, respectively. The Exponential Linear Unit (ELU) was the activation function used. The network had 13,818 trainable parameters, with the vast majority corresponding to the linear layers. This prototype architecture was selected as it yielded the minimum number of parameters that converged on the validation set.

Voting. As each entire trial was split into smaller slides a majority voting for computing the final accuracy of the model was used. For each test trial, all the slides are fed to the network and the predicted class is obtained as the class that is predicted in the majority of the slides. As an even number of slides were used, if a tie occurs, the class of the first slide is selected as the predicted class.

Domain Adaptation and Transfer Learning. As mentioned before, the EEG data has a high inter and intra-subject variability, requiring highly personalized BCIs [17]. In more traditional approaches, the classifiers are trained only on the data from one subject, discarding all the available data from the other subjects, as the difference in the data from different subjects can jeopardize the models' performance. In our experiment, we refer to this approach as the "Simple Model". Aiming to make the most out of the available data, we use the Euclidean Alignment method [7]. This method is designed to transform the different data distributions, which in our case correspond to each subject's data distribution, to be more similar between them. This method has the great advantage to be originally designed and proposed to be applied in the EEG space, then no spectral transformation or dimensionality reduction is needed. We apply this alignment between the data from 9 subjects that were not the subject of interest. Then, we train a CNN with this aligned data. This training step was performed using a learning rate of 0.001, a batch size of 1280, and 40 and 200 as minimum and maximum training epochs, respectively. Early stopping

Table 1: Model Architecture

Layer	Output Shape	Param #
INet	[128, 2]	--
Conv2d	[128, 4, 128, 51]	40
MaxPool2d	[128, 4, 64, 25]	--
Dropout	[128, 4, 64, 25]	--
Conv2d	[128, 8, 64, 25]	296
MaxPool2d	[128, 8, 32, 12]	--
Dropout	[128, 8, 32, 12]	--
Conv2d	[128, 16, 32, 12]	1,168
MaxPool2d	[128, 16, 16, 6]	--
Dropout	[128, 16, 16, 6]	--
Linear	[128, 8]	12,296
Linear	[128, 2]	18

was also implemented, where the method stops after not getting any validation improvement after 20 epochs of tolerance.

After this pre-training process, we transfer this trained model's weights to the final network and fine-tune all the layers with the training data from the subject of interest, as in [3]. The Eucludian model learns how to transfer the data distribution of the subject of interest using only the train fold, emulating a real-world application. The fine-tuning phase was done using a learning rate of 0.001 and 200 training epochs. The early stopping tolerance and minimum epochs were 100 and 40, respectively. For this phase of the training the chosen batch size was 128, as only data from one subject is available. AdamW [9] was used as an optimizer for both parts of the training process. In test time, the alignment model transforms the test data and then is fed to the network to generate the final output. We refer to this approach as the "Pretrain-finetune Model".

Finally, aiming to evaluate the importance of the fine-tuning phase, we also test the performance of the pre-trained model over the aligned data but without performing the fine-tuning phase. We refer to this approach as the "Pretrain Model".

3 Results

3.1 Finding an appropriate window length

In the BCI research community, one of the most common data augmentation methods used is the sliding window approach, which consists in splitting the original data into smaller parts with the window's length. This technique has

two main parameters to be set: the length of the window and the stride. Setting the length of the window presents a particular trade-off: if smaller windows are used, more samples can be generated, but each slide will contain fewer samples, therefore less information. On the other hand, if larger windows are used, fewer samples can be generated but now each slide will contain more information.

To determine the optimal window length we conducted an experiment to compare the performance of the "Simple Model" trained with different window sizes, thus maximizing the amount of available data while maintaining the quality of the training samples. We set the stride parameter equal to the length of the window, so no overlapping samples were presented, in order to avoid redundant information and speed up the training process. Figure 1 depict the obtained distributions for seven different window lengths. Every data point corresponds to each subject and represents the average of the 20 validation accuracies. As expected, it can be seen that too few or too big windows are not of benefit to the model's performance, and better accuracies are obtained between the range of 0.15-0.25 milliseconds of window size. Aiming to use a safe choice a window length of 0.2 milliseconds was chosen to run the rest of the experiments.

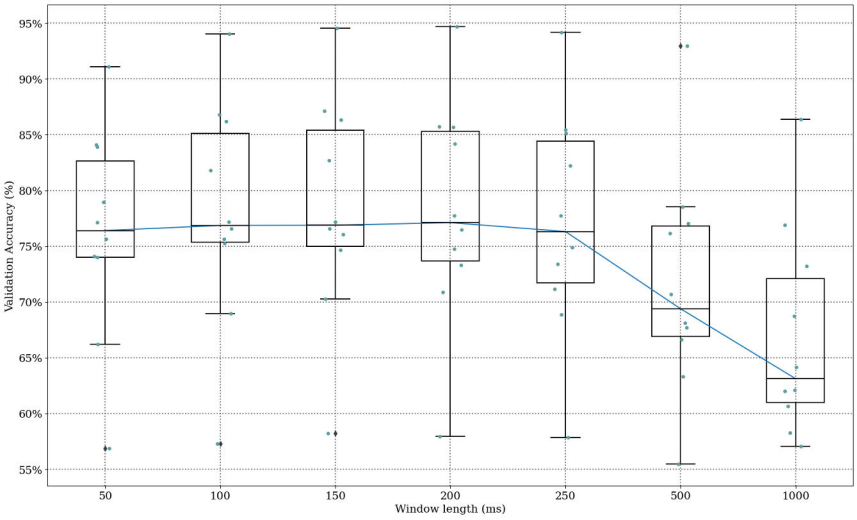

Fig. 1: Average validation accuracy distribution obtained with Simple model. Each data point represents one subject

3.2 Inner speech vs Visualized Condition

To demonstrate the difference between the three methods, a hold-out cross-validation scheme was used, with K = 20, splitting the data in train, validation, and test folds, with 70%, 10%, and 20%, respectively. The random seeds were

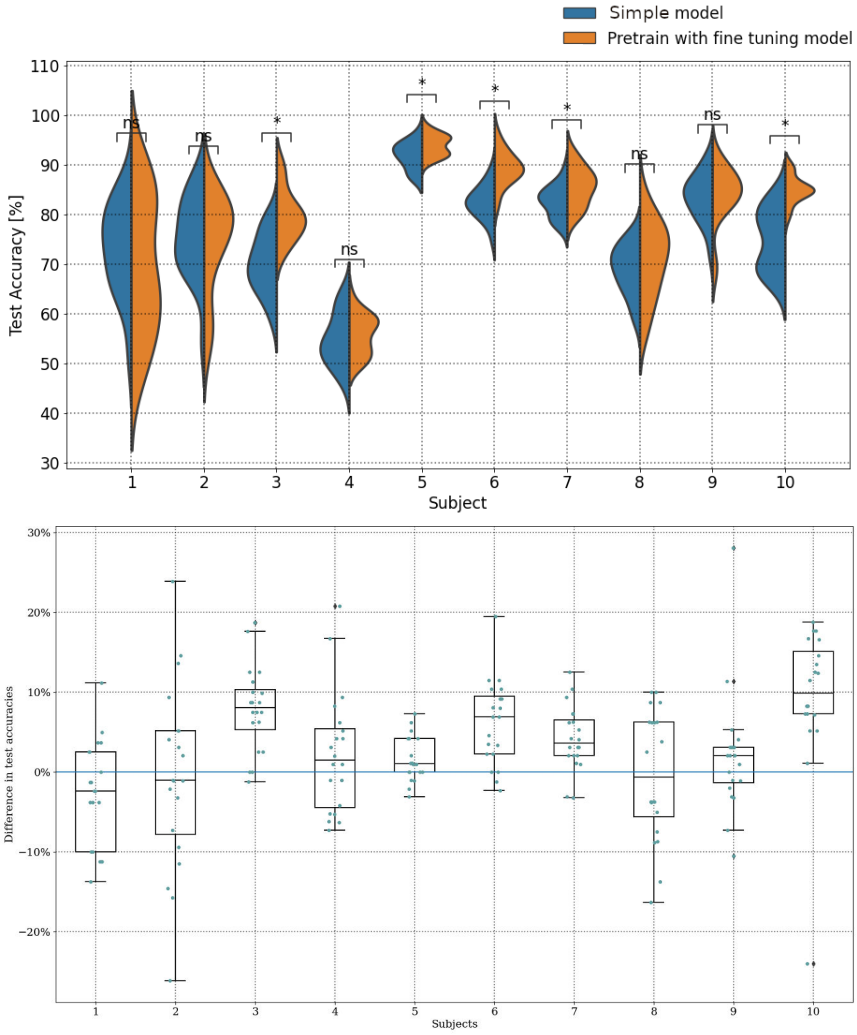

Fig. 2: Violin plots for test accuracy distribution obtained with "Simple" and "Pretrain-finetune" models. For each subject, the statistical significance according to a T-Test-Paired is marked with " * " ($p \leq 0.01$).

fixed across models so the folds were the same, aiming to make a paired and more fair comparison.

Figure 2 shows the comparison between the "Simple" and "Pretrain-finetune" Models. From Figure 2-Top, it can be seen that for five out of ten participants, the accuracy distribution was significantly improved by the "Pretrain-finetune Model" with respect to the "Simple Model", according to a T-Test-Paired with a significance threshold of 0.01. In the rest of the subjects, the "Pretrain-finetune Model" accuracy was similar to or not significantly worse that the one obtained from the "Simple Model". Figure 2-Bottom allows easier comparison between the models' performance, as the difference between the "Pretrain-finetune Model" and the "Simple Model", for each fold, is plotted.

Figure 3 clearly shows that the fine-tuning step is crucial for the network to perform properly on the subject of interest, as the "Pre-train" model is performing at chance level in almost every subject and the "Pre-train fine-tune" Model is significantly better, according with the same statistical test, in all the subjects. This supports the hypothesis that the distributions are highly variable and the network trained over different subjects is not capable to generalize over new test data coming from a different participant. We did not include in the pre-training phase the training fold corresponding to the subject of interest to simulate real-time applications. In the following, the "Pretrain Model" was no longer used.

Finally, a more detailed view of the results is shown in Table 2.

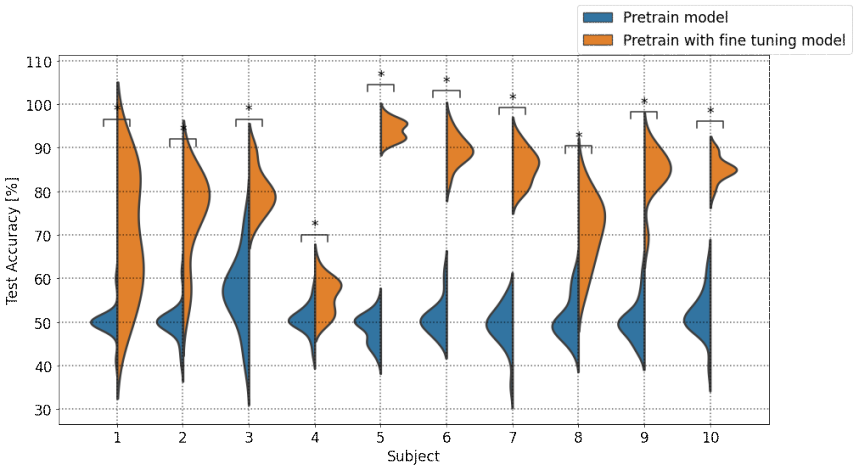

Fig. 3: Violin plots for test accuracy distribution obtained with Pretrain and Pretrain-finetune models. For each subject, the statistical significance according to a T-Test-Paired is marked with " $*$" ($p \leq 0.01$).

Table 2: Test accuracy comparison

	Simple		Pretrain		Pretrain-Finetune	
Subject	Mean	Std	Mean	Std	Mean	Std
1	74.95	11.14	50.06	3.03	69.63	13.36
2	79.68	8.42	49.74	3.75	74.17	9.64
3	79.07	5.76	57.13	8.00	80.06	4.94
4	55.25	5.85	50.36	2.97	55.83	3.88
5	94.01	3.07	48.54	3.14	94.06	2.05
6	82.32	8.14	50.92	3.80	89.43	3.39
7	84.42	4.28	49.06	4.50	85.78	3.80
8	62.49	5.91	50.88	4.65	70.94	7.41
9	83.70	7.95	50.00	4.21	84.06	5.63
10	78.17	14.16	51.77	4.83	85.16	2.66

3.3 Convergence analysis

Another key aspect of the real-scenario applications of the BCI is the convergence time of the algorithms. As recently shown in [2], pre-training approaches tend to shorten training time in fine-tuning phase. We compare the difference in the convergence speed, by comparing the epoch in which the model finishes its training (Last epoch). As the pre-training phase is usually done before acquiring the data from the subject of interest, this time was neglected from the analysis and only the fine-tuning phase was considered for the "Pretrain-finetune" method.

Figure 4-Top depicts the number of epochs the "Simple" and "Pretrain-finetune" models need to finish their training. The last epoch can be reached either for early stopping or for reaching the maximum number of epochs (200). For easier comparison, Figure 4-Bottom shows the difference of these values for each fold. A value lower than 0 implies that the "Pretrain-finetune" model needed fewer epochs than the "Simple" Model. From this Figure, it can be seen that the "Pretrain-finetune" model converged faster for five of ten subjects.

Finally, a complete summary of the results obtained in the Inner speech and Visualized condition classification are presented in Table 3 for easier comparison. In this Table, the average of the difference in each fold, both in accuracy and the last epoch are presented. The proposed training scheme is able to significantly improve the classification performance for subjects 3, 5, 6, 7, and 10 while improving the training time for subjects 1, 2, 4, 6, 7, and 8. For subject 6, a major improvement in both classification and training time performances was achieved.

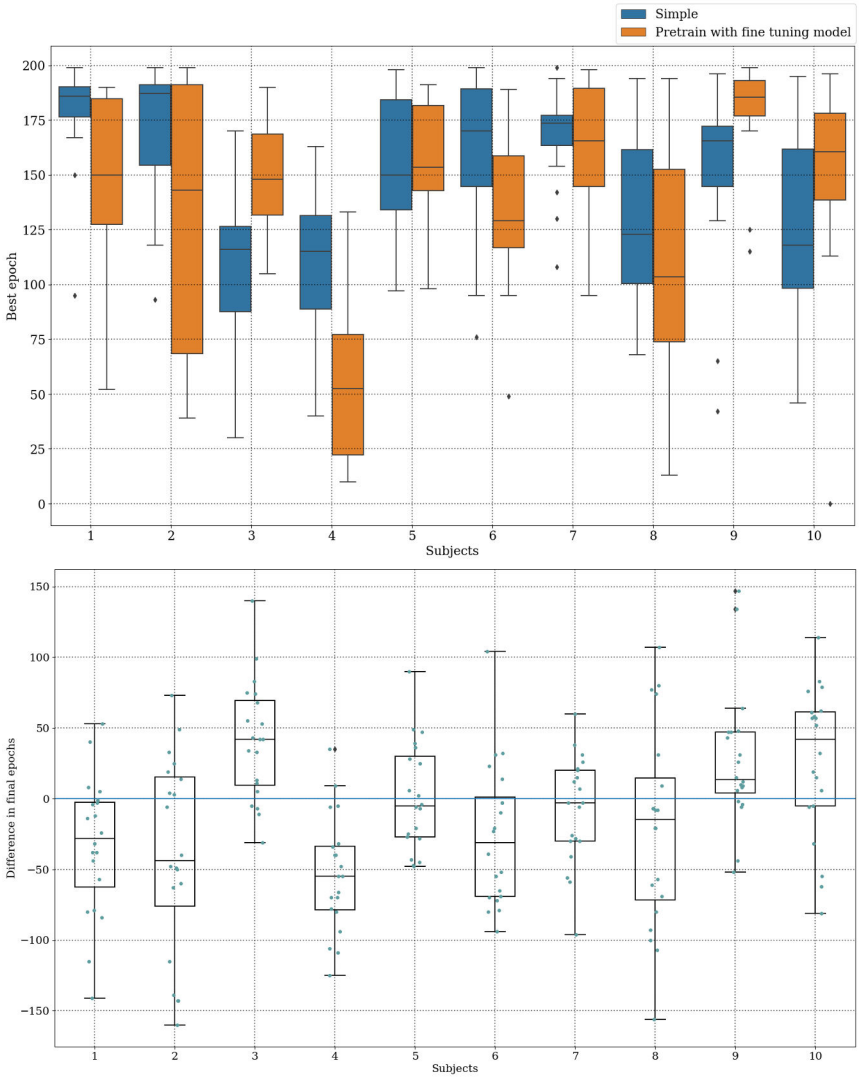

Fig. 4: Top: Last epoch distribution obtained with Simple and Pretrain-finetune models in the Inner speech vs Visualized condition classification. Each boxplot contains 20 points corresponding to each fold of the experiment. Bottom: Fold-by-fold difference.

Table 3: Summary: Inner speech vs Visualized condition

Pretrain-finetune and Simple Average Fold Differences		
Subject	Accuracy	Last Epoch
1	-2.9%	**-33**
2	0.6%	**-39.8**
3	**7.9**%	40.8
4	2.2%	**-53.45**
5	**1.6**%	1.05
6	**6.2**%	**-26.55**
7	**4.2**%	-7.4
8	-0.3%	**-20.85**
9	**2**%	26.95
10	**9.3**%	26.5

3.4 Inner speech vs Pronounced speech

The same experimental setup was used to compare the performance of the different training schemes in the inner vs pronounced speech classification. The obtained subject accuracy rate is depicted in Figure 5-Top. The results of the experiment showed that this problem is easier than the Inner speech vs Visualized condition one, consistent with [14]. From Figure 5-Bottom, it can be easily observed that the "Pretrain-finetune" approach outperformed the "Simple" model in several subjects.

Lastly, in Figure 6 the last epoch for each model is depicted. As can be clearly seen, in the majority of the subjects, a great improvement in training time is achieved, supporting the previous hypothesis that training time can be reduced by the pre-training approaches. A comprehensive view of the experiment is shown in Table 4. Although the classification accuracy was not greatly improved, mainly for the "ceiling" effect, as the problem is almost trivial to solve, the training time was largely improved, supporting again the hypothesis that for achieving comparable accuracy, the proposed approach can converge faster to the solution.

3.5 Code availability

Aiming to facilitate reproducibility and transparency in scientific research, the source code used to run the experiments is publicly available at `https://github.com/lucianozablocki/inner-speech-dl`. We hope others researchers can easily reproduce our results and continue building on this proposed approach and test alternative hypotheses or conduct further analyses using the also publicly available data.

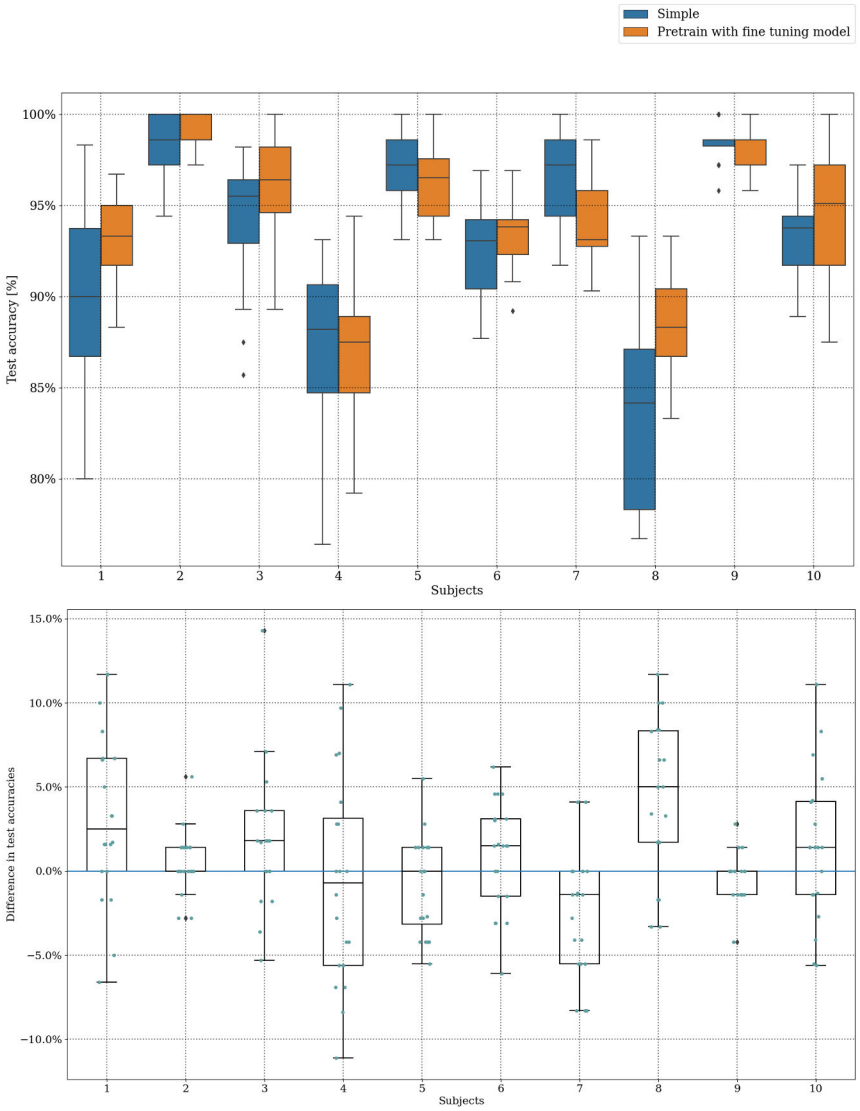

Fig. 5: Top: Test accuracy distribution obtained with Simple and Pretrain-finetune models in Inner speech and Pronounced speech classification. Bottom: Fold-by-fold difference.

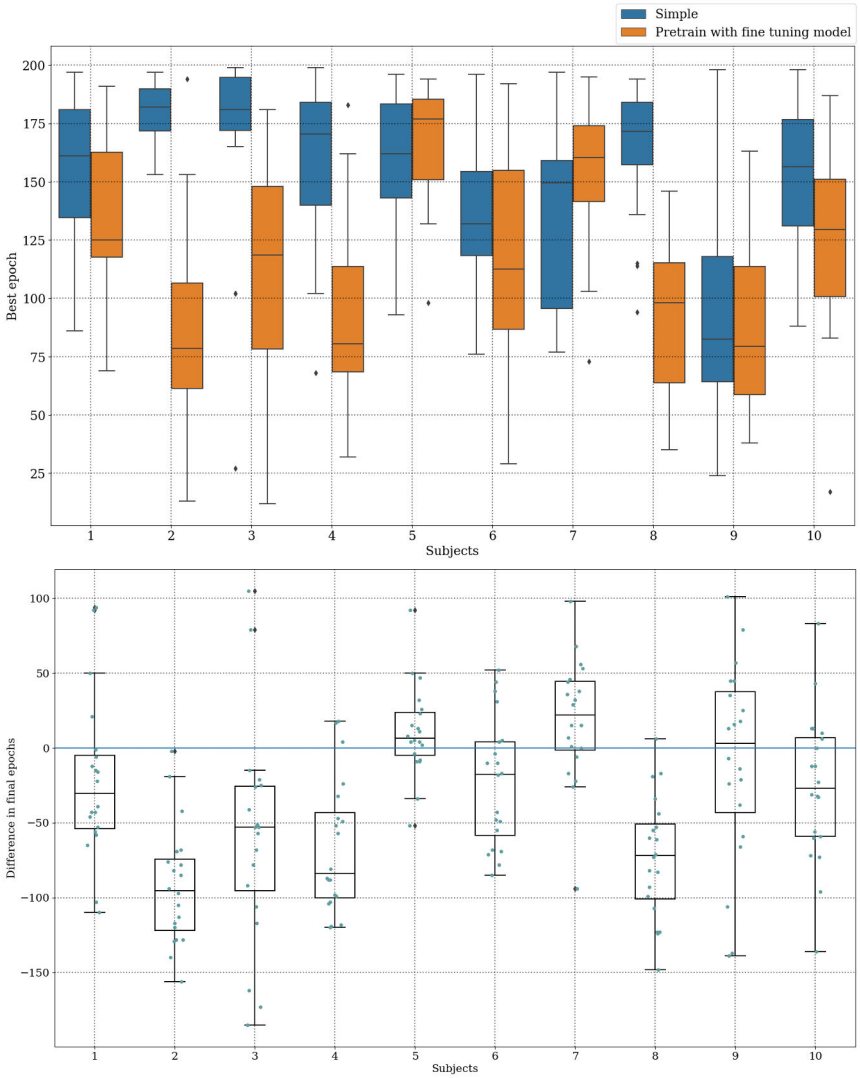

Fig. 6: Top: Last epoch distribution obtained with Simple and Pretrain-finetune models in the Inner speech vs Pronounced speech classification. Each boxplot contains 20 points corresponding to each fold of the experiment. Bottom: Fold-by-fold difference.

Table 4: Summary: Inner speech vs Pronounced speech

Pretrain-finetune and Simple Average Fold Differences		
Subject	Accuracy	Last Epoch
1	**3.2**%	**-21.55**
2	0.5%	**-92.40**
3	**1.8**%	**-59.55**
4	-0.6%	**-66.35**
5	-0.8%	10.80
6	0.9%	**-22.55**
7	-2.5%	18.65
8	**4.3**%	**-73.15**
9	0.3%	**-8.85**
10	**1.4**%	**-26.35**

4 Conclusions and Discussion

We demonstrated that the proposed training approach could incorporate new information into the classifier, significantly improving not only the classification performance but also the training time of the models. This is, without doubt, a path that could be followed to make the most out of the scarce available data when training deep learning models in BCIs applications. Needless to say, further efforts have to be done to classify the different trials generated within the same paradigm.

Another approach to explore could be to analyze the potential benefit to exclude some subjects from the pretraining step, as some of them are more difficult to classify, and therefore can be left out of the pretraining process if the information provided is not useful. This could improve the test accuracy obtained in the fine-tuning phase, and reduce the pretraining time.

Additionally, in our experiments the whole 128 available EEG channels were used as an input to the CNN, however, other approaches, like the used in [1], use just a subset of channels, keeping only the ones that are related to the left hemisphere of the brain, where more speech-related brain activity should be present.

Finally, for the fine-tuning training, all the layers have been trained again with the subject of interest. An interesting approach could be to freeze some of the layers, leaving the parameters unchanged, which can lead to an even faster training time.

References

1. van den Berg, B., van Donkelaar, S., Alimardani, M.: Inner speech classification using EEG signals: A deep learning approach. 2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS) pp. 1–4 (2021)
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S.: Language models are few-shot learners. arXiv preprint arXiv:2105.14447 (2021)
3. Chu, B., Madhavan, V., Beijbom, O., Hoffman, J., Darrell, T.: Best practices for fine-tuning visual classifiers to new domains. European conference on computer vision (2016)
4. Cooney, C., Korik, A., Raffaella, F., Coyle, D.: Classification of imagined spoken word-pairs using convolutional neural networks. In: The 8th Graz BCI Conference, 2019. pp. 338–343 (2019)
5. Courty, N., Flamary, R., Tuia, D., Rakotomamonjy, A.: Optimal transport for domain adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(9), 1853–1865 (2017)
6. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
7. He, H., Wu, D.: Transfer learning for brain–computer interfaces: A euclidean space data alignment approach. IEEE Transactions on Biomedical Engineering **67**(2), 399–410 (2019)
8. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
9. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
10. Mao, W., Fathurrahman, H., Lee, Y., Chang, T.: EEG dataset classification using CNN method. In: Journal of physics: conference series. vol. 1456, p. 012017. IOP Publishing (2020)
11. Nicolas-Alonso, L.F., Gomez-Gil, J.: Brain computer interfaces, a review. sensors **12**(2), 1211–1279 (2012)
12. Nieto, N.: Algoritmos para interfaces cerebro-computadora en paradigmas relacionados con el habla (2022)
13. Nieto, N., Peterson, V., Rufiner, H.L., Kamienkowski, J.E., Spies, R.: Thinking out loud, an open-access EEG-based BCI dataset for inner speech recognition. Scientific Data **9**(1), 1–17 (2022)
14. Nieto, N., Runer, H.L., Spies, R.: Preliminary feasibility analysis of inner speech as a control paradigm for brain-computer interfaces. In: XXII Simposio Argentino de Inteligencia artificial (ASSAI 2021)-JAIIO 50 (Modalidad virtual) (2021)
15. Peterson, V., Nieto, N., Wyser, D., Lambercy, O., Gassert, R., Milone, D.H., Spies, R.D.: Transfer learning based on optimal transport for motor imagery brain-computer interfaces. IEEE Transactions on Biomedical Engineering **69**(2), 807–817 (2021)
16. Rousseau, M.C., Baumstarck, K., Alessandrini, M., Blandin, V., De Villemeur, T.B., Auquier, P.: Quality of life in patients with locked-in syndrome: Evolution over a 6-year period. Orphanet journal of rare diseases **10**(1), 1–8 (2015)
17. Samek, W., Meinecke, F.C., Müller, K.R.: Transferring subspaces between subjects in brain–computer interfacing. IEEE Transactions on Biomedical Engineering **60**(8), 2289–2298 (2013)
18. Tang, Z., Li, C., Sun, S.: Single-trial EEG classification of motor imagery using deep convolutional neural networks. Optik **130**, 11–18 (2017)

19. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. Clinical Neurophysiology **113**(6), 767–791 (2002)
20. Wu, D., Xu, Y., Lu, B.: Transfer learning for EEG-Based Brain-Computer Interfaces: A review of progresses since 2016. arXiv preprint arXiv:2004.06286 (2020)