3-23-2023 10:00 AM

Multi-view Contrastive Learning for Unsupervised Domain Adaptation in Brain-Computer Interfaces

Sepehr Asgarian, *The University of Western Ontario*

Supervisor: Boyu Wang, *The University of Western Ontario*
Co-Supervisor: Yalda Mohsenzadeh, *The University of Western Ontario*
A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Recommended Citation

Asgarian, Sepehr, "Multi-view Contrastive Learning for Unsupervised Domain Adaptation in Brain-Computer Interfaces" (2023). *Electronic Thesis and Dissertation Repository*. 9236.
https://ir.lib.uwo.ca/etd/9236

Abstract

Electroencephalography (EEG) has been widely used to record electromagnetic fields for motor imagery (MI)-based brain-computer interfaces (BCIs). However, collecting MI signals is often time-consuming and challenging to classify due to the inter-subject variability of EEG signals. To address these issues, we propose a novel framework MACNet, which stands for Multi-view Adversarial Contrastive Network. MACNet employs a contrastive learning approach to learn spatial and temporal features in two views, using Riemannian and Euclidean encoders. By jointly extracting underlying features and learning domain-invariant representations in both source and target features, MACNet improves the alignment and accuracy. In addition, we propose a domain mixup for the BCI field at the signal and embedding levels, to improve domain alignment. We evaluate MACNet on two public datasets and demonstrate that it outperforms all previous methods in inter-subject transfer learning. Specifically, MACNet achieves 83.79% accuracy for the BCI Competition IV dataset and 80.00% accuracy for the OpenBMI dataset.

Keywords: EEG, Multi-View Contrastive, Riemannian Neural Network, Unsupervised Domain Adaptation, Neural network, Transfer learning

Summary for Lay Audience

A Brain-Computer Interface (BCI) allows users to control external devices using their brain activity. Motor Imagery (MI) is one of the primary forms of brain-computer interfaces (BCIs) in which the participant is asked to imagine moving different parts of the body. The inter-subject variability of brain signals, which leads to a domain shift between sessions, is a significant issue in the practical use of MI systems. As a result, machine learning models trained on one session perform poorly on the other session since the data distribution varies from what they have learned. This thesis introduced a novel machine-learning model specifically for MI tasks to build a robust data-driven method that extracts and learns domain-invariant features in different views of EEG signals.

Acknowledgements

It's difficult to navigate academic life, especially when you start your research role during the pandemic and you're far from home. I could not have finished my master's degree without the great assistance I got during my master's. As a result, I want to thank everyone who has inspired me and helped me succeed by dedicating this part of my thesis to them.

I would like to express my deepest gratitude to my supervisors, Dr. Boyu Wang and Dr. Yalda Mohsenzadeh, for their constant guidance and support. They taught me many skills, especially how to develop and follow my interests in my research studies. It was an absolute honour to work under their supervision.

Furthermore, I would like to thank the Brain and Mind Institute, BrainsCAN, and Computer Science departments for allowing me to conduct my research and investigate significant research problems. Starting a new life far from home is challenging, and I am grateful for all my friends and lab members for their unforgettable help during the hardships.

Most importantly, I am grateful for my family's help in getting me to a point where my ego can lead the way from then on.

Contents

List of Figures

List of Tables

Table 1: Table of Notation

X	Feature space
Y	Label space
S	Source domain
T	Target domain
X^s	Source feature
Y^s	Source labels
X^t	Target feature
h_r^s	Source Riemannian embedding
h_r^t	Target Riemannian embedding
h_e^s	Source Euclidean embedding
h_e^t	Target Euclidean embedding
x_e^m	Mixup of the first level
h_e^{m2}	Mixup of the second level
f_{Bi}	BiMap layer
f_R	ReEig layer
$\mathfrak{G}$	Riemannian barycenter
f_L	LogEig layer
f_{RBN}	Riemannian batch normalization
f_d	Dropout layer
τ	Temperature parameter
λ	Mixup ratio
δ	Riemannian distance
L_{ce}	Cross entropy loss
L_d	Loss domain
L_{scon}^s	Source contrastive loss
L_{scon}^t	Target contrastive loss
L_{MACNet}	MACNet loss

Chapter 1

Introduction

A brain-computer interface (BCI) is a system that builds a communication channel to help people control external devices by using their brain activities. This procedure begins with the recording of the user's brain activity and continues with signal processing to determine the user's intent. Moreover, BCI can also be used to detect and identify potential brain defects [68, 70]. Among various methods of brain signals, cortex electroencephalography (EEG) [32] gains its population due to its low-cost and non-invasive nature. EEG records brain signals through electrodes placed on specific cortex areas with millisecond-level high temporal resolution. Depending on the system's function, different types of EEG signals can be utilised in BCIs. One of the main types of EEG signals is Motor Imagery (MI). MI refers to the mental movement of a body part without actual action. In such an MI-based BCI system, the subject is requested to envision moving different portions of their body in order to stimulate neuronal activity in certain brain regions associated with the imagined movements. For instance, one of the popular tasks is a left, right hand MI [78]. which may be utilised to instinctively complete tasks with similar lateral characteristics.

There are several applications for BCI systems based on EEG in the real world setting. For instance, by learning neural patterns when emotionally charged information is consumed by EEG signals, BCI can help in identifying a user's affective state, such as being sad, happy, or neutral [110]. Moreover, by imagining movements such as left or right, users can move virtual objects on screen [14]. In this regard, several approaches have been implemented for learning task-relevant discriminative characteristics of EEG data, such as statistical [36] and machine learning models [3, 20, 81, 87]. Although, many BCI applications perform well in the laboratory, it is difficult to create these applications appropriate for real-world settings. There are three main challenges in this field:

First, gathering BCI data is time-consuming since recording EEG signals requires long recording sessions. Additionally, labelling them is difficult. For instance, while collecting and labelling image data, we may take images from various perspectives and lighting conditions, and we can classify the objects after taking the picture for object detection. However, while collecting a BCI dataset, the data must be labelled while the subject executes the activity. It is difficult to determine which class a subject trial belongs to based on the raw signals alone. Consequently, we have limited data available in this field, and most classification methods in machine learning require large training datasets; thus, the number of samples in this field is challenging.

Second, due to the low signal-to-noise ratio (SNR) of brain electrical activity and poor spatial resolution for volume conduction [11], EEG signals are non-stationary within a subject and vary between sessions [62]. This will result in changes in the order of minutes to days, which poses a problem for the long-term dependability of BCIs. In addition, non-stationarities may arise due to changes in mental strategy, mental state, and physical properties of the electrodes in EEG experiments.

Third, EEG signals contain lots of noise and outliers. This is due to the high complexity of brain signals and noisy sensors since they are extremely sensitive to detect even the small potential changes across sessions while recording the experiments. Hence artefacts from non-brain activity sources, such as movement or blinking, are detected in the experiments. Moreover, noise and outliers will negatively impact learning algorithms.

As a result, it is challenging to develop an appropriate, robust machine learning model for different subjects, and sessions [90, 102].

The dimension of raw EEG is large compared to the number of trials. As a result, nonlinearities can be found in high-dimensional multichannel EEG signals, and existing methods are often incapable of learning these nonlinearities. Hence, the majaority of researches in these fields suggest pipelines for BCIs. Therefore, a feature extraction step is applied before feeding the signals to a classifier to extract discriminant features corresponding to the user's different underlying neurological mechanisms.

One of the most effective traditional pre-processing methods for feature extraction in the MI paradigm is the common spatial pattern (CSP) [82] and its variants [86], [61]. Moreover, for binary classification problems (right hand vs. left hand), CSP aims to develop optimal spatial filters such that the variance of spatially filtered signals is maximized for one class and minimized for another class [11]. While employing CSP as the feature extraction approach, several classifiers such as Support Vector Machine (SVM), Naive Bayes (NB), and Linear Discriminant Analysis (LDA) have achieved promising results [26, 29, 37, 96]. However, CSP can easily get distorted by artifacts and outliers [11] and are specific to subjects and sessions.

Due to the advancement of machine learning methods in recent years in many fields, such as computer vision [24,79,95,98], natural language processing [13,31,58], timeseries [23,104] and health systems [22], recently researchers have proposed novel Deep Neural Networks (DNN) methods in BCI fields [3, 20, 108]. Many DNN methods in this field are based on convolutional neural networks (CNN) along with temporal and spatial filters [49, 50, 87]. In comparison to traditional methods, CNN is making significant advances in its ability to represent patterns in brain topography. Lawhern et al. [50] proposed EEGNet, a compact CNN including deathwise and separable convolution incorporating well-known EEG feature extraction. EEGNet achieved high results in Event Related Potential (ERP) and Oscillatory-based BCIs in a variety of BCI paradigms, for both including the MI, P300, and Steady State Visually Evoked Potential (SSVEP). However, using just CNN for feature extraction in BCI tasks is ineffective since the underlying structures of images and signals are different. In particular, for CNN to efficiently extract local features, there are assumptions in computer vision that the structure of input image data be stationary, translation invariant, and stable with respect to local deformations, which is conceptually described as the Euclidean nature [52]. Moreover, most of the DNN methods implemented in this field extract features only in Euclidean space. However, according to electrophysiological and dynamical neuroscience studies, the structure of EEG signals may not be in a Euclidean space; In addition, due to the non-stationary structure of EEG signals, the local statistics differ from the location's [42]. Hence, using DNN implemented in Euclidean space cannot capture all the significant features, such as spatial information, and another view of EEG signals is needed to capture these information.

In addition to the Euclidean view, Riemannian approaches that rely on symmetric positive definite (SPD) matrices represent another view of EEG signals in Riemannian space. In BCI classification algorithms, Riemannian approaches have achieved robust results [19, 100, 105]. They have also shown their superiority in many BCI applications, such as sleep/respiratory state classification [41] and EEG pattern decoding [69]. Riemannian learning methods have the capacity to learn suitable statistical representations, specifically the spatial information underlying SPD manifolds [39]. Recently, a novel Riemannian-based network [40] has been proposed that uses SPD matrices and can capture the spatial information of the input data.

Although DNN and Riemannian models have been tackled with high accuracy, due to the covariate shift, the distribution of the sessions and subjects is different [11]. This will result in the poorer performance of models when transferred to other sessions or subjects. Unsupervised Domain Adaptation (UDA) is a useful approach that can enhance the target domain's performance by reducing the distribution discrepancy between the source and the target domain. UDA tries to efficiently transfer the knowledge learned in the supervised domain to its unsupervised counterpart. Moreover, this approach has been intensively studied in computer vision tasks [27, 46, 57, 59, 92]. Adversarial domain adaption are one of the main groups of UDA methods, which began with the gradient reverse paper [93].

1.1 Motivation

Machine learning has shown great potential for improving our understanding of BCIs and unlocking their full potential for various applications. However, due to the distribution shift between training and test data, traditional machine-learning approaches are limited in their ability to achieve high accuracy when dealing with inter-subject problems in MI data. This thesis aims to address the domain shift problem in MI data from different sessions.

Adversarial domain adaptation is a popular method used to address domain shift problems. Its goal is to achieve domain-level alignment between the source and target domains by reducing the distribution discrepancy between them. While this approach may improve the model's accuracy by extracting more generalized features applicable across domains, it is not guaranteed to obtain high performance on the target domain. Even if the domain discriminator is fully confused, the supervised classifier may still perform poorly on the target domain due to the importance of domain-specific features. To address the domain shift problem in MI data, we propose a method that captures significant features specific to the MI field within both source and target domains. Specifically, our proposed method uses two views of the data (EEG signal and SPD metrics) to capture more comprehensive representations. In particular, our mode captures spatial and temporal features jointly for both the source and target data, which improves the model's ability to learn discriminative and invariant representations across different sessions.

Additionally, since we assume that we do not have any labeled data in the target domain, our proposed method utilizes a contrastive learning approach to encourage the model to learn representations that group data points with similar labels closer together. This approach makes the job of the classifier easier, as it is able to identify patterns more easily when the data points with the same label are closer together in the learned feature space.

To accomplish this, we incorporate adversarial training, with two levels of domain mixup to get the domain invariant features, and multi-view contrastive learning to further extract spatial and temporal features simultaneously in a self-supervised way for both sources and target domains.

1.2 Contributions

In this thesis, different from existing works in the BCI field, our machine learning model aims to capture the features of MI data in different views in both source and target domain, which is invariant between different domains. To achieve this, in addition to the adversarial task, we design a contrastive task that learns feature embedding of the same EEG signal with different views in both source and target domain. In this task, the representation of the different views of the same signal gets closer, and the views of the other EEG signals get farther apart.

We further investigate the performance of our model on the two benchmark datasets, OpenBMI [53], and BCI Competition IV [14] dataset. In general, this thesis has the following contributions:

1. Multi-view adversarial contrastive network (MACNet) is proposed for MI signals. Our method learns domain-invariant representations of MI data from two views, incorporating spatial and temporal information, resulting in a more robust representation learning.

2. Multi-level mixup is proposed at both the signal and embedding levels with soft domain labels, enhancing the feature extractor's generalization ability.

3. Extensive adaption experiments on two datasets have been utilized to illustrate the superior performance of our method.

1.3 Thesis Outline

The first chapter of this thesis presents the problem's definition, challenges, and the suggested solution's scope. In the second chapter, the fundamental ideas, such as the various Euclidean and Riemannian learning methods employed in this project, are addressed, as well as past research in this field. Further, the design and implementation of the suggested approach are detailed in the third chapter. The fourth chapter discusses the evaluation and analysis of the results, while the fifth chapter provides a summary, a conclusion, and suggestions for further research.

Chapter 2

Literature Review

2.1 Electroencephalography & MI

The first study on EEG in humans published in 1929 by Hans Berger [34] which shows that researching and analyzing EEG signals is an old technique. EEG is a non-invasive method that measures electric brain activity using electrodes placed on the head. To be exact, the electrodes are able to record changes in postsynaptic potential caused by neurotransmitter release between a presynaptic axon and a postsynaptic dendrite. The electrode cap (see Figure 2.1) is used to enhance scalp-electrode conductivity to measure the voltage changes in the soma. Moreover, in usual EEG experiments, conductive gel is added between the electrode and the scalp to improve the connection.

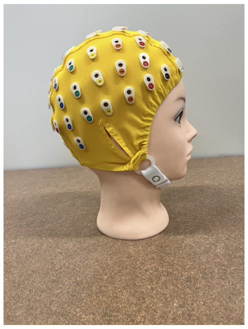


Figure 2.1: Electrode cap

EEG signals are often employed in BCIs due to their low cost, high temporal resolution, and relative simplicity. MI is one of the main categories of BCI experiments that are done by collecting EEG signals from subjects. MI is a cognitive process in which a person imagines completing a movement without actually performing the action and contracting the muscle. There are primarily two varieties of MI, namely visual and kinesthetic. Visual MI is a visualization of the movement matching the visual network. In addition, kinesthetic MI is the ability to imagine performing a movement by having a mental image of the muscle contraction and sensation during an actual movement. It has been demonstrated that the mental visualisation of motor activity, such as the execution of a hand movement, may induce brain activation identical to that of the actual event. There are several benefits to MI. Numerous studies have demonstrated that neural plasticity and brain activity during MI training are comparable to those observed during actual MI practice. Second, there is no restriction on a patient's ability to perform motions. Lastly, the effects of MI training are impacted by the contextual setting and the emotional state of the individuals.

2.2 Euclidean learning

2.2.1 Neural Network

Artificial Neural Networks (ANNs) are biologically inspired as they are composed of computational components that behave in a manner comparable to the fundamental functions of a biological neuron. Artificial neurons in ANN are organised into several layers; this is comparable to the human visual cortex, which consists of a collection of hierarchically organised brain regions. As in a biological neural network, the connection and signal transmission between neurons and layers enable the ANN system to convert the particular input signal into relevant outputs, also known as the prediction. ANN is capable of predicting output based on any given input when the mathematical connection between input and output is nonlinear, complicated, etc. The structure of an ANN comprises an input layer, a hidden layer, and an output layer. As it can be seen in Figure 2.2 input neurons receiving input data compose the input layer.

Additionally, the hidden layer contains neurons that receive input neurons and transmit them to the output layer. Finally, the output layer is composed of neurons that reflect the expected output parameters.

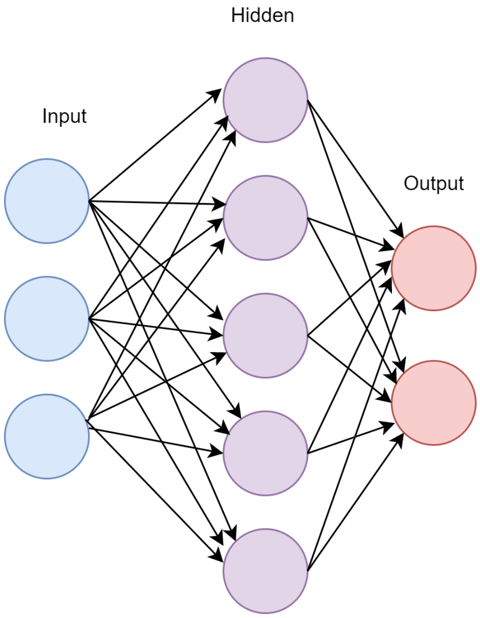


Figure 2.2: Artificial Neural Networks

2.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are a category of neural network. In practice, they are typically employed for the classification of images and other grid-like data. CNN is first introduced by LeCun et al. [51]. There are different layers in CNN's:

Convolutional Layers

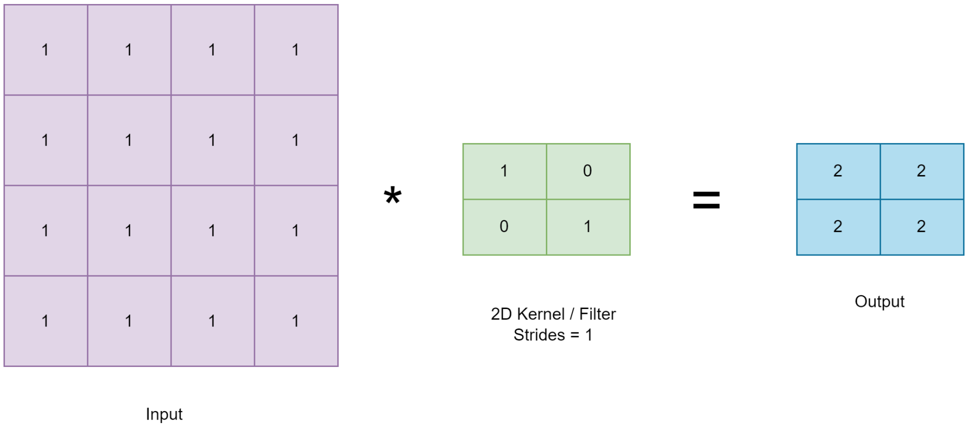
The convolutional layer is the first layer of CNN that includes three main components: input, filters (also known as kernels), and output. It performs element-by-element multiplication with the part of the input that it covers in each stage, and in the next step, it sums their outputs for each sliding operation. The convolution result is then provided to the activation function or feature map. The mathematical definition of convolution is as follows:

$$(f * g)\,(i) = \sum_{m}^{M} g(m) \cdot f(i - m) \tag{2.1}$$

where f represents a one-dimensional input, g represents a one-dimensional kernel, and M represents the size of the input. Moreover, N is the length of the input, and for each $i \in N$, this Equation is computed. The convolution technique can be expanded to higher dimensions; For instance, this method can be expanded to the 2D version by convolving along two axes simultaneously:

$$(f * g)\,(i, j) = \sum_{m}^{M} \sum_{k}^{K} g(m, n) \cdot f(i - m, j - k) \tag{2.2}$$

In addition, one of the useful components of the convolutional layer is the stride. Stride is a neural network filter parameter that affects the amount of movement across input data.



$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

Figure 2.3: Two-dimensional convolution kernel is applied to an input matrix

Pooling Layers

Pooling is a technique for significantly reducing the size of intermediate feature layers. This approach minimizes the amount of network calculations and eliminates overfitting. Similar to convolution layers, pooling slides the filter according to its stride along the input. However, unlike convolution layers, pooling does not include weights or learnable parameters; rather, it computes operations such as the input's maximum, average, and minimum in various regions.

Fully Connected Layers

A fully connected neural network consists of a sequence of layers in which each neuron in one layer is linked to each neuron in the other layer.

2.2.3 EEGNet

EEGNet is a CNN architecture that is adaptable to a variety of BCI paradigms and applications. The authors in their paper proposed and demonstrated the generalizability of a CNN architecture based on different depth-wise convolution layers. Their comparisons with cutting-edge models and approaches suggest that their design has the potential to produce superior outcomes. The 3-layer CNN architecture includes convolution, depth-wise, and separable layers, as well as average-pooling, batch normalization (BN), and dropout layers (see Figure 2.4). Dropout is an effective strategy for preventing overfitting in deep learning. In backpropagation, dropout algorithms ignore neurons that do not contribute. In addition, they have evaluated their methods on many BCI datasets, including MI, SSVEP, and ERP.

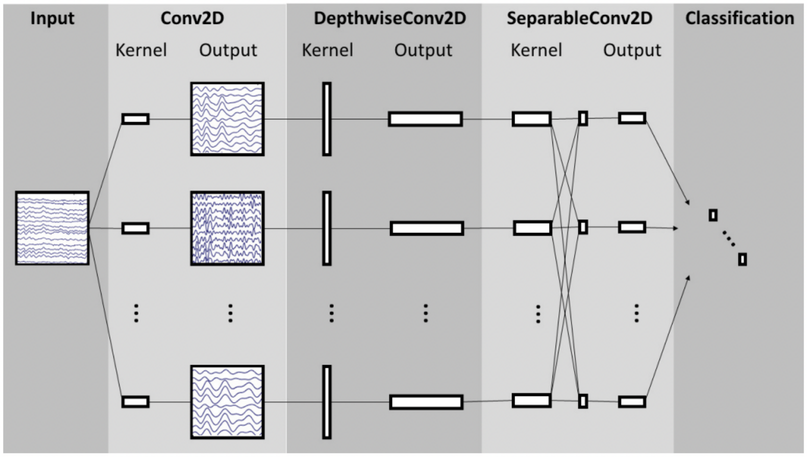


Figure 2.4: Schematic of EEGNet [50]

2.2.4 ShallowConvNet

In comparison to more time-consuming models, ShallowConvNet has shown its superior performance in the subject-dependent MI classification [87]. The first two layers of ShallowConvNet are CNNs for detecting temporal and spatial features. Followed by the first two CNN layers, there is BN, square activation, average pooling, and log activation. Then there is a flatten followed by a dropout layer. The overview of ShallowConvNet can be seen in Figure 2.5.

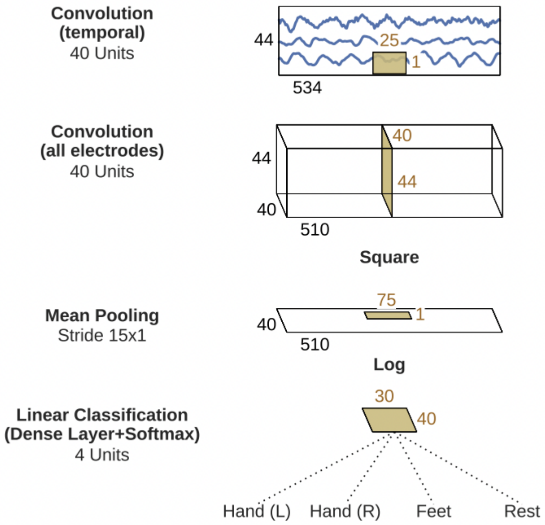


Figure 2.5: Schematic of ShallowConvNet [87]

2.3 Riemannian learning

2.3.1 Covariance Matrices

EEG signals $X_i(t) = [x(t), \ldots, x(t+T-1)] \in \mathbb{R}^{n \times T}$ are recorded with multiple channels and sampled in discrete-time points, where n and T denote the number of channels and samples. Here $x(t)$ denotes the EEG signal vector at a specific time point t. Riemannian approaches are often applied to the Sample Covariance Matrix (SCM) denoted as P_i:

$$P_i = \frac{1}{T_s - 1} X_i X_i^T \tag{2.3}$$

where X_i represents a i-th trial with T_s samples.

2.3.2 Riemannian Manifolds

A topological manifold can be defined as a space in which every point has a neighborhood homeomorphic to $\mathbb{R}^n$. When the topological manifolds are endowed with a differential atlas, they become differential manifolds, where the transitions between mapping manifolds are smooth. In a differential manifold, translating a point can be approximated as its linear transformation. Then, a Riemannian manifold $\mathcal{P}(n)$ is a real smooth manifold. Assume $\mathcal{P}(n)$ be the set of $n \times n$ SPD matrices, which is defined as:

$$\mathcal{P}(n) = \left\{ C \in \mathbb{R}^{n \times n} \mid C^T = C, \boldsymbol{x}^T C \boldsymbol{x} > 0, \forall \boldsymbol{x} \in \mathbb{R}^n \right\} \tag{2.4}$$

Matrices in $\mathcal{P}(n)$ lie in a differentiable Riemannian manifold $\mathcal{M}$ [25], where the tangent space $T_C\mathcal{P}(n)$ of each point has a metric defining the inner product that changes smoothly along with its elements [1].

2.3.3 Riemannian Metrics

Equipping with the inner product, it is possible to compute the length of curves in this manifold. In this thesis, we utilise the affine-invariant Riemannian metric (AIRM) [76], which defines a Riemannian distance $\delta_{\Re}(P_1, P_2)$ between two points, P_1 and P_2, as:

$$\delta_{\Re}(P_1, P_2) = \frac{1}{2} \left\| \log \left(P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}} \right) \right\|_F \tag{2.5}$$

The notion of the mean of a set of SPD matrices $\frac{1}{N} \sum_{i \leq N} P_i$ can also be extended to the Riemannian manifold. In addition, the Riemannian barycenter, also known as the Fréchet mean [103], is introduced. The Riemannian barycenter has a practical advantage in representing geometrical features in Riemannian manifold analysis [77]:

$$\mathrm{Bar}_{\boldsymbol{w}}\left(\{P_i\}_{i \leq N}\right) := \arg\min_{G \in \mathcal{M}} \sum_{i=1}^{N} w_i \delta_{\mathfrak{R}}^2 \left(G, P_i\right) \tag{2.6}$$

In the formula 2.6 a coefficient w_i is associated with each point $\delta_{\mathfrak{R}}$ of the considered set, with $\sum_i w_i = 1$.

2.3.4 SPD Network

Riemannian Neural Network, which defines as an SPD matrix network (SPDNet), consists of four typical kinds of layers analogous to the well-known CNN, including bilinear mapping (BiMap), eigenvalue rectification (ReEig), log eigenvalue (LogEig), and BN layer. To be more specific, the BiMap layer maps the input SPD matrices X_{k-1} into X_k which is a more compact matrices. The k-th BiMap layer $f_{Bi}^{(k)}$ can be defined as follows.

$$\boldsymbol{X}_k = f_{Bi}^{(k)}\left(\boldsymbol{X}_{k-1}; \boldsymbol{W}_k\right) = \boldsymbol{W}_k \boldsymbol{X}_{k-1} \boldsymbol{W}_k^T \tag{2.7}$$

The k-th ReEig layer defined as $f_R^{(k)}$ is a non-linear function that tunes up the small positive eigenvalues of the input matrices by a rectification threshold.

$$\begin{aligned} \boldsymbol{X}_k = f_r^{(k)}\left(\boldsymbol{X}_{k-1}\right) = \; & f_r^{(k)}(\boldsymbol{U}_{k-1}\boldsymbol{\Sigma}_{k-1}\boldsymbol{U}_{k-1}^T) \\ = \; & \boldsymbol{U}_{k-1} \max\left(\epsilon \boldsymbol{I}, \boldsymbol{\Sigma}_{k-1}\right) \boldsymbol{U}_{k-1}^T \end{aligned} \tag{2.8}$$

The Riemannian batch normalization layer (RBN) normalizes training data in every batch to avoid the gradient becoming large or small during back-propagation. However, since all SPD matrices are in the Riemannian manifold, this layer has to be in the same manifold. RBN for SPD matrices is implemented by Brook et al. [12], which uses batch centering and biasing. The Riemannian barycenter $\mathfrak{G}$ and Biasing are calculated from Equation 2.9 and 2.10. Centering from $\mathfrak{G} := \mathrm{Bar}(\mathcal{B})$:

$$\forall i \leq N, \bar{P}_i = \Gamma_{\mathfrak{G} \to I_d}\left(P_i\right) = \mathfrak{G}^{-\frac{1}{2}} P_i \mathfrak{G}^{-\frac{1}{2}} \tag{2.9}$$

Biasing towards parameter G :

$$\forall i \leq N, \tilde{P}_i = \Gamma_{I_d \to G}\left(\bar{P}_i\right) = G^{\frac{1}{2}} \bar{P}_i G^{\frac{1}{2}} \tag{2.10}$$

The LogEig Layer f_L is designed to map the SPD matrices lying in Riemannian geometry into a corresponding structure which is on Euclidean space, thus the flatten layer can be applied into output of LogEig layer [7]:

$$\begin{aligned} \boldsymbol{X}_k &= f_l^{(k)}\left(\boldsymbol{X}_{k-1}\right) = \log\left(\boldsymbol{X}_{k-1}\right) \\ &= f_l^{(k)}(\boldsymbol{U}_{k-1}\boldsymbol{\Sigma}_{k-1}\boldsymbol{U}_{k-1}^T) \\ &= \boldsymbol{U}_{k-1}\log\left(\boldsymbol{\Sigma}_{k-1}\right)\boldsymbol{U}_{k-1}^T \end{aligned} \tag{2.11}$$

The overall architecture of SPDNet can be seen in Figure 2.6.

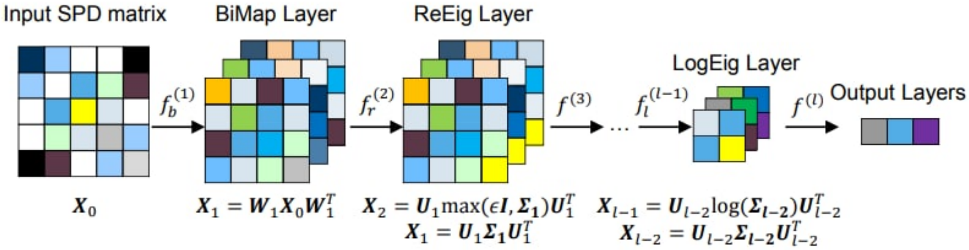

Input SPD matrix
BiMap Layer
ReEig Layer
LogEig Layer
Output Layers
$f_b^{(1)}$
$f_r^{(2)}$
$f^{(3)}$
...
$f_l^{(l-1)}$
$f^{(l)}$
$\boldsymbol{X}_0$
$\boldsymbol{X}_1 = \boldsymbol{W}_1\boldsymbol{X}_0\boldsymbol{W}_1^T$
$\boldsymbol{X}_2 = \boldsymbol{U}_1\max(\epsilon\boldsymbol{I}, \boldsymbol{\Sigma}_1)\boldsymbol{U}_1^T$
$\boldsymbol{X}_1 = \boldsymbol{U}_1\boldsymbol{\Sigma}_1\boldsymbol{U}_1^T$
$\boldsymbol{X}_{l-1} = \boldsymbol{U}_{l-2}\log(\boldsymbol{\Sigma}_{l-2})\boldsymbol{U}_{l-2}^T$
$\boldsymbol{X}_{l-2} = \boldsymbol{U}_{l-2}\boldsymbol{\Sigma}_{l-2}\boldsymbol{U}_{l-2}^T$

Figure 2.6: SPDNet framework proposed by huang et al. [39]

2.4 Supervised Learning

Supervised learning is the most common learning strategy in machine learning when the task has a large amount of labelled data. As its name implies, supervised learning depends on a teacher to penalise the model for producing incorrect results. This method can be compared to the situation in which students learn about the course via their teachers' direct comments.

For instance, a teacher would clarify a student's incorrect answer in the exam, and from the feedback student will learn.

Supervised approaches in machine learning train networks to map input data to a corresponding label. The assumption of this approach is that the model for the designed task contains effective representation at the embedding level of the hidden layers. Given an N-sample set of inputs and outputs for training $\{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, supervised approaches estimate a function $\{(f_\theta : X \rightarrow Y)\}$ that maximises the similarity between the true label y_i and the output forecasted by the function $f_\theta(x)$. In this approach, the set of parameters that have influenced the loss function's output is represented by θ. In addition, gradient descent is employed during training to optimize the parameters θ to minimise a cost function. To evaluate the model function f, the cost function $L(\theta)$ or cross-entropy loss is used to calculate the distance between a given point and the target. Therefore, the more the value of the cost function $L(\theta)$, the more error the cost function has:

$$L(\theta) = -\sum_{c=1}^{M} \mathbf{y}_{o,c} \log\left(\mathbf{p}_{o,c}\right) \tag{2.12}$$

In the Equation above, M represents the number of classes, whereas p is the predicted probability of classifying observation o into class c.

Nevertheless, a common assumption is made by the majority of supervised learning approaches that the training and test data are selected from the same distribution. However, due to the domain shifts [63] this assumption is not always correct, and a classifier trained on the source domain will likely see a performance loss when evaluated on the target domain. In the next section, unsupervised approaches, specifically UDA methods, will be introduced that solve this problem.

2.5 Unsupervised Learning

As discussed previously in the last section, in real-world problems, there is not much label data available for training. This can be due to the fact that gathering data is costly and time-

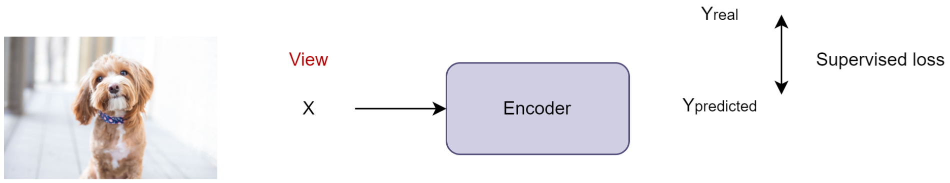


Figure 2.7: Supervised learning

consuming. Furthermore, labelling data for specific tasks such as EEG experiments necessitates the use of experts. On the other hand, there is a vast amount of unlabeled data available, such as texts, images, signals, and others. As a result, it is not advantageous not to use these data. In the past, traditional machine learning methods, such as principal component analysis and clustering algorithms, were used to discover data groupings and hidden patterns in unsupervised data. However, in recent years, DNN algorithms have also made huge progress in solving unsupervised problems. These approaches mainly use downstream tasks after learning useful representations from unlabeled data.

2.5.1 Contrastive Learning

Recently, due to the increase of unlabeled data, self-supervised learning, including representation learning based on contrastive methods, has been widely studied and shown their effectiveness in many applications [2, 67, 85]. In unsupervised contrastive learning, representations of similar samples (positives) are represented close together; in contrast, dissimilar samples (negatives) are represented far apart [15]. Different versions of contrastive learning have been developed so far [15, 17, 33, 73]. SimCLR is one of the successful methods in this category. In this method, an augmented form of the original sample is treated as a positive sample, while the rest of the samples in the batch are treated as negative ones. In this method, a key contribution to making similar samples is data augmentation approaches such as cropping a portion of the data, resizing, adding shifts, and adding gaussian noise. For instance, as can be seen in Figure 2.8, the original data is a dog and a chair. In the first step, data augmentation is applied to the crop and colour distortion of each image. In the next step, the representation of each image is created by passing the augmented pairs into convolutional encoders. Finally, the representations are passed to the projection head, which includes multi-layer perceptrons. Then the loss function between positive and negative pairs is computed. The Equation 2.13 represents the contrastive loss function:

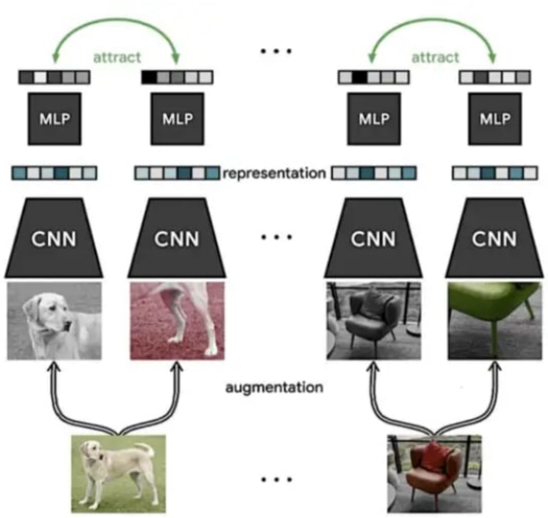


Figure 2.8: Similar images are attracted to each other [15]

$$L_{scon} = -\ \log \frac{\exp\Big(\ sim(z_i, z_j)\ /\tau\ \Big)}{\sum_{k=1\ [k\neq i]}^{2N} \exp\big(\ sim(z_i, z_k)\ /\tau\ \big)} \tag{2.13}$$

Where z_i and z_j are the representations of the augmentation's after projection layer, and τ signifies a temperature parameter. Contrastive loss is calculated over all positive pairings (i, j) and (j, i).

Moreover, The goal of this network is to minimize the distance of two augmented representations of the same object close to each other and simultaneously maximize the distance of the other objects to be far apart.

SimCLR has been demonstrated to outperform supervised approaches on several image classification datasets, as well as prior self-supervised methods on ImageNet [15]. However, compared to computer vision, in the BCI field, data augmentation is challenging since augmenting the EEG signal, such as by adding noise, might change the label of the data. Our approach extends SimCLR; however, we are using different views of EEG signals with different encoders, which helps the feature extractor get more useful features compared to only using one view.

2.6 Multi-view Contrastive Learning

Multi-view learning is a recent field of machine learning that focuses on enhancing generalization performance through the use of different perspectives or views of the same inputs. Multi-view data are popular in real-world applications. For instance, color and texture information are two different views for images and videos. Specifically, taking the same image from various perspectives can also consider as separate views. Data fusion and integration from several feature sets are alternative names for multi-view learning [109].

Multi-view contrastive learning has been implemented in a variety of fields, such as image, audio, graphs, and natural language processing [60, 72, 88, 91, 99, 101]. This method includes learning data representations by comparing several views of the same data to maximize the similarity between views of the same data while minimizing the similarity between views of different data [91]. The objective is to capture a common or mutual representation of the underlying structure of the data from different views. Next, same as simple contrastive methods, the representation after training can be utilized for various downstream tasks, including classification and clustering. As it can be seen in Figure 2.9 there are four different views of the same image, and one view of the other image, the model [91] tries to bring views of the same scene together, while pushing views of different scenes apart in embedding space.

One key advantage of multi-view contrastive learning is that it can leverage multiple sources of information to learn a more comprehensive representation of the data. For example, in image analysis, one view of the data might be the original image, while another view might be the same image with different camera pose. By contrasting these multiple views, the model can learn to capture both the content and the structure of the image, which can be useful for tasks such as image retrieval.

Another advantage of multi-view contrastive learning is to overcome the limitations of traditional supervised learning methods, which require large amounts of labelled data. Unsupervised learning methods, such as multi-view contrastive learning, can be used to learn data representations in the absence of labelled data, which can be especially useful in domains where labelled data is scarce or costly.

Despite these promising results, there are still several challenges associated with multi-view contrastive learning. One challenge is selecting appropriate views of the data to contrast. Different views of the same data can capture different aspects of the data, and selecting views that can capture useful feature is significant, while selecting views that are too similar can not capture different features, and will not improve the accuracy.

In summary, multi-view contrastive learning is a promising approach to self-supervised learning that has shown effectiveness in a variety of domains. By leveraging multiple views of the same data, multi-view contrastive learning can learn comprehensive representations of the underlying structure of the data, which can be useful for a variety of downstream tasks.

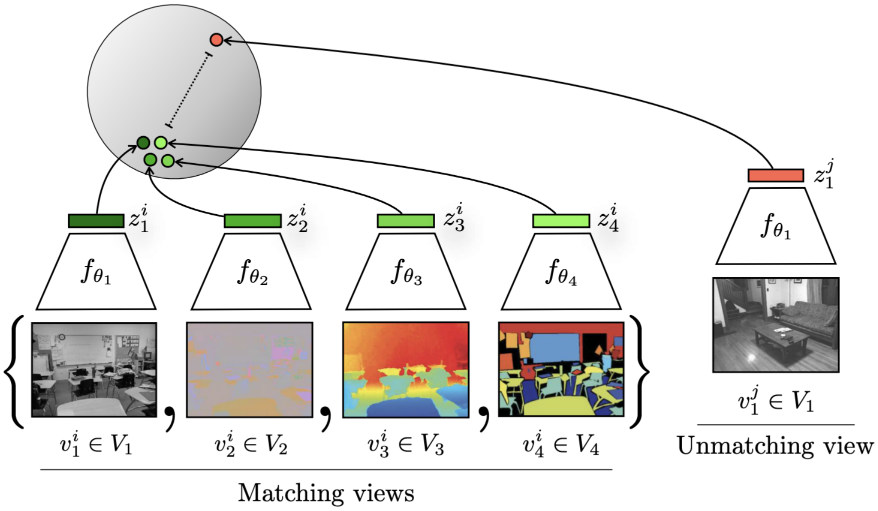


Figure 2.9: Contrastive Multiview Coding [91]

2.6.1 Unsupervised Domain Adaptation

As discussed, when training and testing data are not drawn from the same distribution, the performance of the classifier model trained on the source domain will drop when it is evaluated on the test data. Consequently, regarding this problem, for each new dataset, we should annotate the samples and then retrain the deep learning model to accommodate the new data. However, learning DNNs specifically for large datasets such as ImageNet is not beneficial since it requires a significant amount of computer resources and time. domain adaptation is a useful approach for this problem because it can reduce the need for costly labeled data in the target domain. domain adaptation has been used to either extract robust characteristics for diverse domains or adapt the features from the source domain to the target domain [84]. There are several domain adaptation approaches based on available annotations of target domain data. UDA is one of the main categories of the domain adaptation methods where it assumes that any kind of label (weak or hard) for the target domain data is entirely missing. Many of the domain adaptation methods on which this thesis focuses include adversarial training. In domain adaptation adversarial training, in addition to the label loss that has been presented in the supervised learning section of this chapter, domain loss is used for adopting different domains. The domain loss is similar to the discriminator's loss in Generative adversarial networks (GANs). Hence, to present the background for domain adaptation strategies using adversarial learning, we will first introduce GANs. GANs was introduced in 2013 by Goodfellow et al. [30]. GANs have recently received huge amounts of research in a variety of fields. Many other types of GANs are also presented after the initial research such as [16, 66, 71, 80]. The two main components of GAN's architecture are the generator and discriminator, which are both trained concurrently and compete with each other in an adversarial process in which the generator is trained to make discriminator mistakes. First, the synthetic data, similar to the real data, is generated by the generator G with noisy inputs. The generated fake data will then be used as inputs for the discriminator D in the following step. In an ideal scenario, the discriminator would be incapable of differentiating between real and synthetic data. Thus, the generator and discriminator compete against one another in a min-max game in which both models attempt to optimize Equation 2.15. In this Equation, the generator aims to reduce the loss function, while the discriminator tries to increase it. During the training phase, both the generator and the discriminator attempt to reduce their loss functions 2.16 and 2.17. In Equation 2.15, the min-max formula is introduced:

$$min_G max_D V(G, D) = E(logD(y^i)) + E(log(1 - D(G(x^i)))) \tag{2.15}$$

The Equations 2.16 and 2.17 are loss functions of the generator and discriminator.

$$G_{loss} = \frac{1}{m}\sum_{i=1}^{m}\Big(log(1 - D(G(x^i)))\Big) \tag{2.16}$$

$$D_{loss} = -\frac{1}{m}\sum_{i=1}^{m} log(D(y^i)) - \frac{1}{m}\sum_{i=1}^{m} log(1 - D(G(x^i))) \tag{2.17}$$

where E is the expected value, X^i is the generator input, and y^i is the actual target data. The discriminator's prediction is D, and the generator's prediction is G.

DA adversarial training [28] is one of the most well-known UDA methodologies. This technique aims to learn the domain-invariant features through a feature extractor. In this approach, the feature extractor, a deep CNN, extracts valuable input data characteristics. In the following stage, the fully connected layer, termed label predictor, attempts to predict the label of the provided input data. On the other hand, the domain classifier, which contains fully connected layers, tries to predict the input from which domains. In addition, the gradient reversal layer (GRL) is a layer that exists between the feature extractor and domain classifier. The GRL acts as an identity mapping function for the output of the feature extractor during forward propagation. In contrast, the GRL multiplies a gradient produced from the domain-classification error by a negative factor during backpropagation processing. In adversarial DA, both the feature extractor and the domain classifier compete against each other in a min-max game. Hence, in this technique, we attempt to mislead the domain classifier because if the domain classifier is unable to classify the actual domain, the features collected by the feature extractor are invariant across domains. Therefore, the goal of the adversarial domain adaptation is to create domain-invariant features in latent space.

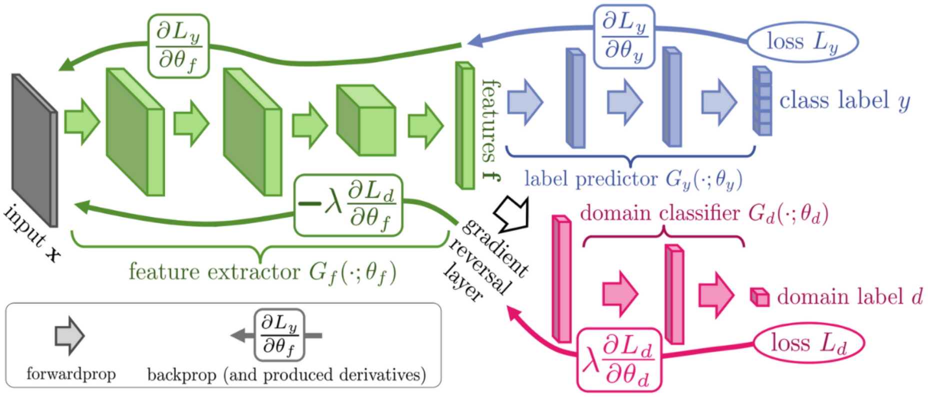

$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features $\mathbf{f}$
label predictor $G_y(\cdot;\theta_y)$
$-\lambda \frac{\partial L_d}{\partial \theta_f}$
input x
feature extractor $G_f(\cdot;\theta_f)$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
loss L_d
$\lambda \frac{\partial L_d}{\partial \theta_d}$
forwardprop
backprop (and produced derivatives)

Figure 2.10: Domain adversarial training [28]

2.7 Related Machine Learning Works

2.7.1 Traditional Methods

CSP is an effective approach for developing spatial filters that extract power information in frequency bands [10]. Additionally, there have been several successful methods to enhance the CSP algorithm. Lemm et al. introduced the Common Spatio-Spectral Pattern (CSSP), which uses a one-time-delayed sample to optimize a filter with the CSP algorithm [54]. Moreover, Kang et al. proposed Composite Common Spatial Pattern (CCSP) [48] that uses composite covariance matrices derived from a linear combination of covariance matrices. Filter Bank Common Spatial Pattern (FBCSP) is another approach based on CSP that utilises a feature selection to choose discriminative CSP features from a bank of several bandpasses and spatial filters [6]. Afterward, they used an SVM classification method to classify the extracted feature. In addition, Heung et al. proposed a novel Bayesian Spatio-Spectral Filter Optimization (BSSFO) technique for discriminative feature extraction for MI classification in an EEG-based BCI, in which the class-discriminative frequency bands and associated spatial filters are optimized using probabilistic and information-theoretic approaches [89]. Moreover, Samek et al. regularize CSP in stationary subspaces [86]. Kang et al. created a composite CSP model that used a linear combination of two approaches in order to improve classification accuracy on a small dataset [48]. Kang and Choi introduced a nonparametric Bayesian framework for multi-subject learning in which they assumed that spatial patterns across subjects share a latent subspace. In addition, they investigated the subject-to-subject relationship in their research [47].

2.7.2 Riemannian Methods

Riemannian Learning has gained popularity in a wide variety of machine learning tasks such as computer vision [39], [35] and medical fields [8, 19, 64]. A. Barachant et al. introduced the Riemannian geometry approach in the BCI field [9]. The main idea of their research is to first employ spatial covariance matrices as descriptors for EEG signals and then use Riemannian geometry for classifying the SPD matrices [9]. Rodrigues et al. presented an SPD manifold framework for extracting features using geometric operations, and they evaluated their model on eight public datasets, obtaining a high classification accuracy [83]. Li et al. worked on the p300 signals and combined the XDAWN spatial filter and SPD manifold for cross-subject classification [55]. In addition, Ju et al. proposed a geometric deep learning framework called Tensor-CSPNet [45], which captures spatial and temporo-spatio-frequency patterns using DNN methods on the SPD manifold.

2.7.3 DNN Methods

In recent years, a much larger volume of work on the application of DNNs in BCI has been published compared to traditional methods explained in the last section. Schirrmeister et al. proposed ConvNet [87] which is one of the popular methods in BCIs. ConvNet includes four CNN layers with 40 kernels per layer, followed by fully connected layers, with average pooling between the CNN layers and Relu activation. Huang et al. introduced Filter-Bank Convolutional Network (FBCNet) [39], that extracts spatially features with spatial filtering applied in the preprocessing step. Moreover, CNNs and fully connected layers are followed for the classification. Zhang et al. introduced EEG Riemannian network [107], they have extracted both spatial and temporal information by Riemannian manifolds and CNNs, and in the next step, a fusion technique is employed that learns attention weights for both embedding features.

2.7.4 Transfer Learning Methods

As described before, collecting EEG data may be challenging due to its randomness, and non-stationarity [43]. In recent years, transfer learning has gained attention for improving the performance and accuracy of BCI classification. Dai et al. proposed a transfer kernel CSP framework for learning a domain-invariant by matching source and target subjects' distributions [21]. Jeon et al. proposed the DNN Transfer model [44] utilises a DNN for transfer learning. Their model, first based on the resting-state EEG power spectral density, chooses a subject whose characteristics of the EEG signal are similar to the target subject, then it uses transfer learning for the target domain. Furthermore, Park et al. [74] improved the filter bank methods using regularization rather than extracting common features. Next, they utilised a feature selection algorithm to find discriminative features. Wang et al. proposed a prototype-based SPD matrix network for domain adaptation [97] on emotion recognition EEG. In their study, they calculated the geometric mean by combining the Riemannian metric and prototype loss. Li et al. proposed an adversarial domain adaptation method that uses association reinforcement in the last layers to adapt the conditional distributions [56]. Moroever, Hong et al. proposed dynamic joint domain adaptation network (DJDAN) [38] that is also adversarial domain adaptation that uses a global discriminator and the local discriminator for alignment.

Chapter 3

Methodology

3.1 Data Preparation

In our experiment, we use two different datasets from the MI paradigm: OpenBMI [53] and BCI Competition IV [14]. The details of each dataset are as follows:

3.1.1 OpenBMI

Provided by Korea University, OpenBMI is the largest publicly available MI dataset to date. Overall, the dataset includes 54 subjects and two sessions for each subject. As a result, it is an ideal dataset for testing our proposed algorithm. The EEG signals in this dataset are recorded at a sampling rate of 1,000 Hz and collected with 62 Ag/AgCl electrodes. The exact location and arrangement of electrodes used are shown in Figure 3.1. The MI paradigm is designed in accordance with the recognised system protocol [18]. For all blocks, the first three seconds of each trial start with a black fixed cross that appears in the center of the monitor to prepare the subject for the MI task. Later, when the right arrow or left arrow appeared as a visual cue, the subject performed the image grasping task with an instructed hand for four seconds. After each task, the screen remains blank for six seconds (±1.5 seconds). The experiment includes training and testing phases. In each stage, 100 experiments on the left and right image tasks are carried out. In the online test phase, according to the real-time classifier output of the EEG signal, the gaze cross appears in the center of the monitor and moves to the right or left.

In this work, 20 Ag/AgCl electrodes have been used as in previous studies from the area of the motor cortex (FC_5, FC_3, FC_1, FC_2, FC_4, FC_6, C_5, C_3, C_1, C_z, C_2, $C_4$4, C_6, CP_5, CP_3, CP_1, CP_z, CP_2, CP_4, and CP_6). In addition, the signal frequencies were reduced from 1,000 to 100 Hz. Furthermore, the EEG period between 0s and 4s following stimulus onset was chosen as the MI interval.

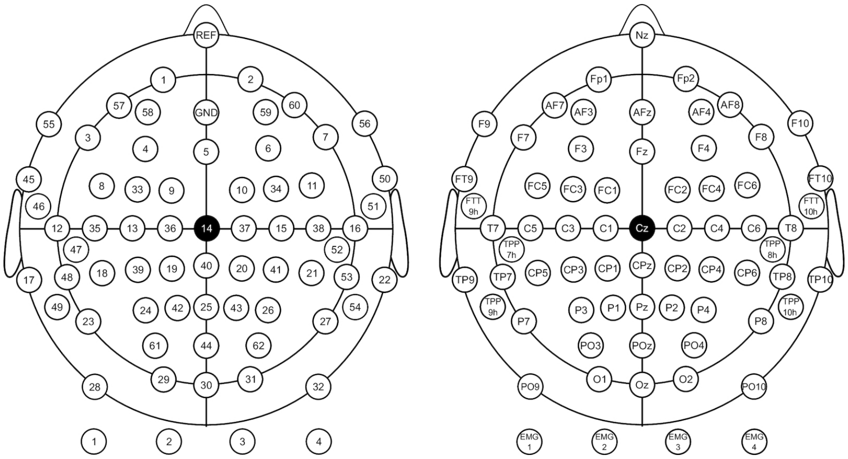


Figure 3.1: The channel configuration of the International 10-20 system (62 EEG and 4 EMG recording electrodes). The left panel indicates the indexing; the right panel corresponds to the location of each electrode.

3.1.2 BCI Competition IV

This data collection consists of EEG data from 9 subjects, offered by Graz University of Technology [14]. First, the subject is seated in front of the screen on the chair. Next, after two seconds (t = 2 s), an arrow on the screen is pointed to the left, right, down, or up and remained on the screen for 1.25 seconds. Therefore, each subject performed 4 different tasks, such as imagining a movement of their right hand, left hand, tongue, and both feet, in this experiment. This dataset contains two session which each of them have been recorded in a separate date with overall 288 trial. Moreover, experiments were obtained with a sampling frequency of 250

Table 3.1: Detailed information about OpenBMI [53] data used in our experiments

Detailed Number of MI data		
Name	Parameters	Extra
Subject	54	None
Classes	2	For Each Subject
Original Data	1000×200×20	Sample Point×Trial×Channel
Riemannian Input Data	200×20×20	Trial×Channel×Channel
Euclidean Input Data	200×20×1000	Trial×Channel×Sample Point

Hz and 22 Ag/AgCl electrodes.

3.1.3 Preprocessing

In this study, before the data gets into MACNet, a preprocessing step is applied. First, since the raw EEG data used in BCI are time-domain signals that vary with time, a third-order Butterworth band-pass filter is utilised between 4 and 38 Hz to create the filtered EEG data within the proper frequency regions. Next, Riemannian prepossessing presented in [106] have applied to the data. Next, for each trial in EEG signal, we calculated the SPD matrix of it. Therefore, EEG signals from both the source and target data were transformed into SPD matrices for a Riemannian view. For instance in BCI Competition IV dataset, the SPD matrix is 22×22 and for OpenBMI dataset, it is 20×20 for each trial. Tables 3.2 and 3.1 contain detailed information about the two datasets.

As indicated, both OpenBMI and BCI Competition IV dataset consists of two sessions. In our studies, we trained our network in one session and tested it in another session. Hence, we have tested the transfer ability of our model in 54 experiments for OpenBMI data and 9 experiments for BCI Competition IV.

Table 3.2: Detailed information about BCI Competition IV [4] data used in our experiments

Detailed Number of MI data

Name	Parameters	Extra
Subject	9	None
Classes	4	For Each Subject
Original Data	1000×288×22	Sample Point×Trial×Channel
Riemannian Input Data	288×22×22	Trial×Channel×Channel
Euclidean Input Data	288×22×1000	Trial×Channel×Sample point

3.2 MACNet Framework

The variability in subjects' conditions across different sessions results in significant differences between the distributions of source and target samples. In this research we analyzed our model in UDA setting. The source domains $(\mathbf{X}^s, \mathbf{Y}^s) = (\mathbf{x}_i^s, \mathbf{y}_i^s)_{i=1}^{N_s}$ consist of N_s labeled samples, while the target domain samples $(\mathbf{X}^t) = (\mathbf{x}_i^t)_{i=1}^{N_t}$ consist of N_t unlabeled samples. In this study, we used the target labels to evaluate the proposed model.

The overall architecture of our proposed MACNet method is illustrated in Figure 3.2. The inputs to the network consist of three different types of data: the source domain, mixup signals, and target domain.

The MACNet framework consists of six components:

(1) The first level mixup, which linearly interpolated source domain EEG signal x_e^s and target domain EEG signal x_e^t to produce mixup signals x_e^m.

(2) The feature extractor network, which comprises a ShallowConvNet and an SPD encoder, maps the Euclidean and Riemannian views of both domains to their corresponding embedding levels, denoted by h_e^s, h_e^t, h_r^s, and h_r^t. The embeddings h_e^s and h_r^s represent the Euclidean and Riemannian embeddings of the source domain, respectively, while h_e^t and h_r^t represent the Euclidean and Riemannian embeddings of the target domain.

(3) The second level mixup, linearly combine feature embeddings of the two domains h_e^s and h_e^t to produce a domain-invariant latent distribution with a more continuous feature distri-

bution, denoted as h_e^{m2}.

(4) The classifier is trained to predict the labels of the source data using h_e^s representation.

(5) The domain classifier is trained to differentiate between the source and target domain embeddings. The input to the domain classifier comprises embeddings from h_e^s, h_e^t, as well as two levels of mixup embeddings from h_e^{m1} and h_e^{m2}.

(6) Contrastive task is applied after applying h_e^s, h_r^s, and h_e^t, h_r^t to the projector network Q.

In the next section, each component described above will be explained in detail.

3.2.1 Feature Extractor

The overall architecture of the feature extractor component can be seen in Figure 3.3. Our feature extractor consists of two encoders for EEG signals with different views. The Euclidean encoder, denoted as E_{sh}, utilizes the ShallowConvNet approach proposed by Schirrmeister et al. [87]. This encoder takes as input the number of trials × number of channels × number of sample points. The encoder has two convolutional layers with 24 kernels per layer, followed by batch normalization, square activation, and a pooling layer with a logarithmic activation function and a dropout rate of 0.6. The last layer of the encoder is a linear layer that matches the dimension of the encoder to the Riemannian feature extractor.

The Riemannian encoder, denoted as E_{sp}, extracts Riemannian and spatial features from the data. As shown in Figure 3.3, the Riemannian encoder consists of seven layers, including BiMap, RBN, ReEig, dropout, and LogEig layers. The input to this network is the number of trials × number of channels × number of channels, as described in Tables 3.1 and 3.2.

Both encoders map the data samples into a representation vector h_e and h_r, respectively, which have the same dimension and are in Euclidean space. Figure 3.3 illustrates the overall architecture of the feature extractor component.

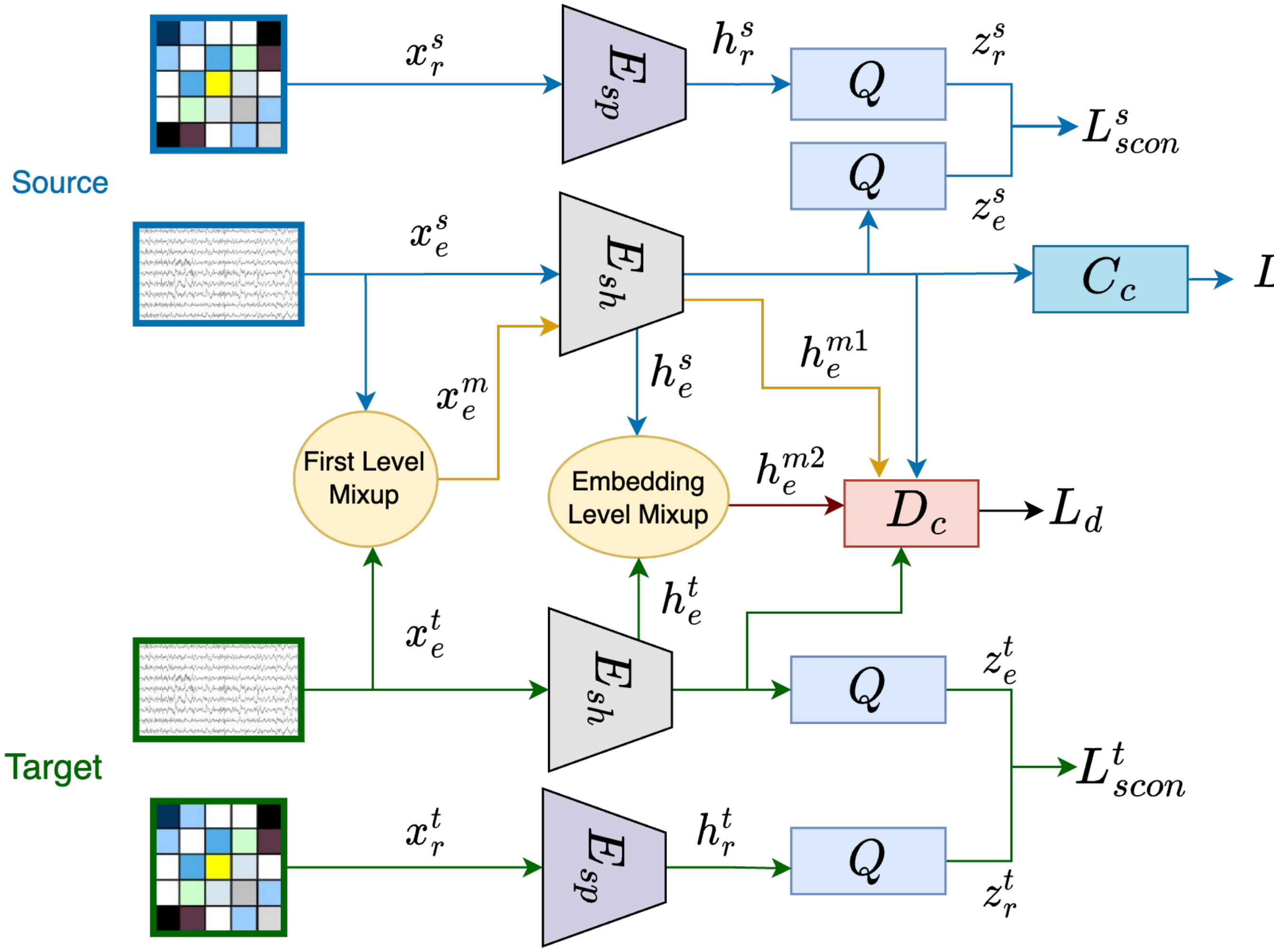


Figure 3.2: The framework of MACNet. The blue lines represent the data from the source domains, while the green lines represent the data from the target domain. The yellow and red lines show the integration of domain mixups in both the signal and embedding levels (first and second level). The feature extractor is comprised of two encoders, E_{sh} and E_{sp}, which represent ShallowConvNet and SPDNet, respectively. Following the encoders, a projection layer Q is an MLP layer that is applied to the embeddings for contrastive loss computation. In the next step, C_c and D_c are classifiers for the class and domain loss.

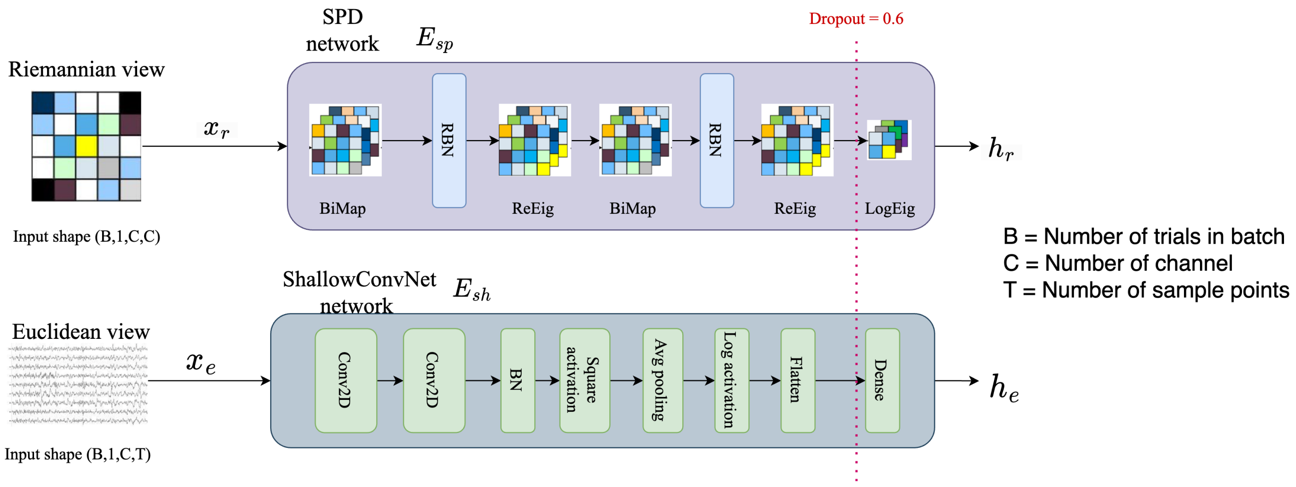


Figure 3.3: The overall architecture of the feature extractor. As it can be seen, feature extractor has two components E_{sp} for SPD matrices data with the shape of (batch, 1, number of channels, number of channels) and E_{sh} for time-domain EEG signals with the shape of (batch, 1, number of channels, number of sample points). The final representation h_r and h_e are both in Euclidean space and have the same dimension.

3.2.2 Domain Mixup On Two Levels

In this study, compared to previous adversarial domain adaptation [75, 93], which used only source and target domains for domain discriminator, in this work, we propose a domain mixup on the signal and embedding levels.

Mixup in the first level

As it can be seen in Figure 3.2 and Figure 3.4, there are three kinds of inputs for MACNet, including source, target, and mixup signals obtained by mixing source and target signals linearly. Therefore, a random pair of source and target sample signals from the batch $(x^s_{e,i}, x^t_{e,i})$ are linearly combined to create mixup signal $x^m_{e,i}$ and a soft domain label l^m_{dom}.

$$x^m_{e,i} = \lambda * x^s_{e,i} + (1 - \lambda) * x^t_{e,i}, \tag{3.1}$$

$$l^m_{dom} = \lambda * l^s_{dom} + (1 - \lambda) * l^t_{dom} \tag{3.2}$$

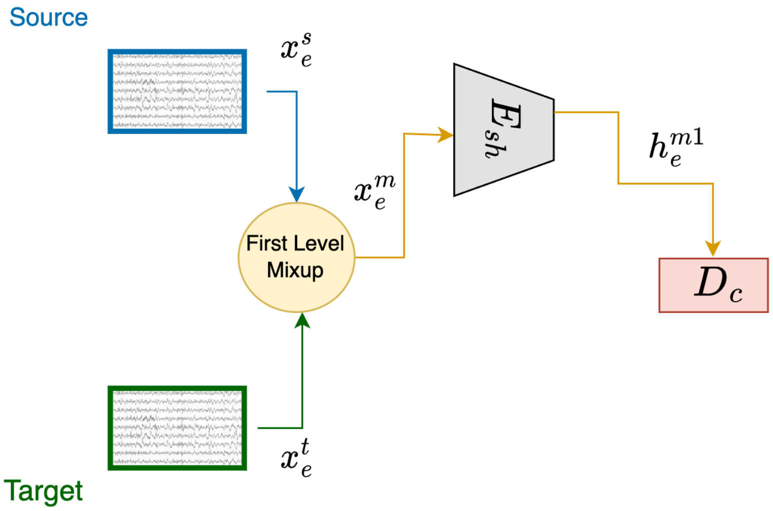


Figure 3.4: Mixup in the first level

where l^s_{dom} and l^t_{dom} represent the label of source and target domains which are manually set to 0 and 1. Moreover, λ is a mixup ratio and randomly sampled from a beta distribution Beta(α, α). In all of our experiments, alpha sets 2.0.

Finally, x_e^m is embedded to h_e^{m1} in the latent space by a encoder E_{sh} (Figure 3.4).

Mixup in the second level

As it can be seen in Figure 3.5 after projecting inputs x_e^s and x_e^t to the encoders, mixup between source and target embedding was employed to produce a more continuous domain-invariant latent distribution, which can be defined as:

$$h_e^{m2} = \lambda * h_e^s + (1 - \lambda) * h_e^t \tag{3.3}$$

where h_e^s is an embedding from a source feature extractor, and h_e^t is an embedding from a target feature extractor. In addition, λ is equal to the one used in the signal level mixup.

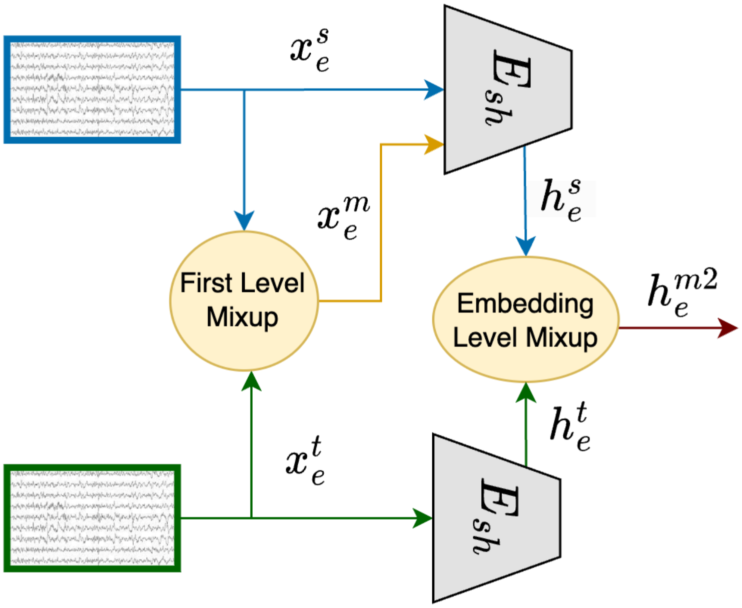


Figure 3.5: Mixup on the embedding level

3.2.3 Supervised Classifier Training

Figure 3.6 indicates the supervised classifier. In order to extract discriminative features from the source domain samples, we utilize a feature extractor denoted as E_{sh}. To enable accurate prediction of the source domain labels, we train a fully connected classifier, denoted as C_c, to predict the corresponding label. As a result, C_c takes the representation h_e^s as input and produces the final prediction. To train the C_c, we use the cross-entropy loss, denoted as L_{ce}^s:

$$L_{ce}^s = \frac{1}{N_s} \sum_{i=1}^{N_s} L(C_c(h_e^s), y_i^s) \tag{3.4}$$

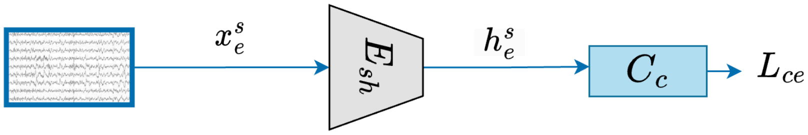


Figure 3.6: Supervised classifier

3.2.4 Domain Classifier Training

The domain classifier model, denoted as D_c, is designed to determine the domain of the input data. As shown in Figure 3.2, the domain classifier model takes four inputs, which are derived from the embeddings of the source and target domains denoted as h_e^s, h_e^t, as well as two levels of mixup embeddings h_e^{m1} and h_e^{m2}.

During training, the domain discriminator model D is optimized to minimize the domain classification loss, while the feature extractor is simultaneously trained to increase this loss.

In our proposed MACNet architecture, we have four domain classification losses corresponding to the source domain, target domain, and two levels of mixup embeddings.

$$L_d^s = \frac{1}{N_s} \sum_{i=1}^{N_s} L(D_c(R(h_{e,i}^s)), d_i^s), \tag{3.5}$$

$$L_d^t = \frac{1}{N_t} \sum_{i=1}^{N_t} L(D_c(R(h_{e,i}^t)), d_i^t), \tag{3.6}$$

$$L_d^{m1} = \frac{1}{N_{m1}} \sum_{i=1}^{N_{m1}} L(D_c(R(h_{e,i}^{m1}))), l_{dom}^m), \tag{3.7}$$

$$L_d^{m2} = \frac{1}{N_{m2}} \sum_{i=1}^{N_{m2}} L(D_c(R(h_{e,i}^{m2})), l_{dom}^m), \tag{3.8}$$

$$L_d = L_d^s + L_d^t + L_d^{m1} + L_d^{m2} \tag{3.9}$$

where $L_d, L_d^s, L_d^t, L_d^{m1}, L_d^{m2}$ represent overall domain loss for source, target, mixup for signal level and embedding level. Moreover, R represents the reversal layer.

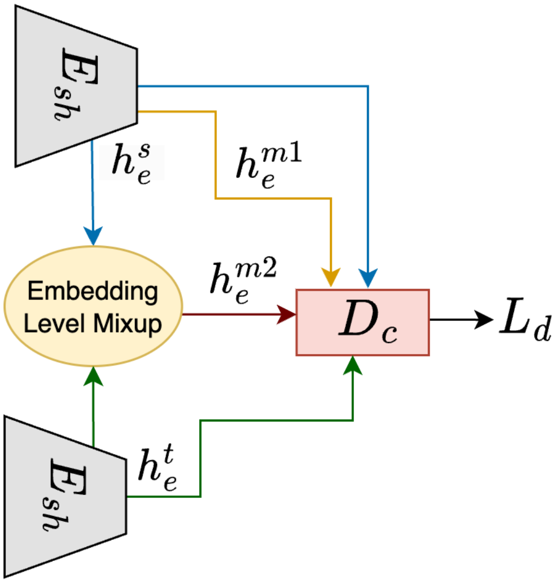


Figure 3.7: Domain classifier

3.2.5 Contrastive Learning With Two Views

In MACNet, both Riemannian and Euclidean representations are captured in the embeddings z_e^s, z_r^s, z_e^t and z_r^t, for both source and target data after projection layer Q (Figure 3.8). As discussed, MACNet uses multi-view feature extractor, which captures valuable representations by extracting spatial and temporal characteristics jointly in a self-supervised way from both the target and source domains. Hence, the performance of inter-subject adaptation can be further improved compared to the last proposed methods. Prior to extracting the high-dimensional latent representations, the Euclidean and Riemannian views are sent to their respective encoders, ShallowconvNet and SPDNet. Both encoders use separate projection heads to transfer the high-dimensional latent representations, where contrastive losses are applied between z_e^s and z_r^s, and between z_e^t and z_r^t. Therefore, MACNet has two contrastive losses, one for the source and one for the target domain defined as L_{scon}^s and L_{scon}^t. The contrastive loss enhances the similarity between two enhanced views of the same EEG signal while decreasing its similarity with other EEG signals. As a result, the classification task for the model become easier.

In overall, MACNet loss is:

$$L_{MACNet} = L_{ce}^s + L_d + L_{scon}^s + L_{scon}^t \tag{3.10}$$

3.3 Training Procedure

In the training process, for training and testing, we used K80 with 12GB GPU memory. Our MACNet framework is developed in PyTorch. In order to minimize the loss, in addition to the SPDNet initial optimizer [12,39], AdamW is used as an optimizer. Moreover, the learning rate sets to 0.01, and for each training process, the iteration is 250. Furthermore, the temperature for contrastive loss is set to 0.07.

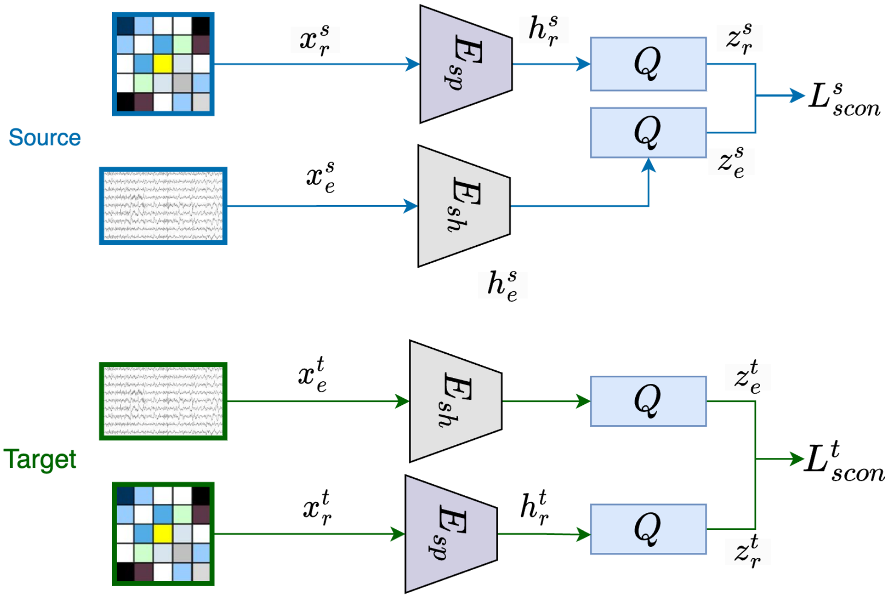


Figure 3.8: Multi-view contrastive

Chapter 4

Results

4.1 Baseline Approaches

In this section, we evaluate the performance of MACNet with other state-of-the-art baseline approaches on two benchmark datasets. Our baseline approaches have been divided into three parts. We have selected two traditional approaches for the first part, namely FBCSP [5] and CCSP [48]. Seven DNN approaches, including SPDNet [39], EEGNet [50], DeepConvNet [87], FBCNet [39], ShallowConvNet [87], Tensor-CSPNet [45], and EEG Riemannian [107], comprise the second part. In the last part, in order to better compare the performance of MACNet, we had selected three other UDA methods, namely ConvNet+DANN, EEGNet+DANN and DJDAN [38]. In this section each baseline method is explained in detail.

4.1.1 Traditional Baseline Methods

FBCSP [39] approach is one of the popular methods for analyzing EEG signals. In this model, CSP is applied to sub-bands of the EEG signal to derive sub-band scores, which are then used as input for a classification algorithm. FBCSP has demonstrated outstanding performance in MI and is regarded as the most representative variant of the CSP method. The FBCSP Python toolbox is available at GitHub [1].

[1] https://fbcsptoolbox.github.io

Another popular traditional method in the BCI field is the CCSP approach [48]. This model uses CSP and principal component analysis to extract several spatial filters that can capture discriminative information across frequency bands. CCSP has shown to improve classification accuracy in various BCI applications, such as MI and emotion recognition, compared to previous feature extraction approaches.

4.1.2 Deep Learning Baseline Methods

In this thesis, we implemented three popular CNN-based models: ShallowConvNet, EEGNet, and ConvNet. In Chapter 2, a detailed explanation of each model is presented. ShallowConvNet and EEGNet models are publicly available for testing on GitHub [2]. Additionally, ConvNet is available on GitHub [3].

FBCNet [39] is a CNN-based method inspired by the success of FBCSP in analyzing EEG signals. FBCNet applies customized CNN layers to individual sub-bands of EEG signals to extract temporal and spatial frequency characteristics. The Python toolbox code for FBCNet is readily available on [4] and has been used to obtain results in our study.

Tensor-CSPNet [45] is a framework for geometric deep learning that capture spatial covariance matrices derived from EEG signals on SPD manifolds. Tensor-CSPNet integrates effective classifiers with deep neural networks on SPD manifolds. The repository of their work is publicly available on GitHub [5].

EEG Riemannian Network [107] is a novel approach for EEG-based BCI that extract spatial and temporal information from Riemannian manifolds and CNNs. In their model, they convert EEG signals to Riemannian covariance matrices, which are then used to extract temporal features by CNNs. Utilizing a fusion technique that learns attention weights for spatial and temporal characteristics, the model can choose the most discriminative features for classi-

[2] https://github.com/Mrswolf/brainda

[3] https://github.com/braindecode/braindecode

[4] https://github.com/ravikiranmane/FBCNet

[5] https://github.com/GeometricBCI/Tensor-CSPNet-and-Graph-CSPNet

fication. The repository of their work is publicly available on GitHub [6].

4.1.3 UDA Baseline Methods

DJDAN [38] is a domain adaptation model based on adversarial learning to improve EEG signal categorization in the target domain by using source domain knowledge. Their method aligns the marginal distribution across domains using a global discriminator and reduces the conditional distribution discrepancy between sub-domains using a local discriminator that conditions deep representations and classifier predictions to learn a domain-invariant feature representation. The authors' proposed setting for the BCI Competition IV aligns with the setting used in our thesis, making it a relevant comparison for our study. However, they have not provided code for their method, preventing us from directly comparing our model with theirs on the OpenBMI dataset.

Many UDA papers have used the Domain Adversarial Neural Network (DANN) proposed in [28] as a baseline to compare their networks. In this thesis, we have also compared our network with this baseline, however, we have customised DANN specifically for our task. Hence, the feature extractors of the two networks presented for DANN are EEGNet [50] and ConvNet [87].

4.2 Evaluation

As described previously, different sessions for each subject have considerable variances since subjects' conditions under each session are various. The average prediction accuracy of all subjects for both datasets is presented in Table 4.1 to demonstrate the transferability performance of MACNet. Subsequently, we measured the quality of predictions using classification accuracy presented as follows:

$$Accuracy = \frac{T_p + T_n}{(T_p + T_n + F_p + F_n)} \tag{4.1}$$

[6] https://github.com/guangyizhangbci/EEGRiemannian

where T_p is true positive, T_n true negative, F_p false positive, and F_n false negative. The objective is to identify a model that provides the best fit and accuracy.

4.3 Comparison with Baseline Approaches

Our proposed method, MACNet, was compared against several baseline methods on both datasets, and the results are presented in Table 4.1. One of the recent UDA methods, DJDAN [38], achieved a state-of-the-art performance of 81.52% on the BCI Competition IV dataset. However, MACNet outperformed DJDAN [65] with a high accuracy of 83.79%, representing a 2.27% improvement. Moreover, MACNet achieved state-of-the-art performance on the OpenBMI dataset, achieving an accuracy of 80.00% across all 54 subjects, with an average increase of 3.15% compared to the best baseline method.

In summary, MACNet outperformed all traditional methods (CSP and its variants) and other DNN models (EEGNet, ShallowConvNet, ConvNet) and domain adaptation methods (DJDAN, ConvNet+DANN, EEGNet+DANN) on both datasets. Figure 4.1 shows a comparison of MACNet's performance against different baselines. As can be seen, for all subjects in both datasets, the points are above the green line, demonstrating the high classification ability of MACNet.

4.4 Comparison with Different Transfer Setup

Initially, in the last section, all the baseline methods used all of source data and all data of target domain without its labels for DA. In this part, we decreased the amount of the source and target data to evaluate our method in more extreme setups and compare it with other baseline approaches. The results are shown in Figures 4.2, 4.4, 4.3 and 4.5, for both dataset, where the scatter points represent the accuracy for the test data.

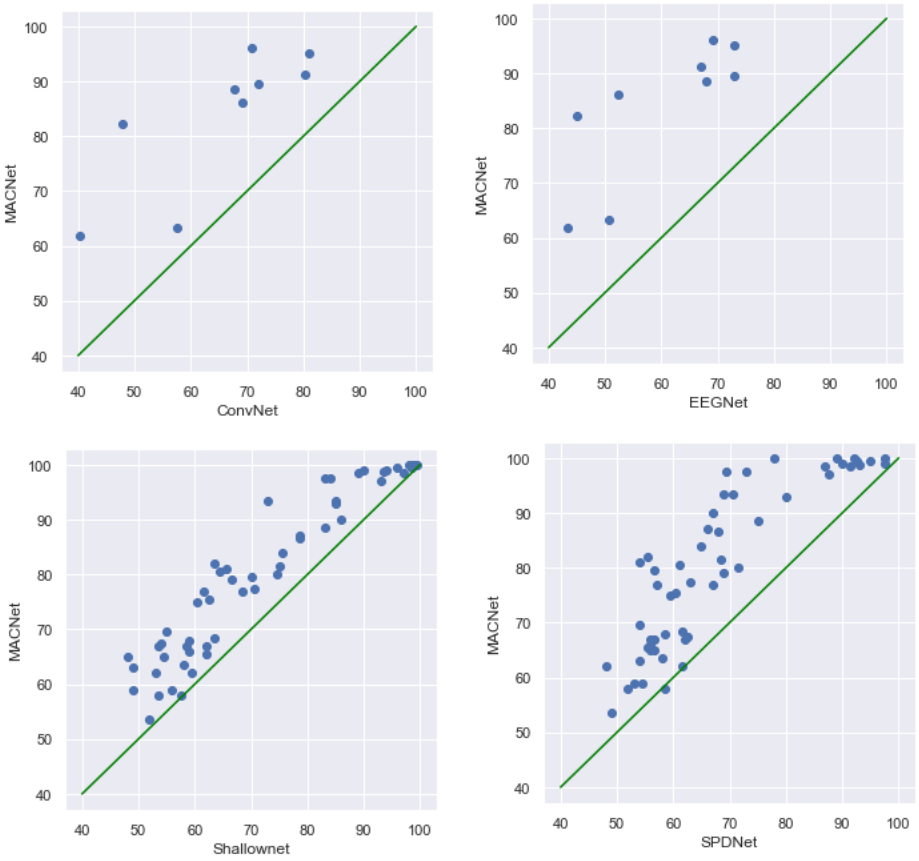


Figure 4.1: The comparison of mean classification accuracy in a session-to-session transfer experiment between various baseline approaches and MACNet. The green diagonal line indicates the boundary line where the comparable method performs equally as the MACNet. The first row is on BCI Competition IV, and the second is OpenBMI data.

Table 4.1: Classification results experimented with different methods in all subjects. Moreover, for the rows that are shown ”-,” the method is not implemented for the second dataset, and their code was not also available.

Accuracy in % for different methods		
Method	Datasets	
	BCI Competition IV [14]	OpenBMI [53]
FBCSP [5]	67.75	60.36
CCSP [48]	66.51	62.21
SPDNet [39]	55.67	67.77
EEGNet [50]	60.31	63.63
DeepConvNet [87]	65.17	60.77
FBCNet [39]	76.2	67.19
ShallowConvNet [87]	68.2	71.79
Tensor-CSPNet [45]	72.96	69.65
EEG Riemannian [107]	75.51	72.21
DJDAN [38]	81.52	-
ConvNet+DANN	69.09	73.24
EEGNet+DANN	77.81	76.85
MACNet	**83.79**	**80.00**

4.4.1 Comparison of Different Amount of Source Data

To evaluate the impact of the number of source data on the performance of our proposed method, we gradually increased the amount of source data from 10% to 100% of the samples, in increments of 20%. When the number of source data is limited, the domain classifier may struggle to learn a reliable decision boundary to distinguish between the two domains, leading to poor feature extraction and limited generalization to the target domain. Figures 4.2 and 4.3 demonstrate the ability of MACNet and other baseline UDA methods for both datasets.

Our experimental results demonstrate that the accuracy of the two benchmark approaches decreases more significantly when the number of source data is low. Specifically, when only 10% of labeled data is given, our method outperforms the other two UDA methods by a significant margin. MACNet achieves 54.68% on BCI Competition IV while the other two baseline are less than 40%. Moreover, MACNet outperformed the other two baseline models with 67.23% compare to 66% and 64% in the OpenBMI dataset. As the amount of labeled data increases to 30%, our method achieves an accuracy of 68% for the BCI Competition IV dataset, while the other two UDA models remain below 60%. Additionally, for the OpenBMI dataset with 30% of source data, MACNet improves by more than 2%.

When 50% of the source data is used for both datasets, the accuracy of both baseline approaches is approximately 62.57% EEGNet+DANN and 58.14% ConvNet+DANN for BCI Competition IV dataset, and 71.8% EEGNet+DANN and 70.02% ConvNet+DANN for OpenBMI dataset. However, our proposed model achieves 74.53% and 77.32% for the two datasets, respectively, outperforming the other baselines by a significant margin. These results demonstrate the efficiency of our technique, particularly in situations where there is a limited amount of labeled data available.

4.4.2 Comparison of Different Amount of Target Data

In the second step of our experimental evaluation, we increased the amount of target data from 10% to 100% in increments of 20%, and the results are presented in Figures 4.4 and 4.5. Our analysis shows that the performance of the other two UDA approaches decreases significantly when only 10% of the target data is used for alignment in both datasets. Specifically, in the BCI Competition IV dataset, both baseline approaches achieve an accuracy below 68%, while our proposed method MACNet achieves 76%. Similarly, in the OpenBMI dataset, both baseline approaches perform below 67%, while MACNet achieves an accuracy of more than 72%. Furthermore, our approach consistently outperforms the other two UDA methods for different percentages of target data used in the model, demonstrating a significant improvement in performance.

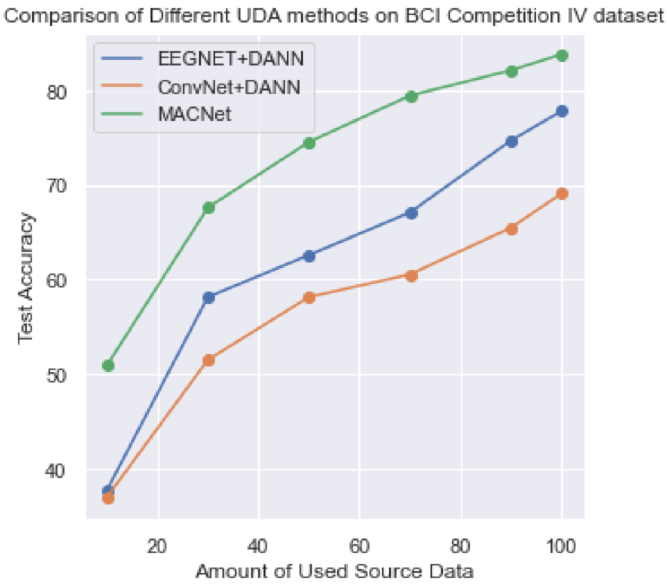


Figure 4.2: Performance comparison between our method and two other benchmark UDA models for different amount of source data (%) used during training BCI Competition IV dataset.

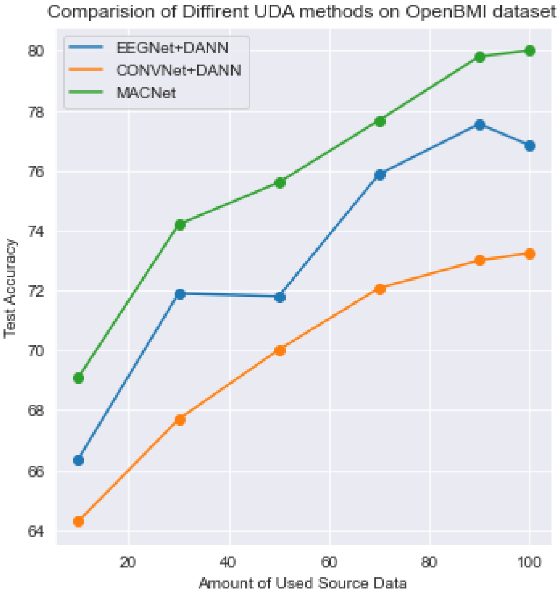


Figure 4.3: Performance comparison between our method and two other benchmark UDA models for different amount of source data (%) used during training Open BMI dataset.

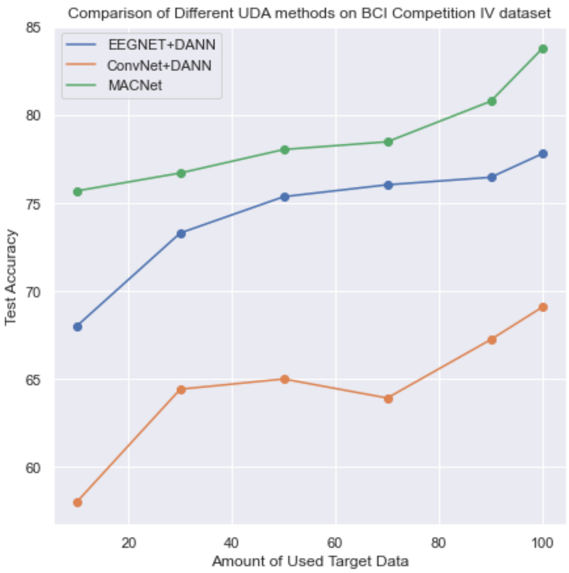


Figure 4.4: Performance comparison between our method and two other benchmark UDA models for different amount of target data (%) used during training for BCI Competition IV dataset.

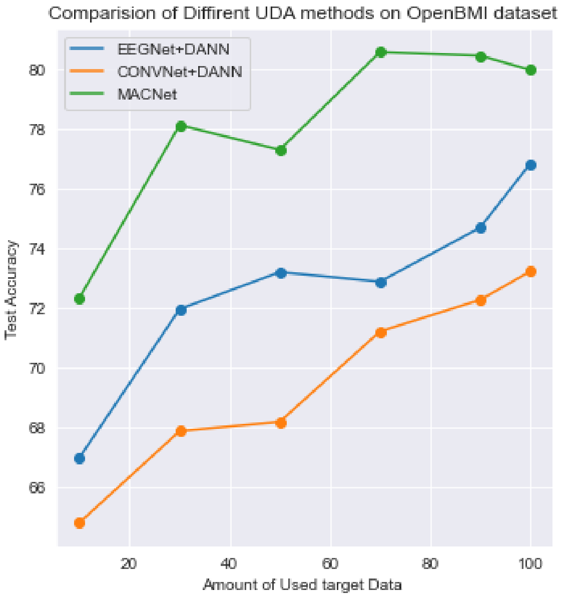


Figure 4.5: Performance comparison between our method and two other benchmark UDA models for different amount of target data (%) used during training for Open BMI dataset.

4.5 Ablation Study

To evaluate the significance of the components of MACNet on its performance, this paper presents an ablation study. The experiment is conducted with constant hyperparameters, and two components are investigated. The first component is L_d^{m2}, which represents the domain classifier loss for mixup in the embedding level. The second component is the multi-view contrastive loss, which is performed on source and target data L_{scon}^s and L_{scon}^t.

Table 4.2 shows the average accuracy of all presented methods. The first set of experiments (first raw) is the scenario where the model is trained without contrastive loss and mixup in the embedding level. In the second set of experiments (second raw), the model is trained without contrastive loss, and in the last experiment, the model is trained without mixup in the embedding level. Furthermore, Figures 4.7 and 4.6 illustrate all of the proposed scenarios for both datasets for each specific subject.

The results show that adding mixup in the embedding level to the baseline model significantly improves the model's performance, as seen in the second and fourth rows of Table 4.2 for both datasets. This is because mixup at the embedding level enables the model to be more continuous in a domain-invariant embedding space.

Additionally, the contrastive component is also analyzed in the study. The third and fourth rows of Table 4.2 indicate that adding contrastive loss to the model enhances its performance for both datasets. This finding is consistent with the results shown in Figures 4.7 and 4.6. The contrastive task extracts valuable features on both views, which are pulled together to create a more semantic representation. This process helps the model to have a better representation of domain-invariant features.

4.6 Visualization

T-distributed stochastic neighbor embedding (t-SNE) [94] is a powerful technique for visualizing high-dimensional data by reducing the dimensions to 2 or 3-dimensional space. In this section, we utilize t-SNE to visualize the embeddings of MACNet to investigate its classifica-

Table 4.2: Classification results experimented in all of the subjects with different methods

Mixup embedding	Contrastive	BCI Competition IV	OpenBMI
×	×	81.48	76.99
✓	×	82.94	77.12
×	✓	82.98	79.18
✓	✓	83.79	80.00

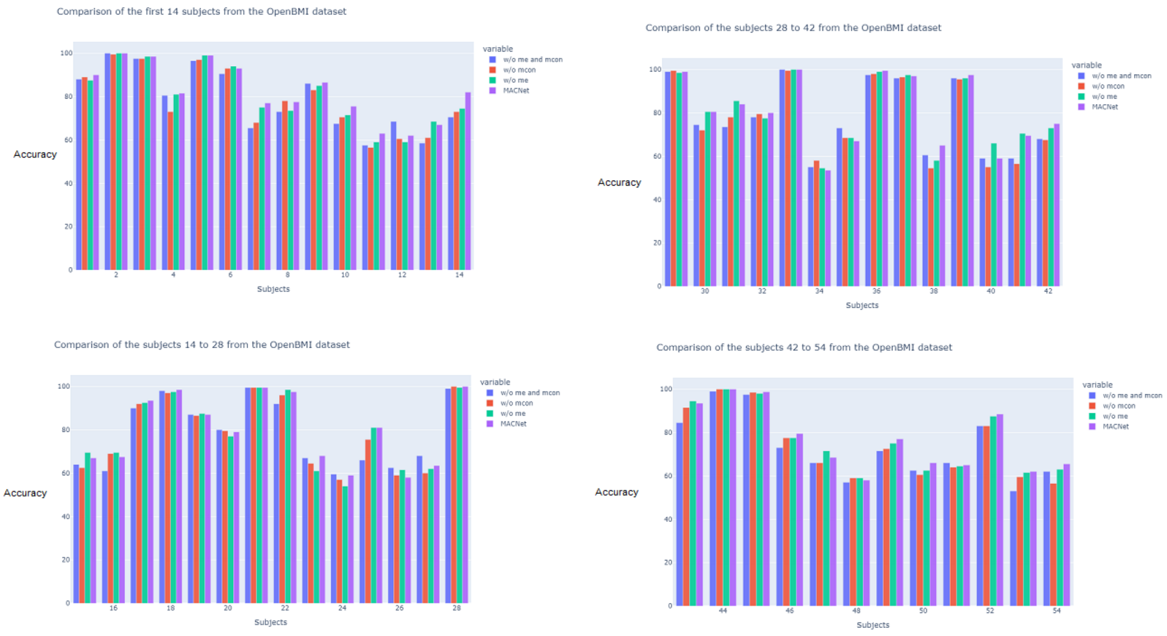


Figure 4.6: Comparison of the different proposed methods on the OpenBMI dataset

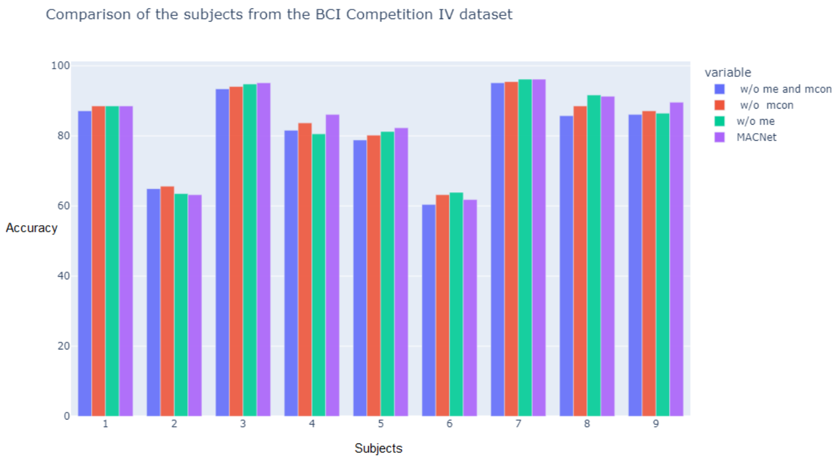


Figure 4.7: Comparison of the different proposed methods on BCI Competition IV dataset

tion performance and domain alignment ability. Specifically, we demonstrate the effectiveness of MACNet by embedding it in a two-dimensional space through: (1) pulling together the source and target classes of both domains based on contrastive loss, and (2) aligning both the source and target domains based on the multi-level mixup and adversarial loss. Figures 4.8 and 4.9 show the test data for subjects (A03, A04, A08, A09) of the BCI Competition IV dataset and subjects (O09, O28) of the OpenBMI dataset, respectively.

As shown in both figures, the left side represents the test embedding before applying MACNet, where the labels are mixed and cannot be clustered. However, the right side shows the final epoch of MACNet on the test data, where the feature distribution is discriminative and can effectively classify the target data.

Furthermore, Figure 4.10 demonstrates the domain alignment ability of MACNet. Before applying MACNet, the original features in the source and target domains are very sparse and distant from each other, making it challenging for the model to correctly classify the labels. However, after MACNet, the feature embeddings are completely overlapped and mixed, indicating that MACNet successfully aligns both the source and target domain distributions and creates domain-invariant features in the latent space.

Overall, t-SNE visualization results show that MACNet can effectively extract domain-invariant features and improve the model's classification performance by aligning the source and target domains.

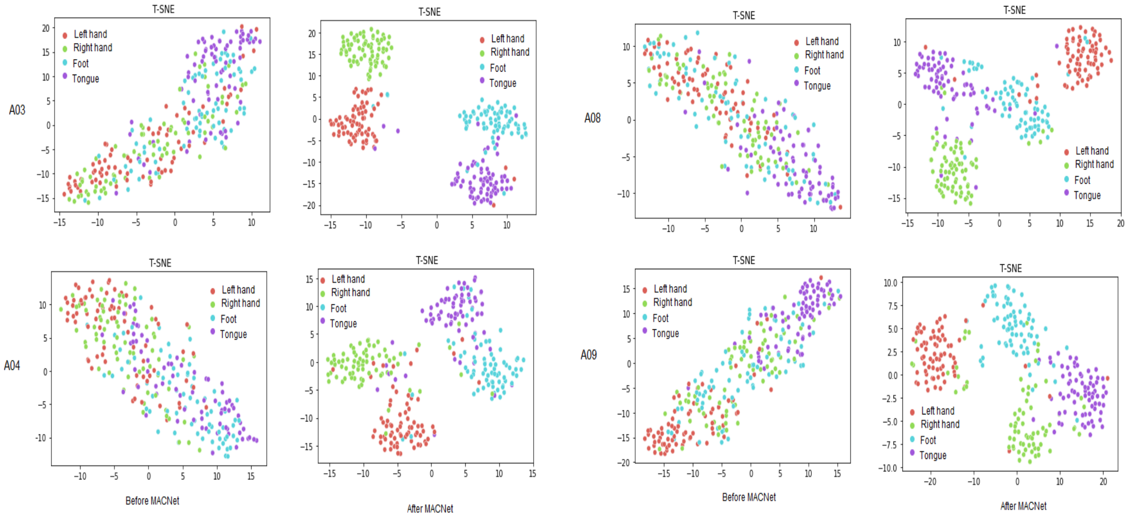


Figure 4.8: Feature embedding of the test data before MACNet and after MACNet for BCI Competition IV on subjects 3, 4, 8 and 9

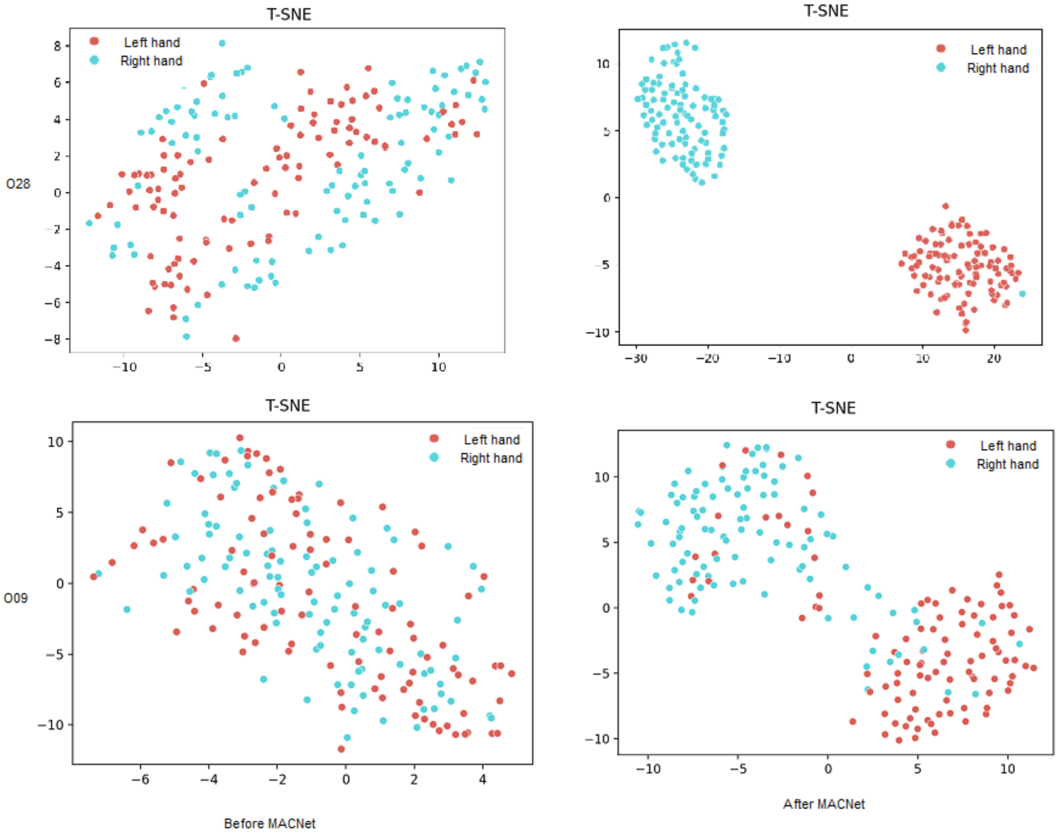


Figure 4.9: Feature embedding of the test data before MACNet and after MACNet for OpenBMI on subjects 9 and 28

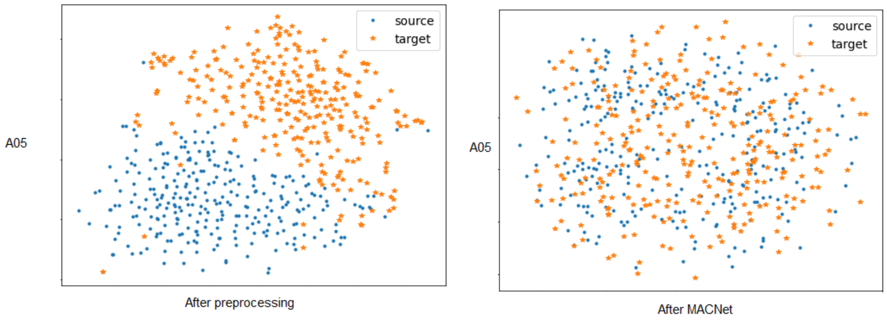


Figure 4.10: Embedding of source and target domain of subject 5 of the BCI Competition IV dataset before and after MACNet

Chapter 5

Conclusion and Future works

5.1 Conclusion

We investigate the problem of UDA in MI data in this thesis. In a BCI context, due to the challenges, it is difficult and expensive to obtain more data. Moreover, due to the domain shift problem, the disturbance of training and test data is different. Hence, the model will perform poorly when it is evaluated on the test data. To address this issue, one of the most effective solutions is UDA, which uses the data from the source domain with its label and only requires the collection of unlabeled data for the target domain.

Even though there have been some studies in this field for an inter-subject task in MI, such as traditional learning, DNN, and DA, there are still many ways to improve performance in the context. This thesis proposes a novel multi-view UDA method to handle the EEG-based MI classification task called MACNet. MACNet extracts spatio-temporal information from multiple views of both the Riemannian manifold and Euclidean space to learn better representations. We proposed a multi-level mixup that generates a more continuous domain-invariant latent space to enhance a domain-invariant representation among both source and target domains. In addition, we test our framework with two public datasets for MI classification. Our results illustrate the robustness of our model in both binary classification and multi-class classification in both datasets. Furthermore, extensive experiments on all methods presented in this thesis were conducted, and overall, MACNet outperformed all state-of-the-art models.

5.2 Future Works

In this thesis we have have focused on the domain adaptation problem in inter-subject of MI data. However, DA problem in these type of data is not yet fully solved and need additional research.

Multi-Source DA

One exciting research approach in this field is implementing multi-source DA for BCIs. In this setup, the training set includes multiple domains, each of which has various distributions. For instance, the training data can be 8 out of 9 subjects, and we will try to DA with another remaining subject. In addition to this method, another exciting approach can be multi-source to multi-target UDA. For instance, a UDA model learns on the BCI Competition IV dataset and aligns with OpenBMI.

Domain Generalization

In this thesis, the data were nonstationary and needed to be aligned. As a result, the source and target data were aligned, using both source and unlabeled target data. However, in many tasks, there are no target data available; hence in these types of problems, the objective is that the model learns from several domains and can generalize well for the unseen data.

As noted in the previous sections, numerous DA works have been presented in computer vision and natural language processing; however, there have not been many studies on DA difficulties in BCIs, and neither are there many codes accessible for these types of challenges. However, as indicated, this field has a lot of potentials to grow. Thus, it is hoped that the outcomes of this thesis will inspire other researchers to make domain adaptations and continue this field for different types of BCI data.

Bibliography

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] Haider Al-Tahan and Yalda Mohsenzadeh. Clar: Contrastive learning of auditory representations. In *International Conference on Artificial Intelligence and Statistics*, pages 2530–2538. PMLR, 2021.

[3] Hamdi Altaheri, Ghulam Muhammad, Mansour Alsulaiman, Syed Umar Amin, Ghadir Ali Altuwaijri, Wadood Abdul, Mohamed A Bencherif, and Mohammed Faisal. Deep learning techniques for classification of electroencephalogram (eeg) motor imagery (mi) signals: a review. *Neural Computing and Applications*, pages 1–42, 2021.

[4] Kai Keng Ang, Zheng Yang Chin, Chuanchu Wang, Cuntai Guan, and Haihong Zhang. Filter bank common spatial pattern algorithm on bci competition iv datasets 2a and 2b. *Frontiers in neuroscience*, 6:39, 2012.

[5] Kai Keng Ang, Zheng Yang Chin, Haihong Zhang, and Cuntai Guan. Filter bank common spatial pattern (fbcsp) in brain-computer interface. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 2390–2397. IEEE, 2008.

[6] KK Ang, ZY Chin, H Zhang, and C Guan. Ieee international joint conference on neural networks, 2008. ijcnn 2008.(ieee world congress on computational intelligence). 2008.

[7] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM journal on matrix analysis and applications*, 29(1):328–347, 2007.

[8] Ruslan Aydarkhanov, Marija Uščumlić, Ricardo Chavarriaga, Lucian Gheorghe, and Jose del R Millan. Spatial covariance improves bci performance for late erps components with high temporal variability. *Journal of Neural Engineering*, 17(3):036030, 2020.

[9] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Multiclass brain–computer interface classification by riemannian geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928, 2011.

[10] Benjamin Blankertz, Guido Dornhege, Matthias Krauledat, Klaus-Robert Müller, and Gabriel Curio. The non-invasive berlin brain–computer interface: fast acquisition of effective performance in untrained subjects. *NeuroImage*, 37(2):539–550, 2007.

[11] 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, 2007.

[12] Daniel Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, and Matthieu Cord. Riemannian batch normalization for spd neural networks. In *Advances in Neural Information Processing Systems*, pages 15489–15500, 2019.

[13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[14] Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, and Gert Pfurtscheller. Bci competition 2008–graz data set a. *Institute for Knowledge Discov-*

ery (Laboratory of Brain-Computer Interfaces), Graz University of Technology, 16:1–6, 2008.

[15] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[16] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[17] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[18] H Cho, M Ahn, S Ahn, et al. Supporting data for "eeg datasets for motor imagery brain computer interface.". *GigaScience Database*, 2017.

[19] Marco Congedo, Alexandre Barachant, and Rajendra Bhatia. Riemannian geometry for eeg-based brain-computer interfaces; a primer and a review. *Brain-Computer Interfaces*, 4(3):155–174, 2017.

[20] Alexander Craik, Yongtian He, and Jose L Contreras-Vidal. Deep learning for electroencephalogram (eeg) classification tasks: a review. *Journal of neural engineering*, 16(3):031001, 2019.

[21] Mengxi Dai, Dezhi Zheng, Shucong Liu, and Pengju Zhang. Transfer kernel common spatial patterns for motor imagery brain-computer interface classification. *Computational and mathematical methods in medicine*, 2018, 2018.

[22] Andre Esteva, Katherine Chou, Serena Yeung, Nikhil Naik, Ali Madani, Ali Mottaghi, Yun Liu, Eric Topol, Jeff Dean, and Richard Socher. Deep learning-enabled medical computer vision. *NPJ digital medicine*, 4(1):1–9, 2021.

[23] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Transfer learning for time series classification. In *2018 IEEE international conference on big data (Big Data)*, pages 1367–1376. IEEE, 2018.

[24] Ruicheng Feng, Chongyi Li, Huaijin Chen, Shuai Li, Chen Change Loy, and Jinwei Gu. Removing diffraction image artifacts in under-display camera via dynamic skip connection network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 662–671, June 2021.

[25] Wolfgang Förstner and Boudewijn Moonen. A metric for covariance matrices. In *Geodesy-the Challenge of the 3rd Millennium*, pages 299–309. Springer, 2003.

[26] Rongrong Fu, Yongsheng Tian, Tiantian Bao, Zong Meng, and Peiming Shi. Improvement motor imagery eeg classification based on regularized linear discriminant analysis. *Journal of medical systems*, 43(6):1–13, 2019.

[27] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[28] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[29] Habibeh Ghaheri and Alireza Ahmadyfard. Extracting common spatial patterns from eeg time segments for classifying motor imagery classes in a brain computer interface (bci). *Scientia Iranica*, 20(6):2061–2072, 2013.

[30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[31] Yanjie Gou, Yinjie Lei, Lingqiao Liu, Yong Dai, and Chunxu Shen. Contextualize knowledge bases with transformer for end-to-end task-oriented dialogue systems. *arXiv preprint arXiv:2010.05740*, 2020.

[32] L John Greenfield, James D Geyer, and Paul R Carney. *Reading EEGs: A practical approach*. Lippincott Williams & Wilkins, 2012.

[33] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[34] Lindsay F Haas. Hans berger (1873–1941), richard caton (1842–1926), and electroencephalography. *Journal of Neurology, Neurosurgery & Psychiatry*, 74(1):9–9, 2003.

[35] Mehrtash Harandi and Mathieu Salzmann. Riemannian coding and dictionary learning: Kernels to the rescue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3926–3935, 2015.

[36] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[37] Michael Hersche, Tino Rellstab, Pasquale Davide Schiavone, Lukas Cavigelli, Luca Benini, and Abbas Rahimi. Fast and accurate multiclass inference for mi-bcis using large multiscale temporal and spectral features. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1690–1694. IEEE, 2018.

[38] Xiaolin Hong, Qingqing Zheng, Luyan Liu, Peiyin Chen, Kai Ma, Zhongke Gao, and Yefeng Zheng. Dynamic joint domain adaptation network for motor imagery classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29:556–565, 2021.

[39] Zhiwu Huang and Luc Van Gool. A riemannian network for spd matrix learning. *arXiv preprint arXiv:1608.04233*, 2016.

[40] Zhiwu Huang and Luc Van Gool. A riemannian network for spd matrix learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[41] Anna L Hudson, Xavier Navarro-Sune, Jacques Martinerie, Pierre Pouget, Mathieu Raux, Mario Chavez, and Thomas Similowski. Electroencephalographic detection of respiratory-related cortical activity in humans: from event-related approaches to continuous connectivity evaluation. *Journal of Neurophysiology*, 115(4):2214–2223, 2016.

[42] Eugene M Izhikevich. *Dynamical systems in neuroscience*. MIT press, 2007.

[43] Vinay Jayaram, Morteza Alamgir, Yasemin Altun, Bernhard Scholkopf, and Moritz Grosse-Wentrup. Transfer learning in brain-computer interfaces. *IEEE Computational Intelligence Magazine*, 11(1):20–31, 2016.

[44] Eunjin Jeon, Wonjun Ko, and Heung-Il Suk. Domain adaptation with source selection for motor-imagery based bci. In *2019 7th International Winter Conference on Brain-Computer Interface (BCI)*, pages 1–4. IEEE, 2019.

[45] Ce Ju and Cuntai Guan. Tensor-cspnet: A novel geometric deep learning framework for motor imagery classification. *arXiv preprint arXiv:2202.02472*, 2022.

[46] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[47] Hyohyeong Kang and Seungjin Choi. Bayesian common spatial patterns for multi-subject eeg classification. *Neural Networks*, 57:39–50, 2014.

[48] Hyohyeong Kang, Yunjun Nam, and Seungjin Choi. Composite common spatial pattern for subject-to-subject transfer. *IEEE Signal Processing Letters*, 16(8):683–686, 2009.

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

[50] 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, 2018.

[51] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

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

[53] Min-Ho Lee, O-Yeon Kwon, Yong-Jeong Kim, Hong-Kyung Kim, Young-Eun Lee, John Williamson, Siamac Fazli, and Seong-Whan Lee. Eeg dataset and openbmi toolbox for three bci paradigms: an investigation into bci illiteracy. *GigaScience*, 8(5):giz002, 2019.

[54] Steven Lemm, Benjamin Blankertz, Gabriel Curio, and K-R Muller. Spatio-spectral filters for improving the classification of single trial eeg. *IEEE transactions on biomedical engineering*, 52(9):1541–1548, 2005.

[55] Feng Li, Yi Xia, Fei Wang, Dengyong Zhang, Xiaoyu Li, and Fan He. Transfer learning algorithm of p300-eeg signal based on xdawn spatial filter and riemannian geometry classifier. *Applied Sciences*, 10(5):1804, 2020.

[56] Jinpeng Li, Shuang Qiu, Changde Du, Yixin Wang, and Huiguang He. Domain adaptation for eeg emotion recognition based on latent representation similarity. *IEEE Transactions on Cognitive and Developmental Systems*, 12(2):344–353, 2019.

[57] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.

[58] Bin Liang, Hang Su, Rongdi Yin, Lin Gui, Min Yang, Qin Zhao, Xiaoqi Yu, and Ruifeng Xu. Beta distribution guided aspect-aware graph for aspect category sentiment analysis with affective knowledge. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 208–218, 2021.

[59] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

[60] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. Completer: Incomplete multi-view clustering via contrastive prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11174–11183, 2021.

[61] Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. A review of classification algorithms for eeg-based brain–computer interfaces: a 10 year update. *Journal of neural engineering*, 15(3):031005, 2018.

[62] Fabien Lotte, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche, and Bruno Arnaldi. A review of classification algorithms for eeg-based brain–computer interfaces. *Journal of neural engineering*, 4(2):R1, 2007.

[63] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2019.

[64] Abu Saleh Musa Miah, Md Abdur Rahim, and Jungpil Shin. Motor-imagery classification using riemannian geometry with median absolute deviation. *Electronics*, 9(10):1584, 2020.

[65] Zhengqing Miao, Xin Zhang, Carlo Menon, Yelong Zheng, Meirong Zhao, and Dong Ming. Priming cross-session motor imagery classification with a universal deep domain adaptation framework. *arXiv preprint arXiv:2202.09559*, 2022.

[66] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[67] Mostafa Neo Mohsenvand, Mohammad Rasool Izadi, and Pattie Maes. Contrastive representation learning for electroencephalogram classification. In *Machine Learning for Health*, pages 238–253. PMLR, 2020.

[68] Masaki Nakanishi, Yu-Te Wang, Tzyy-Ping Jung, John K Zao, Yu-Yi Chien, Alberto Diniz-Filho, Fabio B Daga, Yuan-Pin Lin, Yijun Wang, and Felipe A Medeiros. Detecting glaucoma with a portable brain-computer interface for objective assessment of visual function loss. *JAMA ophthalmology*, 135(6):550–557, 2017.

[69] Chuong H Nguyen and Panagiotis Artemiadis. Eeg feature descriptors and discriminant analysis under riemannian manifold perspective. *Neurocomputing*, 275:1871–1883, 2018.

[70] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil. Brain computer interfaces, a review. *sensors*, 12(2):1211–1279, 2012.

[71] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017.

[72] Erlin Pan and Zhao Kang. Multi-view contrastive graph clustering. *Advances in neural information processing systems*, 34:2148–2159, 2021.

[73] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *Proceed-*

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11205–11214, 2021.

[74] Sang-Hoon Park and Sang-Goog Lee. Small sample setting and frequency band selection problem solving using subband regularized common spatial pattern. *IEEE Sensors Journal*, 17(10):2977–2983, 2017.

[75] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[76] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A riemannian framework for tensor computing. *International Journal of computer vision*, 66(1):41–66, 2006.

[77] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A riemannian framework for tensor computing. *International Journal of computer vision*, 66(1):41–66, 2006.

[78] Gert Pfurtscheller, Ch Neuper, Doris Flotzinger, and Martin Pregenzer. Eeg-based discrimination between imagination of right and left hand movement. *Electroencephalography and clinical Neurophysiology*, 103(6):642–651, 1997.

[79] Roi Pony, Itay Naeh, and Shie Mannor. Over-the-air adversarial flickering attacks against video recognition networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 515–524, June 2021.

[80] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[81] Herbert Ramoser, Johannes Muller-Gerking, and Gert Pfurtscheller. Optimal spatial filtering of single trial eeg during imagined hand movement. *IEEE transactions on rehabilitation engineering*, 8(4):441–446, 2000.

[82] Herbert Ramoser, Johannes Muller-Gerking, and Gert Pfurtscheller. Optimal spatial filtering of single trial eeg during imagined hand movement. *IEEE transactions on rehabilitation engineering*, 8(4):441–446, 2000.

[83] Pedro Luiz Coelho Rodrigues, Christian Jutten, and Marco Congedo. Riemannian procrustes analysis: transfer learning for brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, 66(8):2390–2401, 2018.

[84] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):801–814, 2018.

[85] Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive learning of general-purpose audio representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3875–3879. IEEE, 2021.

[86] Wojciech Samek, Carmen Vidaurre, Klaus-Robert Müller, and Motoaki Kawanabe. Stationary common spatial patterns for brain–computer interfacing. *Journal of neural engineering*, 9(2):026013, 2012.

[87] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human brain mapping*, 38(11):5391–5420, 2017.

[88] Bin Shan, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-vil 2.0: Multi-view contrastive learning for image-text pre-training. *arXiv preprint arXiv:2209.15270*, 2022.

[89] Heung-Il Suk and Seong-Whan Lee. A novel bayesian framework for discriminative feature extraction in brain-computer interfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):286–299, 2012.

[90] Yousef Rezaei Tabar and Ugur Halici. A novel deep learning approach for classification of eeg motor imagery signals. *Journal of neural engineering*, 14(1):016003, 2016.

[91] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020.

[92] Marco Toldo, Andrea Maracani, Umberto Michieli, and Pietro Zanuttigh. Unsupervised domain adaptation in semantic segmentation: a review. *Technologies*, 8(2):35, 2020.

[93] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[94] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[95] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[96] Hua Wang, Yanchun Zhang, et al. Detection of motor imagery eeg signals employing naïve bayes based learning process. *Measurement*, 86:148–158, 2016.

[97] Yixin Wang, Shuang Qiu, Xuelin Ma, and Huiguang He. A prototype-based spd matrix network for domain adaptation eeg emotion recognition. *Pattern Recognition*, 110:107626, 2021.

[98] Bohan Wu, Suraj Nair, Roberto Martin-Martin, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2318–2328, June 2021.

[99] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Xiang Ao, Xin Chen, Xu Zhang, Fuzhen Zhuang, Leyu Lin, and Qing He. Multi-view multi-behavior contrastive learning in recommendation. In *Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, April 11–14, 2022, Proceedings, Part II*, pages 166–182. Springer, 2022.

[100] Jiachen Xu, Vinay Jayaram, Bernhard Schölkopf, and Moritz Grosse-Wentrup. Feature extraction from the hermitian manifold for brain-computer interfaces. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 965–968. IEEE, 2019.

[101] Jie Xu, Huayi Tang, Yazhou Ren, Liang Peng, Xiaofeng Zhu, and Lifang He. Multi-level feature learning for contrastive multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16051–16060, 2022.

[102] Huijuan Yang, Siavash Sakhavi, Kai Keng Ang, and Cuntai Guan. On the use of convolutional neural networks and augmented csp features for multi-class motor imagery of eeg signals classification. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2620–2623. IEEE, 2015.

[103] Le Yang, Marc Arnaudon, and Frédéric Barbaresco. Riemannian median, geometry of covariance matrices and radar target detection. In *The 7th European Radar Conference*, pages 415–418. IEEE, 2010.

[104] Rui Ye and Qun Dai. Implementing transfer learning across different datasets for time series forecasting. *Pattern Recognition*, 109:107617, 2021.

[105] Florian Yger, Maxime Berar, and Fabien Lotte. Riemannian approaches in brain-computer interfaces: a review. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(10):1753–1762, 2016.

[106] Paolo Zanini, Marco Congedo, Christian Jutten, Salem Said, and Yannick Berthoumieu. Transfer learning: A riemannian geometry framework with applications to brain–

computer interfaces. *IEEE Transactions on Biomedical Engineering*, 65(5):1107–1116, 2017.

[107] Ali Zhang, Guangyi Tand Etemad. Spatio-temporal eeg representation learning on riemannian manifold and euclidean space. *arXiv preprint arXiv:2008.08633*, 2020.

[108] Xiang Zhang, Lina Yao, Xianzhi Wang, Jessica Monaghan, David Mcalpine, and Yu Zhang. A survey on deep learning based brain computer interface: Recent advances and new frontiers. *arXiv preprint arXiv:1905.04149*, 66, 2019.

[109] Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun. Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38:43–54, 2017.

[110] Wei-Long Zheng and Bao-Liang Lu. Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks. *IEEE Transactions on autonomous mental development*, 7(3):162–175, 2015.

Curriculum Vitae

Name:	Sepehr Asgarian
Post-Secondary	Bacholer of Computer Engineering (2016-2021)
	Amirkabir University of Technology, Tehran, Iran
Awards:	BrainsCAN Scholar Issued by Canada First Research Excellence Fund (CFREF)
Experience	Teaching Assistant
	University of Western Ontario
	2021-Now
	Teaching Assistant
	Amirkabir University of Technology
	2017-2020