POC-CSP: A novel Parameterised and Orthogonally-Constrained Neural Network layer for learning Common Spatial Patterns (CSP) in EEG signals

First name, Last name

Melbourne, Australia

E-mail: name@unimelb.edu.au

September 2023

Abstract. Common Spatial Patterns (CSP) has been established as a power feature extraction method in EEG signal processing with machine learning. However, it has shortcomings like sensitivity to noise and rigidity in the value of the weights. In this paper, we introduce a novel neural network architecture layer (POC-CSP) that transforms CSP into a trainable machine learning model that can learn from training data, regularised, and be used in an end-to-end classification network. Evaluating POC-CSP, we show it outperforms conventional CSP in BCI motor imagery task and has the ability to generalise well to unseen data if uses as a pre-trained neural network. POC-CSP can be prepended to any neural network classifier and trained end-to-end to improve EEG signal decoding.

Keywords: Common Spatial Patterns, Neural Networks, BCI Motor Imagery

1. Introduction

The use of Common Spatial Pattern (CSP)[1] has garnered special interest for analysing multichannel electroencephalography (EEG) or magnetoencephalography (MEG) data. CSP is a spatial filter that maximises the difference in variance between two classes of signals, such as movement and non-movement, or rest and mental activity, and can be extended to more than two classes. CSP has broad implications in various fields of signal processing, including Brain-Computer Interfaces (BCI) and image processing [2].

Motor Imagery (MI) in BCIs infers and distinguishes different imagined movements, such as left hand, right hand, or foot movements, based on acquired brain signals. These systems can help individuals with movement disorders, such as locked-in syndrome, tetraplegia, or cerebral palsy, by establishing a direct connection between the brain and an external device, such as a speller, wheelchair, or prosthesis, in order to control them [3, 4].

MI systems process some form of brain signal (e.g., EEG) and use statistical or machine learning techniques to distinguish movement classes. CSP can be used in this process to extract features from EEG signals. The extracted features can be used to train machine learning algorithms to recognize the intention of the user and control external devices. CSP has shown promising results in improving the accuracy and robustness of MI-based BCI.

CSP learns spatial filters that maximise the variance of EEG signals filtered in a specific frequency band for one class and minimise the variance for the other class [5, 6]. The variance of EEG signals in a particular frequency band represents the power of the signal in that band, so CSP can optimise discrimination for BCI through band power features [5]. These properties have established CSP as a powerful tool in BCI signal processing, especially MI. It provides various benefits in BCI tasks, which makes it a highly utilised method. Some of the useful CSP features are the following.

(i) ***Offline Calibration:*** CSP is typically trained offline using labelled data collected during calibration or training sessions. This offline training phase allows CSP to adapt to the individual characteristics of each user's brain activity, improving the performance of the BCI system. By learning the spatial filters from a training dataset, CSP can generalise its discrimination capabilities to unseen EEG data during online BCI operation, where real-time control is required [7].

(ii) ***Adaptability to Different BCI Paradigms:*** The versatility of CSP makes it applicable to various BCI paradigms beyond motor imagery. It has been successfully employed in different domains, such as speech recognition, visual attention detection, and cognitive workload assessment. The ability of CSP to extract relevant spatial features from EEG signals makes it a valuable tool for enhancing the discrimination of mental states in a wide range of BCI applications [8].

(iii) ***Frequency Band Optimization:*** In addition to spatial filtering, CSP can be extended to optimise the frequency bands used for discrimination. By focusing

on specific frequency bands associated with the desired brain activity, CSP can further enhance the separation between classes. This frequency band optimization is achieved by adjusting the spatial filters for different frequency sub-bands, allowing CSP to use the distinctive characteristics of brain rhythms in each band. By leveraging frequency-specific information, CSP provides more accurate and robust discrimination in BCI systems [9].

However, despite its strengths, the CSP method has some shortcomings and can be further improved. The aim of this research is to address these shortcomings and improve the utility of CSP in machine learning-based BCI.

We have reimagined CSP as a trainable Machine Learning algorithm, specifically a Neural Network (NN). We assume a learning phase and an inference phase for conventional CSP algorithms. In the learning phase, the weights are learned by calculating eigenvectors of the covariance matrix of data; in the inference phase, these weights are applied to the input signal. These vectors learn the directions that maximise variance for one class and minimise it for the other one in a binary setting.

In our architecture, we propose a NN layer mimicking CSP for feature extraction, which can be used to create end-to-end NN models. It is trainable with backpropagation and can easily be implemented with most of the common NN frameworks (e.g., Pytorch). We applied a set of constraints on the weights through the NN layer architecture to guide the training process to learn CSP-like weights.

Several benefits can be achieved through this architecture and these are the contributions of this research:

(i) Improving downstream classification accuracy by reducing noise and cross-subject variation. CSP is known for its sensitivity to noise and tendency to overfit with small training sets [10]. CSP finds spatial patterns using most significant variance directions of the data of each class. Although it normalises common intraclass variances, class-specific data variances can still represent some noise. It is not clear if these variance directions found by CSP necessarily account for target class discrimination or if they are just random high oscillations caused by an out-of-system factor or EEG cap artefact. By implementing CSP as an NN layer, we are able to reduce the size of the parameter set of the CSP module significantly (half of the size of [11]) and also utilise NN-specific regularisation methods, like drop-out to make the model more robust.

(ii) Improving downstream classification accuracy by guiding the classification process through supervised labels. In our model, learning spatial filters is treated as an optimization task and is guided by sample label supervision so that it can find filters that contribute the most to class discrimination. This enables the optimisation to explore a broad search space of parameters that can hold any value as long as they satisfy the optimisation objective. This is as opposed to the rigid closed-form formula of traditional CSP where weights are achieved based on statistical properties.

(iii) CSP can be a part of an end-to-end NN with pre-processing and classifier layers. End-to-end training has multiple benefits, namely reducing the need for domain experts, training all components with a single objective consistent with the model's main goal, and simplifying the model [12, 13]. Recently, various attempts have been made to replace traditional ML pipelines of sequential steps with end-to-end models in different domains [14]. By implementing CSP as an NN layer, we can incorporate it into any end-to-end network instead of having it as a separate pre-processing step. Although end-to-end models may need more training data due to larger parameter space, we have integrated CSP into an end-to-end model while keeping the size of parameter space small. This has not resulted in any loss in performance, so we have avoided the shortcomings of end-to-end models, while leveraging their benefits.

We validated our proposed architecture using the BCI Competition IV public dataset [15], which is one of the most benchmarked BCI datasets available. Our proposed architecture outperforms traditional formulated CSP and previous attempts at different implementations of CSP as a part of a machine learning architecture. We have also shown how the proposed architecture can mimic CSP behaviour.

2. Method

Our overall goal was to create a Neural Network (NN) layer mimicking CSP (POC-CSP) for feature extraction, which can be then used to create an end-to-end NN-based BCI model. To achieve this, we studied the properties and components of a CSP algorithm, then used the characteristics of CSP components to create the equivalent neural network modules. We evaluated the resulting model in a number of experiments using a publicly available dataset. In this section, we outline this dataset, the architecture details, and how we validated our model.

2.1. Data

We chose a publicly available and highly benchmarked dataset to evaluate our approach and made comparison with existing CSP approaches and variants. The BCI competition dataset is one of the most well-known public EEG datasets and is fit for this purpose. The BCI Competition IV-2a dataset [15] consists of recordings with 22 EEG electrodes on nine participants. Signals between 0.5 Hz and 100 Hz were bandpass filtered before being sampled at 250 Hz. The left hand, right hand, both feet, and tongue were the four body parts that all participants were asked to imagine moving. Additionally, data on eye movements is provided by three electrooculography (EOG) channels. This provides a multi-class classification framework that is usually a point of struggle for CSP algorithms.

The dataset consists of two sessions per individual, each of which has 288 trials. We utilised the first session for training and the second for testing. Each trial consisted of 2 seconds of inactivity, 1 second of cue, 4 seconds of motor imagery activity (overlapping with the cue), and 1.5 seconds of break. We only used the 4 seconds of motor imagery for the experiments. To prepare the data, we removed DC using a high pass filter with a cutoff frequency of 2 Hz [15].

2.2. The Common Spatial Patterns (CSP) Method

The CSP method [1] is an algorithm that aims to find spatial components in signals that simultaneously maximise the variance of one class (e.g., right-hand movement) while minimising it for the other classes (e.g., left-hand and feet movements). Therefore, CSP can separate a multivariate signal into a set of additive subcomponents. This method is based on Principal Component Analysis (PCA), which ranks components in the order of the most variance represented in the whole dataset. However, unlike PCA, CSP normalises the common variance between classes by applying a transformation (whitening) before finding the components that can contribute the most in distinguishing the target class from the other classes.

To calculate CSP, we start from a given zero-centred input signal X, the normalised

covariance matrix can be calculated as follows:

$$R = \frac{XX^T}{trace(XX^T)} \tag{1}$$

where $trace(.)$ is the sum of the diagonal elements of a given matrix. We can decompose R using Eigen analysis to obtain an eigenvector matrix U and corresponding eigenvalues λ where U is $orthonormal(UU^T = I)$ and λ contains covariances of the data in the directions of column vectors of U. Here, U can be applied to the data as a transformation $Y = U^T X$ to get decorrelated temporal patterns. Meaning that $YY^T = \lambda$. Then dividing the decorrelated data by λ will standardise the data by setting its covariances to I. Applying these transformations sequentially we obtain:

$$P = \sqrt{\frac{1}{\lambda}}U \tag{2}$$

This is the whitening transformation. If we calculate covariance matrices R_p for positive classes and R_n for negative classes, then $R_p + R_n = R$. Therefore, applying the above whitening transformation will lead to R_p and R_n to share the same eigenvectors B with their eigenvalues being reversely ordered [16]. This means that directions with highest variance for positive class will have the lowest variance for negative class and vice versa. This is an important property and we use this later on in developing our POC-CSP layer.

With B and P defined, the CSP transform can be written and applied to the input as follows:

$$Y = B^T P X \tag{3}$$

Here, B is an orthonormal matrix that, when applied to a datapoint, will project it on a new coordinate system. Thus, B can be interpreted as a new basis for our data that helps with better discrimination of target classes. B is also referred to as CSP weights.

The CSP algorithm in its original form considers the CSP weights as an orthonormal basis. In other words, applying CSP weights will be similar to a change of coordinate system, where it rotates the original coordinate system to the one in which data has the most variance for one class along some axis compared to other classes.

In order to replicate these CSP characteristics in a NN layer, one possible approach is to consider CSP weights to be the trainable parameters of our model [11]. However, this is not sufficient in achieving a CSP-like vector of features as there are no conditions and constraints to guide the training process. Weights in this method lack the characteristics of the CSP weights. To overcome this issue, we put constraints on the weights to make sure the model will learn weights most similar to CSP's.

2.3. CSP Constraints

As discussed, CSP essentially finds a basis transformation for the data that is assumed to create a better discrimination of target classes. However, this new basis might not be able to capture the most informative spatial patterns as there might be noise in the positive or negative classes with a higher variance that can misguide this algorithm and cause overfitting. In our approach, we overcome this challenge by learning in a NN using backpropagation. We expect the optimisation process to learn the most relevant spatial patterns for classifying the target classes. This means our model can learn to ignore unimportant and environmental noise while learning relevant information about the specified task.

Applying orthogonal constraints on linear and CNN models can help with better generalisation and avoiding overfitting [17]. Constraining model weights to orthonormal matrices will force the model to learn the most informative uncorrelated patterns. Meanwhile, the model will have more freedom to explore compared to conventional CSP as the weights are not derived by a closed-form formula, while the training is more directed than CSP-NN [11] to create CSP-like features. Therefore, our approach can avoid overfitting and provide a better generalisation.

Another important feature of CSP basis function is the type of transformation it represents. B is an orthogonal matrix made of eigenvectors of covariance matrix R; therefore, it represents a rotation or reflections transformation in any dimensioned space. However, our POC-CSP layer is not guaranteed to create rotation or reflection transformations in spaces of higher than three dimensions. These transformations represent what is called rigid motions, which are isometries and, therefore, distance preserving maps [18]. Such rigid motions preserve dot products, shapes, and angles as well and, therefore, samples will remain discriminative after applying the transformation. So, in our architecture, when we constrain the POC-SCP layer weights to special orthogonal matrices, it will not necessarily learn a rotation or reflection like the CSP algorithm; however, because the learnt transformation is a rigid motion, this does not negatively impact discrimination between class samples.

2.4. Parameterisation

With the constraints defined, we need to convert this constrained problem into an unconstrained one in order for the NN optimiser to learn the weights in a machine learning paradigm. To this end, we convert the constrained problem into an unconstrained problem using a parameterisation method based in Lie Group theory [17]. This approach has the benefit of creating a non-constrained optimisation problem without creating additional minima compared to the original optimisation.

The model weights are parameterised as a skew symmetric matrix that is then mapped back to orthogonal space using an exponential map. In the theory of Lie groups, the Lie exponential map is the connection between the geometry of Lie groups and the structure of their Lie algebras. Considering the special orthogonal group, the

Lie Algebra of the group will be the tangent space at the identity element of the group defined as skew-symmetric matrices:

$$so(n) = \{A \in R^{n*n} | A + A^T = 0\} \tag{4}$$

where A is an upper triangular matrix with its diagonal elements being zero. We can consider $A \in R^{\frac{n*(n-1)}{2}}$ since most of the elements of are zero. This reduces the number of parameters of our layer to almost half, which, in turn, expedites training and reduces overfitting.

Having A , we can obtain as follows:

$$B = \exp(A) \tag{5}$$

where $\exp(A)$ is the matrix exponential map of A. Just like CSP, we consider $NC * C$ matrices as layer weights, where N is the number of classes and C is the number of input signal channels (EEG channels in our application). These matrices, as detailed above, are parameterised with skew-symmetric matrices, which can be mapped back to orthogonal matrices using a matrix exponential map [17].

2.5. POC-CSP Layer

To create the NN layer, we convolve each column of each of these orthogonal matrices as 1D weight vectors of length C along the temporal dimension. This will be equivalent to transforming data by applying the transformation matrix B in the CSP algorithm, but in a NN fashion.

We define the trainable parameters of the layer to be an upper-triangular matrix. So, starting from an upper-triangular matrix, we apply the exponential map to obtain our orthogonal matrix. So, like CSP, our answer is an orthonormal matrix that can represent independent special features. The outline of this layer is displayed in Figure 1

Up to this point, we have left out the whitening step from our model design. Since the whitening transformation is not an orthogonal nor a distance preserving transformation [16], and also it is an entirely statistical measure, it would be better to apply it as a pre-processing step directly on the input data. Using the property mentioned above where $R_p + R_n = R$, applying PCA pre-whitening to input data and then transforming these data using parameterised B is equivalent to the CSP transformation.

2.6. Validation

Here, we compare and validate our model against the traditional CSP algorithm. First, since the final weights of the CSP are orthonormal, we constrained our model weights to achieve this. From the algebraic definition, we have that each skew-symmetric matrix can be mapped to an orthogonal matrix in the special orthogonal group.

Another CSP feature is that the resulting weights are rotation transformations. In our case, since we did not limit the weights in such a way, our weights are rigid

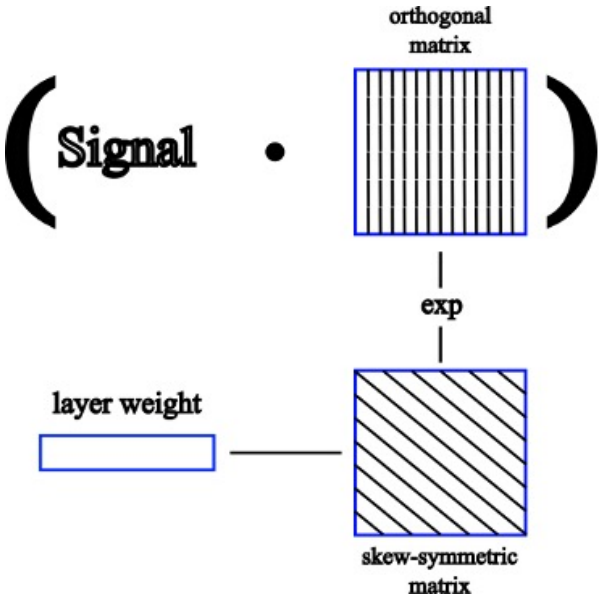

Figure 1. Schematic of POC-CSP layer. Layer parameters are a vector where a skew-symmetric matrix is created with it. Then applying an exponential map on the skew-symmetric matrix gives an orthogonal matrix, which is the transformation applied to signals.

motion transformations. As we demonstrated before, not only does this feature not negatively affect the optimisation process, but also it encompasses a larger search space and, therefore, can find more accurate weights during the training phase. So, we consider this an improvement over the CSP algorithm.

Finally, to test the neural network end-to-end, we empirically validated the POC-CSP layer output to that of the traditional CSP algorithm, using the same input. For the purposes of this test, we assumed the weights derived from the closed-form CSP formula are the trained weights. To this end, we first solved the problem with a traditional CSP algorithm to obtain the CSP weights and results of applying them on the input. Then, we used these weights as our CSP layer weights (skipping training for this test) and ran the NN in a feed-forward manner on frozen model weights. The results were the same as the CSP output. Therefore, if the model can learn the same weights as CSP weights, the layer output will be the same as can be achieved with closed-form CSP. In the next section, we will elaborate more on how we trained the POC-CSP weights using training data.

3. Results

In this section, we will elaborate on a series of experiments we performed to evaluate the POC-CSP layer and benchmark its performance against other architectures in a motor imagery task.

First, we benchmarked the performance of POC-CSP against both the conventional CSP and the CSP-NN architecture of [11] (Experiment 1). POC-CSP outperformed both these architectures on the BCI IV 2a dataset for motor imagery.

In Experiment 2, we aimed to derive the best end-to-end architecture using the POC-CSP layer feature extractor. The CSP algorithm acts as a feature extractor that can be prepended to a classifier. We evaluated POC-CSP with three different neural network classifier modules for this experiment:

(i) A feed-forward fully-connected neural network (DNN)
(ii) A convolutional neural network (CNN)
(iii) A hybrid neural network called EEGNet introduced in [19] (EEGNet)

Finally in Experiment 3, we evaluated the experiment 2's model performance with two paradigms of subject-specific and multi-subject evaluation. The subject-specific approach provides insights into how well the models can perform on different subjects and their individual EEG signal characteristics. This is crucial because inter-subject variability is a major challenge in EEG-based BCI systems. The multi-subject approach, on the other hand, is useful for assessing the model's generalisation capability and its ability to perform well on unseen data.

Most previous work in this field have reported their results using the subject-specific approach, since learning an individual participant's EEG patterns is more tractable for current models compared to generalising on EEG data of multiple participants. We found out that all of the tested previous methods suffered a significant drop of accuracy when trained on the whole dataset of all subjects combined. This is a shortcoming we are addressing in our evaluation.

3.1. Experiment 1

The aim of the first experiment was to benchmark POC-CSP's performance against the traditional CSP and the only other model in literature that has worked on a neural network representation for CSP [11].

Three models were compared in this experiment, each having a different CSP-based feature extractor combined with the same simple classifier network. The Hybrid-CSP, which is the conventional CSP as feature extractor; the CSP-NN network of [11], which according to the authors is a convolutional neural network representation of CSP; and POC-CSP, which is our CSP neural network layer.

Since the work of [11] has not been open-sourced, we ran experiments based on our own implementation of their work, keeping the hyperparameters as close as possible to the original paper where they were available.

Table 1. A comparison of 3 models in experiment 1 on training and validation accuracy

Model	*Training acc.*	*Validation acc.*
Hybrid-CSP	0.47	0.44
CSP-NN [11]	0.42	0.43
POC-CSP	**0.55**	**0.52**

All models have a fully-connected simple neural network layer used as a classifier. It takes features generated by the CSP models to perform classification. This has one hidden layer of 500 units with hyperbolic tangent activation and a dropout rate of 0.5. Then another 4-unit dense layer with a softmax activation classifies the 4 different labels.

The network is trained with SGD and Nesterov with a momentum of 0.9 and a learning rate of 0.0001. We found that the early stopping strategy used by [11] was preventing the model from completing its training. Hence, we trained all models for 1000 epochs and selected the model weights from an epoch with the highest validation accuracy for comparison.

We found our model converging faster and having a smoother learning curve with less fluctuations. It also outperformed both models on the BCI IV 2a dataset for motor imagery. The results are summarised in table 1:

This experiment shows that our hypothesis that POC-CSP improves on CSP's feature extraction capabilities holds and it can improve an end-to-end classification system. Although CSP-NN shows similar performance to conventional CSP, in the discussion section, we argue that their network is not a CSP function and is more similar to a convolutional neural network.

3.2. Experiment 2

Experiment 1 served as a benchmark to evaluate POC-CSP against conventional CSP and CSP-NN. In order to keep the comparisons fair, we restricted ourselves to using the classifier sub-network and hyperparameters used in [11]. However, those conditions are not the optimal setting for the best end-to-end classifier for the motor imagery task. In experiment 2, we aimed to find the best fitted classifier sub-network taking into account the specifications of features extracted by the POC-CSP module.

We experimented with 3 different neural-network based classifiers to appended to the POC-CSP layer and trained in an end-to-end fashion.

3.2.1. ***Feed-forward, fully-connected neural network (DNN)*** This classifier has one hidden layer of 500 neurons with hyperbolic tangent activation, a dropout rate of 0.5 to regularise for overfitting, followed by a classification layer of 4 neurons with a softmax activation for classifying the 4 class labels.

3.2.2. ***Convolutional neural network (CNN)*** This classifier has two 2D convolution layers, each with 10 filters, a stride size of one, and Rectified Linear Unit (ReLU) activation functions. The initial kernel size is set to (25, 1), followed by a subsequent kernel size of (7, 1). Batch normalisation is used after each convolutional layer, ensuring stable and accelerated convergence during training. The convolution layers are followed by flattening and two fully-connected layers with ReLU activation. A duo of dropout layers, each with a 0.25 dropout rate, is interposed within this segment. Finally, a dense output layer with softmax activation is used to classify inputs into four distinct classes.

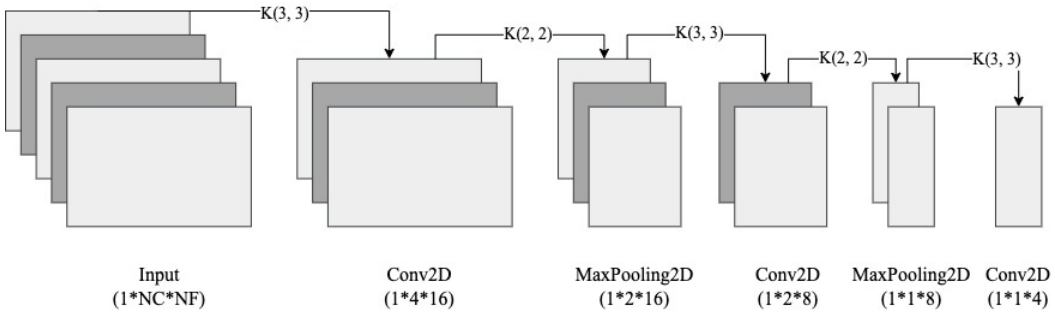

Figure 2. The CNN architecture, showing the layers and their configuration NC = Number of Classes, NF = Number of CSP Feature[PLACEHOLDER]

3.2.3. ***EEGNet*** This classifier follows the architecture proposed by [10]. It consists of a 2D convolution layer with linear activation, padding mode set to the same and filters are of size (1, 64). A batch normalisation is applied followed by a 2D depthwise convolution of size (C, 1) with linear activation and another batch normalisation layer. An ELU activation is applied after the batch norm layer, then the features are averaged pooled with a kernel of size (1, 4) followed by a dropout layer with 0.5 rate. Next, a separable convolution with a kernel size of (1, 16) is followed by a batch norm, an ELU activation, a pooling layer with kernel size (1, 8) and a dropout of 0.5. Then the features are flattened and classified with a fully connected layer and a softmax activation of 4 neurons.

Other hyper parameters, including learning rate, batch size, dropout, and momentum, were explored for a combination yielding the best performance. To show that features extracted by POC-CSP are more discriminating, we also experimented with conventional CSP features combined with these classifiers. The results of the main experiments are shown in Table 2.

We found that the combination of POC-CSP and EEGNetv2 trained with the batch size of 32 and dropout rate of 0.5 trained with [optimizer (with momentum if SGD)] yielded the best performance. In this setting, we used a learning rate decay with an initial learning rate of 0.001, applied every 250 epochs, and reduced the rate by a factor of 0.1 each time. This helps optimise the training by gradually reducing the step size

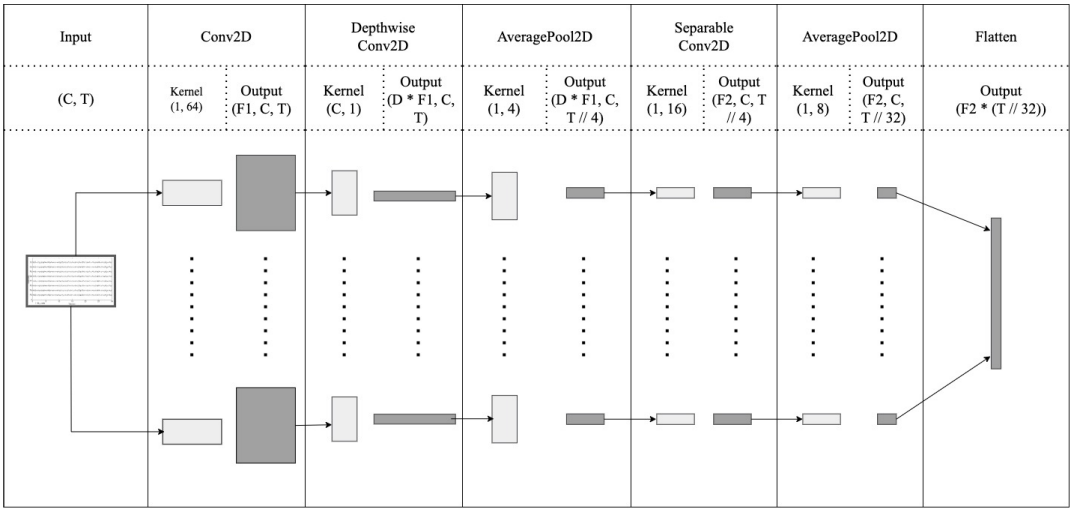

Figure 3. The EEGNet architecture, showing the layers and their configuration [PLACEHOLDER], In the EEGNet architecture, denoted as C (for the number of channels), T (for the number of time points), F1 (representing the number of temporal filters), D (signifying the depth multiplier, i.e., the number of spatial filters), F2 (corresponding to the number of pointwise filters), and N (indicating the number of classes) are the relevant parameters.

Table 2. A comparison of different classifiers appended to both POC-CSP layer and conventional CSP for an end-to-end motor imagery BCI decoder

Model	*Training acc.*	*Validation acc.*
$CSP–NN(Experiment1)$	0.425	0.43
$CSP + DNN$	0.47	0.44
$CSP + CNN$	0.55	0.45
$CSP + EEGNet$	0.00	0.00
$POC–CSP + DNN$	0.69	0.53
$POC–CSP + CNN$	0.78	0.56
$POC - CSP + EEGNetV2$	**0.81**	**0.70**

to fine-tune model convergence.

Model capacity increases by replacing the simple DNN classifier with CNN and further increase is observed with EEGNet. As expected, conventional CSP is outperformed by POC-CSP, where even with the EEGNet as a classifier, the performance did not reach the level observed in POC-CSP and a DNN classifier.

3.3. Experiment 3

Experiment 2 allowed us to assess the best configuration and architecture for an end-to-end BCI decoder with the POC-CSP layer. The evaluation approach used in this and experiment 1 was the standard subject-specific approach. In the subject-specific

approach, models are trained on a portion of a subject's data individually (here the training subset of the BCI IV 2a dataset for each participant), and evaluated on the unseen data from the same subject (here the validation subset of the same participant). The resulting model performance is the average accuracy among all subjects. This approach enables investigation of the effect of inter-subject variability on accuracy and is the dominant approach in the literature for benchmarking decoders. However, the subject-specific approach has shortcomings both in achieving the best possible performance and in providing a full picture of the model capability. A model trained in this approach can never generalise to new subjects without first obtaining significant data from the new subject, in fact, it is not a single model, but a set of subject-specific models. Moreover, this approach is not leveraging the commonalities existing between subjects to improve models or reduce the amount of training data required. In experiment 3, we introduced a multi-subject approach to address these shortcomings and compared the results with the subject-specific approach. The multi-subject approach involves training the model once on all participants using aggregated training superset data. The trained model is evaluated on the combination of the validation data of all participants. The model can be used like this for any new participant or alternatively this can act as a pre-trained model and further enhanced by fine-tuning it on a small training set of a new subject. We have evaluated both approaches in this experiment using the best model derived in the previous experiment (POC-CSP layer with EEGNet). In the fine-tuned version, we left one subject out for the pre-training (using the entire data of the remaining subjects) and then fine-tuned the model on 50% of that subject's data. The evaluation was performed on the remaining 50% of the data. This is a 30% reduction in the amount of training data of that specific subject compared to the subject-specific approach. The results are shown in Table 3. The pre-trained and fine-tuned model not only achieved a higher accuracy than the subject-specific model, but is also more practical in a real-world BCI scenario. This is a significant achievement as a single pre-trained model can be used with any new subject with less training required to fine-tune the model.

Table 3. the performance of multi-subject (with (A) and without fine-tuning (B)) and subject-specific (Exp 2) (C) results achieved from the POC-CSP/EEGNet model, compared across each participant P1 to P9 and on average.

Participants	*A*	*B*	*C*
P1	**0.85**	0.79	0.81
P2	0.56	0.51	**0.59**
P3	**0.92**	0.90	0.84
P4	0.67	0.57	**0.69**
P5	**0.53**	0.34	0.46
P6	**0.55**	0.42	0.52
P7	**0.81**	0.63	0.78
P8	0.81	**0.82**	0.78
P9	**0.83**	0.76	0.81
Average	**0.725**	0.64	0.70

4. Discussion & Conclusion

In this research, we proposed a novel neural network architecture (the POC-CSP Layer) that transforms the Common Spatial Patterns (CSP) operations into a trainable machine learning model. By doing so, we managed to improve the CSP algorithm in better generalisation, better learning from data, and being able to integrate into an end-to-end neural network. We achieve this by modelling CSP's defining constraints as a neural network layer.

To validate these claims, we first showed that POC-CSP re-creates the fundamental characteristics of CSP and thus it can be a replacement for it. We then evaluated its performance against conventional CSP and the CSP-NN architecture [11] that claims a neural network-based CSP model. POC-CSP outperforms both models, supporting our claims that it is an improvement over the conventional CSP.

We argue that the CSP-NN architecture cannot be considered a CSP equivalent as it does not mimic CSP's behaviour nor it adheres to the constraints that define the CSP operation. CSP-NN is simply a neural network for EEG signal feature extraction. Moreover, when compared to the POC-CSP layer in a similar experiment, it shows an inferior performance.

We also developed an end-to-end network for BCI motor imagery decoding using POC-CSP, after experimenting with different pre-processing and classifier options. These experiments highlighted the importance of the classifier component of the end-to-end network to be able to use the rich features extracted using our POC-CSP layer.

Finally, we showed how the model can be pre-trained on a dataset and fine-tuned on new subjects using reduced training data, thus creating a practical MI decoder. This addresses a shortcoming in BCI research, where mostly separate models are trained for each trial participant. Not only this approach has very limited generalisation capacity, but also it is not utilising the commonalities between subjects to improve the overall performance of the model. We have shown that a pre-trained foundation model tuned on a subset of data of any new subject performs better than a model trained specifically on that subject with more subject-specific training data. This is due to the foundation model's ability to utilise a much larger multi-subject datasets.

Our work can be expanded by adapting the POC-CSP layer in other end-to-end networks with more complex classifiers, in domains and datasets that have more training data available. We intend to show the utility of the model on other BCI tasks as the model itself is task-agnostic in nature. It will also be interesting to explore the effects of combining multiple datasets in the same domain in the pre-training stage and the models ability to generalise under that setting.

References

[1] Z. J. Koles. The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG. *Electroencephalography and Clinical Neurophysiology*, 79(6):440–447, December 1991.

[2] Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Basilio Sierra, Itziar Irigoien, Igor Rodriguez-Rodriguez, and Izaro Goienetxea. Using Common Spatial Patterns to Select Relevant Pixels for Video Activity Recognition. *Applied Sciences*, 10(22):8075, November 2020.

[3] Jonathan S. Brumberg, Jeremy D. Burnison, and Kevin M. Pitt. Using Motor Imagery to Control Brain-Computer Interfaces for Communication. In Dylan D. Schmorrow and Cali M. Fidopiastis, editors, *Foundations of Augmented Cognition: Neuroergonomics and Operational Neuroscience*, volume 9743, pages 14–25. Springer International Publishing, Cham, 2016.

[4] Hyeon Kyu Lee and Young-Seok Choi. A convolution neural networks scheme for classification of motor imagery EEG based on wavelet time-frequecy image. In *2018 International Conference on Information Networking (ICOIN)*, pages 906–909, Chiang Mai, Thailand, January 2018. IEEE.

[5] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, 8(4):441–446, December 2000.

[6] Benjamin Blankertz, Ryota Tomioka, Steven Lemm, Motoaki Kawanabe, and Klaus-robert Muller. Optimizing Spatial filters for Robust EEG Single-Trial Analysis. *IEEE Signal Processing Magazine*, 25(1):41–56, 2008.

[7] Matthias Krauledat, Michael Tangermann, Benjamin Blankertz, and Klaus-Robert Müller. Towards Zero Training for Brain-Computer Interfacing. *PLoS ONE*, 3(8):e2967, August 2008.

[8] F Lotte, M Congedo, A Lécuyer, F Lamarche, and B Arnaldi. A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, 4(2):R1–R13, June 2007.

[9] Wei Wu, Xiaorong Gao, and Shangkai Gao. One-Versus-the-Rest(OVR) Algorithm: An Extension of Common Spatial Patterns(CSP) Algorithm to Multi-class Case. 2018.

[10] M. Grosse-Wentrup, C. Liefhold, K. Gramann, and M. Buss. Beamforming in Noninvasive Brain–Computer Interfaces. *IEEE Transactions on Biomedical Engineering*, 56(4):1209–1219, April 2009.

[11] Maryanovsky, Daniel, Mousavi, Mahta, Moreno, Nathanial, and Sa, Virginia de. Csp-Nn: A Convolutional Neural Network Implementation Of Common Spatial Patterns.

[12] Jinyu Li. Recent Advances in End-to-End Automatic Speech Recognition. 2021.

[13] Rohit Prabhavalkar, Takaaki Hori, Tara N. Sainath, Ralf Schlüter, and Shinji Watanabe. End-to-End Speech Recognition: A Survey. 2023.

[14] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pages 369–376, Pittsburgh, Pennsylvania, 2006. ACM Press.

[15] Michael Tangermann, Klaus-Robert Müller, Ad Aertsen, Niels Birbaumer, Christoph Braun, Clemens Brunner, Robert Leeb, Carsten Mehring, Kai J. Miller, Gernot R. Müller-Putz, Guido Nolte, Gert Pfurtscheller, Hubert Preissl, Gerwin Schalk, Alois Schlögl, Carmen Vidaurre, Stephan Waldert, and Benjamin Blankertz. Review of the BCI Competition IV. *Frontiers in Neuroscience*, 6, 2012.

[16] Introduction to Statistical Pattern Recognition - 2nd Edition.

[17] Mario Lezcano-Casado and David Martínez-Rubio. Cheap Orthogonal Constraints in Neural Networks: A Simple Parametrization of the Orthogonal and Unitary Group, May 2019. arXiv:1901.08428 [cs, stat].

[18] Charles C. Pinter. *A book of abstract algebra*. Dover Publications, Mineola, N.Y, dover ed edition, 2010.

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