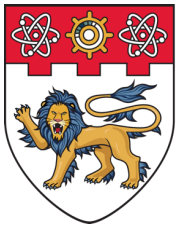
Geometric Methods for Covariance-Based Neural Decoding



NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Ju Ce (鞠策)

College of Computing and Data Science

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2024

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

22/Jan./2024

Date

Ju Ce (鞠策)

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

22/Jan./2024

Date

Prof. Guan Cuntai

Authorship Attribution Statement

This thesis contains material from five papers published in the following peer-reviewed journals/papers accepted at a conference where I am listed as the first author.

Chapter 5 is published as Ce Ju and Cuntai Guan, Tensor-CSPNet: A Novel Geometric Deep Learning Framework for Motor Imagery Classification, IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 12, pp. 10955-10969, Dec. 2023.
The contributions of the co-authors are as follows:

- Ce Ju designed the study, conceived the methodology, derived the formulae, wrote codes for the proposed approach, conducted experiments, prepared the manuscript drafts, and addressed reviewer comments.
- Cuntai Guan supervised the work and advised on the ideation, experimental design, writing, and manuscript review.

Chapter 6 are published as Ce Ju and Cuntai Guan, Graph Neural Networks on SPD Manifolds for Motor Imagery Classification: A Perspective from the Time-Frequency Analysis, IEEE Transactions on Neural Networks and Learning Systems, 2023. [1]
The contributions of the co-authors are as follows:

- Ce Ju designed the study, conceived the methodology, derived the formulae, provided theoretical analysis, wrote codes for the proposed approach, conducted experiments, prepared the manuscript drafts, and addressed reviewer comments.
- Cuntai Guan supervised the work and advised on the ideation, experimental design, writing, and manuscript review.

Chapter 7 is available as Ce Ju and Cuntai Guan, Deep Optimal Transport on SPD Manifolds for Domain Adaptation, under Review.
The contributions of the co-authors are as follows:

- Ce Ju designed the study, conceived the methodology, derived the formulae, provided theoretical analysis, wrote codes for the proposed approach, conducted experiments, prepared the manuscript drafts, and addressed reviewer comments.

[1] This is an early access article in IEEE Xplore, available in an electronic archive prior to its appearance in a regular issue of the journal.

- Cuntai Guan supervised the work and advised on the ideation, experimental design, writing, and manuscript review.

Chapter 8 is published as Ce Ju, Reinmar Josef Kobler, and Cuntai Guan, Score-Based Data Generation for EEG Spatial Covariance Matrices: Towards Boosting BCI Performance, the 45th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2023.
The contributions of the co-authors are as follows:

- Ce Ju designed the study, conceived the methodology, derived the formulae, wrote codes for the proposed approach, conducted experiments, prepared the manuscript drafts, and addressed reviewer comments.
- Reinmar Josef Kobler and Cuntai Guan provided comments on the writing.

Chapter 9 is published as Ce Ju†, Reinmar Josef Kobler†, Liyao Tang, Cuntai Guan, and Motoaki Kawanabe, Deep Geodesic Canonical Correlation Analysis for Covariance-Based Neuroimaging Data, the twelfth International Conference on Learning Representations, 2024.
The contributions of the co-authors are as follows:

- Ce Ju† designed the study, conceived the methodology, derived the formulae, provided theoretical analysis, wrote codes for the proposed approach, conducted parts of experiments, prepared the manuscript drafts (excluding the section on simultaneous EEG-fMRI tasks), performed overall text optimization, and addressed reviewer comments.
- Reinmar Josef Kobler† (Co-First Author) designed the study, supplied initial processing scripts for simultaneous EEG-fMRI data, optimized the codes, conducted experiments for the simultaneous EEG-fMRI task, contributed to writing the experimental section of the paper, co-performed overall text optimization, and co-addressed reviewer comments.
- Liyao Tang conducted experiments for the multi-view EEG task, provided results in Table 9.3, Table 9.4, Figure 9.12, and Figure 9.13, and proofread the manuscript.
- Cuntai Guan and Motoaki Kawanabe proofread the manuscript.

22/Jan./2024

Date

Ju Ce (鞠策)

Acknowledgements

The memory of attending differential geometry courses at Sun Yat-sen University, China over a decade ago remains vivid. As a second-year mathematics student, I was deeply fascinated by the beauty of mathematics and often spent time in the library reading lecture notes from renowned mathematicians. When I later returned to the university to work on neural decoding algorithms in brain-computer interfaces, I applied various geometric methods rooted in mathematics to analyze the covariance-based neural signals and develop novel decoding algorithms. This exploration led to a new technical perspective that became the foundation of this study. Although this journey diverged from my initial path in pure mathematics, it remains an exceptional experience in my life.

The concepts and tools presented in this thesis are the culmination of my experiences and efforts from diverse academic institutions and technology enterprises across China, the United States, Canada, Singapore, and Japan. I extend my sincere gratitude to the mathematicians, engineers, and entrepreneurs who generously shared their insights and provided invaluable assistance throughout this journey.

I am deeply grateful to my thesis advisor, Prof. Guan Cuntai, for his motivated and flexible advising style, which allowed me to pursue an innovative path by integrating my interests and expertise into this study. I provide special thanks to Prof. Guan Cuntai and the S-Lab at Nanyang Technological University, Singapore, for their financial support in this research. I also express my heartfelt gratitude to the members of my thesis advisory committee and the administrative staff of the College of Computing and Data Science at Nanyang Technological University for their patience and timely assistance, which significantly contributed to the smooth progress of my entire program.

Additionally, I acknowledge Dr. Reinmar Josef Kobler and Dr. Motoaki Kawanabe for their assistance and financial support from the Advanced Telecommunications

Research Institute International, Japan. This support facilitated an academic visit to their institution, allowing me to further explore cutting-edge topics along the research path proposed in this study.

Lastly, I extend my heartfelt gratitude to my family for their constant support and encouragement. The self-motivation, strength, and spirit passed down from my ancestors who worked in mathematics and the aerospace industry have been invaluable throughout this journey.

To the Brain-Computer Interface Society

Contents

Acknowledgements ix

Abstract xvii

List of Figures xvii

List of Tables xxvii

Symbols and Acronyms xxxi

1 Introduction 3
- 1.1 Backgrounds and Motivations . . . 3
- 1.2 Main Contributions . . . 5
- 1.3 Chapter Outline . . . 9

I Preliminaries 11

2 Motor Imagery Classification 13
- 2.1 EEG-Based Motor Imagery Classifiers . . . 15
- 2.2 Evaluation Scenarios and Datasets . . . 18
  - 2.2.1 MI-EEG Datasets . . . 18
  - 2.2.2 Simultenous EEG-fMRI Dataset . . . 22

3 Covariance-Based Neuroimaging Data 25
- 3.1 Riemannian Geometry . . . 27
- 3.2 Symmetric Positive Definite Manifolds . . . 31
- 3.3 Riemannian-Based Approaches . . . 33

4 Geometric Statistics and Geometric Methods 35
- 4.1 Geometric Statistics . . . 35
  - 4.1.1 Fréchet Mean . . . 35
  - 4.1.2 Principal Geodesic Analysis . . . 36
  - 4.1.3 Geodesic Regression . . . 37

4.2 Geometric Methods . . . 38
4.2.1 Geometric Deep Learning . . . 38
4.2.2 Manifold Learning . . . 42
4.2.3 Information Geometry . . . 42
4.2.4 Optimal Transport . . . 43

II Geometric Classifier and Its Applications 47

5 Tensor-CSPNet 49
5.1 Network Architecture . . . 50
5.2 Experimental Results . . . 53
5.2.1 Results of Classification Performance . . . 53
5.2.2 Results of Interpretability Analysis . . . 56
5.2.3 Results of Visualization . . . 58
5.3 Discussions . . . 58

6 Graph-CSPNet 63
6.1 Network Architecture . . . 64
6.2 Experimental Results . . . 70
6.2.1 Results of Classification Performance . . . 71
6.2.2 Ablation Study of Hyperparameters . . . 73
6.3 Discussions . . . 74

7 Optimal Transport-Domain Adaptation on SPD Manifolds 85
7.1 Methodology . . . 89
7.1.1 Optimal Transport-Domain Adaptation . . . 89
7.1.2 Deep Optimal Transport . . . 92
7.2 Experimental Results . . . 95
7.2.1 Deep Optimal Transport-Based Classifier . . . 95
7.2.2 Evaluation Settings . . . 97
7.2.3 Results of Synthetic Dataset . . . 99
7.2.4 Results of Real-World EEG Datasets . . . 100
7.3 Discussions . . . 104

8 Score-Based Data Generation for Spatial Covariance Matrices 109
8.1 Methodology . . . 110
8.2 Experimental Results . . . 111
8.2.1 Evaluation Settings . . . 111
8.2.2 Results of Visualization . . . 112
8.2.3 Results of Classification Performance . . . 119
8.3 Discussions . . . 121

9 Deep Geodesic Canonical Correlation Analysis 123

9.1 Prelimnary . . . 125
9.2 Methodology . . . 127
9.2.1 Neural Network-Based Solution . . . 131
9.2.2 Relaxed Orthogonal Projection . . . 131
9.2.3 Loss Function . . . 137
9.3 Experimental Results . . . 139
9.3.1 Simulations . . . 139
9.3.2 Simultaneous EEG-fMRI . . . 141
9.3.3 Multi-View EEG . . . 147
9.4 Discussions . . . 155

III Conclusions 157

10 Conclusions 159
10.1 Summary and Significance . . . 159
10.2 Future Directions . . . 160

IV Appendices 161

A Diffusion Model 163
A.1 Score-Based Generative Modeling . . . 163
A.2 Diffusing Samples through Stochastic Diffusion Equations . . . 164

B Canonical Correlation Analysis 167
B.1 Canonical Correlation Analysis . . . 167
B.2 Deep Canonical Correlation Analysis . . . 168
B.3 Riemannian Canonical Correlation Analysis . . . 168

List of Author's Publications 171

Bibliography 173

Abstract

Neuroimaging tasks present significant challenges in signal processing and analysis due to factors such as low signal-to-noise ratios, high non-stationarity, and limited dataset sizes. Furthermore, understanding brain dynamics is complicated by the coupling mechanisms across various neuroimaging modalities. To address these challenges, my study introduces an alternative approach by formulating covariance-based neuroimaging data on symmetric positive definite manifolds. I integrate various geometric methods to model this data and develop geometric deep learning frameworks for multiple neuroimaging tasks, including EEG-based motor imagery classification and the multimodal fusion of simultaneous EEG-fMRI data.

List of Figures

1 Illustration of a Geometric Brain-Computer Interface: This figure depicts a conceptual system architecture in which cyborgs control a robotic arm through a brain-computer interface. As cyborgs visualize actions such as grasping, neural signals generated in their brains are captured and transmitted to a computer via a signal decoder. The computer then applies geometric deep learning models on symmetric positive definite cones, equipped with the affine-invariant Riemannian metric, denoted by (S_{++}, g^{AIRM}), as proposed in this study, to govern the robotic arm's movements. 1

1.1 Figure (a) illustrates the signals collected by EEG/ECoG from the anatomical structures of the human head, while Figure (b) illustrates an EEG cap. 4
1.2 Flowchart of Chapters in this Thesis. 10

2.1 Figure (a) illustrates the primary motor cortex, responsible for controlling various motor movements, including those of the fingers, hands, wrists, and facial muscles on the opposite side of the body. Figure (b) shows the Penfield Homunculus, also known as the Penfield motor cortex topographic map, which graphically represents the organization and spatial layout of the motor cortex in the brain. 14
2.2 Simultaneous EEG-fMRI Dataset: Visualization of Cross-Validation Scenarios and Sliding Window Extraction. Left: The dataset includes two runs per subject, with each run consisting of eight trials lasting 30 seconds each (with no breaks in between). During these trials, subjects were either resting with their eyes open (EO) or eyes closed (EC). From each trial, we extracted one epoch of 20 seconds duration, with a 2-second offset. Middle: The extracted epochs were divided into cross-validation (CV) folds according to different scenarios. We utilized three CV strategies: 10-fold stratified, leave-one-run-out (LORO), and leave-one-subject-out (LOSO). Right: After creating the CV splits, sliding windows were extracted from each epoch. 24

5.1 Architecture of Tensor-CSPNet: In Line 1, EEG signals undergo tensor stacking, segmenting them into time-space-frequency tensors. Line 2 employs the BiMap, RieBN, and ReEig layers to effectively capture spatial nuances. In Line 3, temporal dynamics are addressed using two-dimensional CNNs on the tangent space. The final classification is performed by fully connected neural networks with a cross-entropy loss function. 50
5.2 Illustration of Temporal Convolutional Layer: F blocks of $1 \times o^2$ rectangles are flattened, and W lines are concatenated, i.e., $n_F = F$ and $n_W = W$. For instance, in the case of 5-CSPNet architecture, the values of n_F, o, and n_W are 9, 20, and 5, respectively. This implies that each line is a 1×3600 flattened tensor, and the entire rectangle's shape is 5×3600. 52
5.3 Heatmap Visualization of Relevance Patterns for 5-CSPNet$^{(9,1,1)}$. . 57
5.4 Visualization of Intermediate Outputs in 5-CSPNet$^{(9,1,1)}$ with $o = 22$ for Subject No.28 of the KU dataset Using t-SNE: The model processes EEG signals using time windows ranging from $1 \sim 1.5$ s, $1.5 \sim 2.0$ s, $2.0 \sim 2.5$ s, $2.5 \sim 3.0$ s, to $3.0 \sim 3.5$ s. Figure (a) illustrates the original EEG signals divided into 5 time windows using a segmentation strategy. Figure (b) displays the outputs from the common spatial pattern layer, where blue and yellow/green clusters correspond to two distinct motor imagery tasks. Each cluster is further subdivided into five sub-clusters corresponding to the individual time windows. Figure (c) shows the outputs following the temporal concatenation layer, where the blue and yellow/green clusters merge into two more cohesive clusters, with the yellow/green cluster positioned between the blue clusters. Finally, Figure (d) presents the outputs after further processing in the temporal concatenation layer, along with the associated class labels. Here, the blue and yellow/green clusters are separated by a decision boundary, with the two classes nearly symmetrically distributed on either side of this boundary. 60
5.5 Two-dimensional Projection of Subject No.28 in the KU dataset Using t-SNE: Each subject participates in two sessions. The first session (S1) is used as the training set, while the two halves of the second session (S2) are designated as the validation and test sets. The lengths of the time windows are (a) 2500 ms, (b) 500 ms, (c) 250 ms, and (d) 125 ms, with no overlap between them. Each point in the two-dimensional projection is obtained by reducing the dimensionality of a $9 \times 20 \times 20$-dimensional point, which includes 20 electrodes in the motor cortex region and spans nine frequency bands. This format serves as the input to Tensor-CSPNet. 61

6.1 Architecture of Graph-CSPNet: In this structure, the EEG signal is segmented into multiple divisions within the time-frequency domain. The spatial covariance matrices derived from these segments form the vertices of the time-frequency graph, which is constructed using an innovative nonparametric statistical method. After establishing the time-frequency graph, an SPD matrix-valued graph convolutional network is deployed. This network includes the Graph-BiMap layer and RieBN, which together extract crucial classification information while preserving the ability to distinguish between different task classes. Subsequently, the LOG layer maps the SPD matrices onto the tangent space. These transformed matrices are then fed into the cross-entropy layer for computations. 64

6.2 Classification performance of Graph-CSPNet with multiple time and frequency directions on the BNCI2014001 (BCIC-IV-2a) dataset. The (forward) time direction includes $0 \leq x_\theta \leq 4$, $0 \leq x_\mu \leq 4$, $0 \leq x_\beta \leq 4$, and $0 \leq x_\gamma \leq 8$. The frequency direction is preset to (1,1,4,3). 74

6.3 Spectrum Distribution Shift: In the KU dataset, a two-dimensional projection of Subject No.1 using t-SNE is displayed. Each session for the subject contains 200 trials in total. In each subfigure, the bright red point represents the 200th trial in that session, while 199 green points depict the first 199 trials. The points have been dimensionally reduced from 20×20-dimensional spatial covariance matrices through t-SNE. The 60 dark red points symbolize the spatial covariance matrices derived from 60 EEG segments, as specified in the segmentation plan in Table 6.1, of the last trial (represented by the bright red point), while the 11940 blue points ($11940 = 199 \times 60$) represent the spatial covariance matrices derived from the remaining 199 trials. 75

6.4 Discrete Spectrograms of Variant Configuration Time-Frequency Graphs: The LGT method has four *time direction* numbers representing the forward steps in the components of θ, μ, β, and γ. The *frequency direction* numbers are always set to Time Direction (1, 1, 4, 3). The discrete spectrograms, ranging from (a) to (c), are calculated by evaluating the lower frequency band's spectrum power on the grids ($4 \sim 16$ Hz) across every five grids on the time axis, with a grid width of 500 ms and a height of 4 Hz. The higher frequency band ($16 \sim 40$ Hz) is calculated across each grid, with a grid width of 250 ms and a height of 4 Hz. Spectrogram (a) represents the original spectrum distribution of Subject No. 1 in the KU dataset, while spectrograms (b) and (c) are the spectrum distributions after the LGT method on (a). 76

6.5 Variant Configurations of Time-frequency Graphs: The time-frequency graphs (a) and (b), which are derived from spectrograms (b) and (c), respectively, in Figure 6.4, contain 60 nodes and 390 edges. Each node represents a grid or a couple of grids, with low time resolution for the low frequency (4 to 16 Hz) and high time resolution for the high frequency (16 to 40 Hz). The spectrum of a node evaluates adjacent nodes along the time axis and consists of four graph components corresponding to four frequency bands, i.e., θ, μ, β, and γ. The edge weight of each two adjacent nodes is the geodesic distance between the two points on $(\mathcal{S}_{++}, g^{AIRM})$ and is reconstructed using the multidimensional scaling algorithm. 77

7.1 Illustration of Deep Optimal Transport: Multi-channel signals are first transformed into covariance matrices, placing these matrices on $(\mathcal{S}_{++}, g^{LEM})$. The SPD matrix-valued data is then processed through the BiMap, ReEig, and LOG layers, mapping it to the tangent space at the identity. The loss function incorporates cross-entropy loss, marginal distribution adaptation, and conditional distribution adaptation. 92

7.2 Illustration of DOT-Based Motor Imagery Classifier: The proposed classifier simultaneously transports the source and target Fréchet means in each frequency band to a common subspace within the same frequency band. Assumption 7.2 ensures that each frequency component contributes equally to the classification. Assumption 7.3 allows each pair of transformations to occur within the same EEG frequency band. 96

7.3 Illustrations for experimental settings on three datasets: (a). KU; (b). BNCI2014001; (c). BNCI2015001. 97

7.4 Synthetic Data of 2-dimensional SPD Cone: The subfigures from left to right and top to bottom are labeled as (a) to (d). In subfigures (a) to (d), the red points represent the source domain, while the blue points indicate their locations after the transformation. 101

8.1 The sampling procedure can be depicted subsequently: Initially, we possess a noise matrix, represented as X_T. By employing the score-based generative modeling technique, we synthesize the spatial covariance matrix, X_0, with an intermediate state denoted as X_t. The generated spatial covariance matrix X_0 is approximated as being nearly SPD due to the acquired knowledge. To counterbalance numerical inconsistencies, we ensure the projection of X_0 as SPD by imposing a threshold upon all eigenvalues, denoted as $\epsilon = 1e - 4$. The arrangement of spatial covariance matrice channels proceeds sequentially from beginning to end: FC-5/3/1/2/4/6, C-5/3/1/z/2/4/5, and CP-5/3/1/z/2/4/6. 110

8.2 Illustration of the raw and generating distributions of two-dimensional projections of EEG spatial covariance matrices. The Fréchet means of both distributions are marked with triangle and cross signs. . . 114
8.3 Illustration of Fréchet means of spatial covariance matrices in the raw dataset within all the nine frequency bands. 115
8.4 Illustration of Fréchet means of spatial covariance matrices derived from the left and right-hand trials in the raw dataset within the nine frequency bands. The highlight entities of spatial covariance matrices in subfigure 8.4a (Mu and Beta bands) are located in the regions of FC4, C4, and CP4 over the scalp, while those in subfigure 8.4b fall in the regions of FC3, C3, and CP3. 116
8.5 Illustration of Fréchet means of covariance matrices within the nine frequency bands for the left and right-hand trials in the generating dataset within the nine frequency bands. The highlight entities of spatial covariance matrices in subfigure 8.5a (Mu and Beta bands) are located in the regions of FC4, C4, and CP4 over the scalp, while those in subfigure 8.5b fall in the regions of FC3, C3, and CP3. . 117
8.6 Conditional EEG spatial covariance matrice generation: In each line, we plot a picked spatial covariance matrix derived from an actual EEG segment for a category and another two generated samples within the same class. 118

9.1 Deep Geodesic CCA (DeepGeoCCA) framework. We consider paired views of multivariate time-series data (e.g., simultaneously recorded EEG and fMRI data). Individual views are transformed (e.g., covariance matrices) to form paired views on SPD manifolds $\{(S_i, \bar{S}_i)|S_i \in \mathcal{S}_{++}^{n_1}, \bar{S}_i \in \mathcal{S}_{++}^{n_2}\}_{i=1}^{N}$. A *Dimension Adaptation* module transforms the views so that they reside within $\mathcal{S}_{++}^{n}$, a cone in $\mathbb{R}^{n\times n}$. On this cone, a *Transformation* module learns to center the data around e and project them onto geodesics $\exp_e(tv)$ and $\exp_e(tw)$ to form latent SPD representations $\{(Z_i, \bar{Z}_i)\}_{i=1}^{N}$. Our primary objective is to maximize geodesic correlation, which measures correlation along the speeds v and w, after *Tangent Space Projection* at the identify matrix e. 124
9.2 Geodesic Correlation on $\mathcal{S}_{++}^{n}$. 128

9.3 Illustration of Correlation, Riemannian Correlation, and Geodesic Correlation: From left to right, the concepts of Correlation, Riemannian Correlation, and Geodesic Correlation are depicted with specific descriptions as follows: 1). Correlation: This measures the relationship between paired data without requiring proximity between the pairs.; 2). Riemannian Correlation: This measure computes the correlation between paired projection data on the tangent spaces of each modality, considering their respective Fréchet means. The projection data is obtained by projecting the raw data onto geodesic submanifolds; 3). Geodesic Correlation: This measure calculates the correlation between paired projection data in the tangent space at the identity point e. The projection data is obtained by projecting the raw data onto or near geodesics. 129
9.4 Illustration of Relaxation Deviation: The yellow region represents the double cone, where the maximum cosine value of the included angles is ε. The region within the red dashed lines denotes the tubular neighborhood of width δ along the speed of geodesic u on the tangent space. 133
9.5 Simulations on $\mathcal{S}^2_{++}$. **a**, Generative process and visualization of generated observations ($N = 500$) with a shared latent source. **b**, Visualization of the latent representations learned with DeepGeoCCA ($\varepsilon = 0.75$). **c**, Boxplots summarize the simulation results for test data (10-fold CV). DeepGeoCCA can obtain high geodesic correlation (left panel; higher is better) and at the same time approx. constrain the representations to geodesics (right panel; coefficient of determination R^2; mean across modalities; higher is better). . . 139
9.6 Illustration of network architecture for the simulations with paired SPD manifold-valued data. The BiMap layers used unconstrained parameters. For the ReiEig layers, we used a threshold of 1.0. The tangent space projection module used an SPDMBN layer without learnable parameters, i.e., the rebias parameter was fixed at the identity matrix e and the scaling parameter was fixed to 1.. 141
9.7 Illustration of Network Architecture for the EEG-fMRI experiment. For EEG views $\bar{S}$, we used either Tensor-CSPNet or TSMNet. For the ReiEig layer within Tensor-CSPNet, we used a threshold of 0.5 and unconstrained parameters in BiMap. For TSMNet, we used standard parameters except for 10 latent SPD dimensions instead of 20. The tangent space projection module used a domain-specific SPDMBN layer without learnable parameters. For fMRI views $\bar{S}$, we use a standard single Transformer-encoder layer as implemented in the torch. 142

9.8 DeepGeoCCA for simultaneous EEG-fMRI data. Using simultaneously recorded observations $(S_i, \bar{S}_i)$, we aim to learn a latent space where brain dynamics $(Z_i, \bar{Z}_i)$, shared between EEG and fMRI, covary with high-congruence. After preprocessing and extracting sliding windows, we use established decoder models to convert fMRI time-series activity [1] and oscillatory EEG activity [2, 3] to latent representations. t-SNE visualizations (perplexity=30) summarize input and latent data distributions after model fitting. 144

9.9 Graphical representation of the results summarized in Table 9.3 for the correlation metric on test set data. Each dot summarizes a CV split result. (**left**) 10-fold CV scenario. (**middle**) LORO scenario (16 runs=folds). Red dots highlight fold for which the test set was sampled from outlier subjects (subjects 1 and 5). (**right**) LOSO scenario (8 subjects = folds). 146

9.10 Illustration of network architecture for the KU dataset: The spatial covariance matrices from both the 2-channel EEG setup (i.e., C3 and C4) and the 20-channel EEG setup (i.e., FC-5/3/1/2/4/6, C-5/3/1/z/2/4/5, and CP-5/3/1/z/2/4/6) are transformed into a common dimension using the Dimension Adaptation Network. During the training of the self-supervised learning (indicated by the solid black lines), the Transformation Network and Projection Head retain the parameters in the neural networks, and geodesic correlation is calculated. The dashed red lines represent the downstream task classifier. Specifically, in the upper right corner of Riemannian BN*, the asterisk $*$ denotes that we use a specially designed Riemannian BN to perform center-parallel translation of outputs to point e. . . 148

9.11 Illustration of network architecture for the BNCI2015001 dataset: The spatial covariance matrices for the 2-channel EEG setup are derived from FCz and C3, while for the 20-channel EEG setup, they are derived from FC-3/z/4, C-5/3/1/z/2/4/6, and CP-3/z/4. The description of this architecture is consistent with it provided in Figure 9.10. 149

9.12 This figure displays the average accuracy changes (%) for each subject in the KU dataset across two 10-fold cross-validation scenarios S_1 and S_2, and a holdout scenario $S_1 \mapsto S_2$. Subjects with larger improvements are positioned at the top, arranged in descending order from top to bottom. Green represents an increase, while blue represents a decrease. Corresponding increases are labeled with the respective subject number. 151

9.13 This figure displays the average accuracy changes (%) for each subject in the BNCI2015001 dataset across two 10-fold cross-validation scenarios S_A and S_B, and a holdout scenario $S_A \mapsto S_B$. Subjects with larger improvements are positioned at the top, arranged in descending order from top to bottom. Green represents an increase, while blue represents a decrease. Corresponding increases are labeled with the respective subject number. 152

List of Tables

1.1 Categories of EEG-Based Motor Imagery Classifiers. 8

2.1 Summary of Dataset Settings for Five Selected Publicly Available MI-EEG Datasets. In the experiments, we exclusively use EEG electrodes from the primary motor cortex of the subjects. The original dataset settings are provided in parentheses for reference. 21

3.1 Operator on SPD manifolds equipped with g^{AIRM} and g^{LEM}: $S_1, S_2, P \in \mathcal{S}_{++}$ and $v, w \in \mathcal{T}_P\mathcal{S}_{++}$; exp and log are the matrix exponential and the logarithm; Tr is the trace of a matrix; $D\log$ and $D\exp$ are the derivatives of log and exp, respectively. 33

5.1 Notations for Hyper-Parameters in w-CSPNet$^{(m,n,l)}$ $@(p,q,r)$. . . . 53

5.2 Aerage accuracies and standard deviations for subject-specific analyses on the KU, BNCI2014001, BNCI2014002, BNCI2015001, and Cho2017 datasets. Each entry in the table is presented as average accuracy (standard deviation), with the highest-performing value highlighted in bold. 55

5.3 The outcomes in the holdout scenario of the KU dataset with varying q values in the temporal concatenation layer of 5-CSPNet and 10-CSPNet. The parameter q represents the height of two-dimensional CNNs within the temporal concatenation layer. 59

6.1 A non-overlapping and non-uniform segmentation plan for Graph-CSPNet. The table provided lists each frequency band's time-window length (seconds) as a distinct entity. 72

6.2 Average accuracies and corresponding standard deviations derived from subject-specific analyses of the KU, Cho2017, BNCI2014001, BNCI2014002, and BNCI2015001 dataset. Each result in the table is expressed as the average accuracy accompanied by its corresponding standard deviation. Notably, the optimal outcome for each analysis is highlighted in boldface, thus providing an enhanced visual representation of the best-performing metrics. 72

6.3 Parameters in a two-layer Graph-CSPNet with input tensor shape (N, o_1, o_1). 81

6.4 Comparison between Tensor-CSPNet and Graph-CSPNet. 81

7.1 History of the OT-DA Framework. 88
7.2 Comparative experiment table for three public datasets: All results in the table are in percentages (%), and the comparison methods include four categories, nine methods, and the corresponding three base classifiers. The original results of the base classifiers without considering transfer methods are at the bottom of the table. In the table, we use the symbol "/" to indicate that this method does not exist in the scenario constructed by this dataset. 103
7.3 Average Log-Euclidean distances between the Fréchet means of EEG spatial covariance matrices generated from different frequency bands in the source and target domains for the KU (54 subjects), BNCI2014001 (9 subjects), and BNCI2015001 (12 subjects) datasets. The shortest average Log-Euclidean distance in each row is highlighted in boldface. 107

8.1 Confusion Matrix: Predicted labels in a total of 8400. 119
8.2 Cross-session classification with data augmentation approach: Each column depicts the number of samples incorporated into the training session. The samples are divided equally between the two classes: left-hand and right-hand. The selected cross-session scenario originates from the training and evaluation sessions in the KU dataset. The initial session of 200 trials and the added samples serve as the training data, while the first half of the second session, comprising 100 trials, is utilized for validation and the latter half, consisting of 100 trials, for testing purposes. The results (%) presented encompass the mean of 10 times runs across all scenarios and the optimal performance. 120

9.1 Difference between Correlation, Riemannian Correlation, and Geodesic Correlation 130
9.2 Simultaneous EEG-fMRI dataset results. As before, test-set model performance (higher is better) is evaluated with correlation and R^2 metrics across 10-fold CV with stratification across subjects and runs. Exhaustive permutation t-tests (df=9, 7 tests with t-max adjustment) were used to identify significant differences between *TSMNet+DeepGeoCCA* and baseline methods. 145
9.3 Simultaneous EEG-fMRI dataset results. Additional test-set results for the EEG-fMRI dataset that extend Table 9.2 with regard to the considered evaluation scenario. In addition to the 10-fold CV scenario results reported in Table 9.2, this table also summarizes results for leave-one-run-out (LORO) and leave-one-subject-out (LOSO) CV scenarios. For LORO and LOSO, we report summary statistics (mean and std) with (w/) and without (w/o) outlier subjects (i.e., subjects 1 and 5). 146

9.4 Multi-View EEG downstream results. Average (std across subjects in brackets) classification accuracy for the motor imagery task (higher is better). We considered two public datasets and scenarios: within-session 10-fold CV (sessions S_1 and S_2) and a hold-out scenario (train S_1; test S_2). For the KU and BNCI2015001 datasets, the full-channel EEG views comprise 20 and 13 channels covering the sensorimotor cortex. The network architecture was Tensor-CSPNet. 153

9.5 Simultaneous EEG-fMRI dataset results. Statistical test results for downstream test set accuracy differences in the multi-view EEG experiment, extending Table 9.4. We used permutation, paired t-tests to identify significant differences between the lower bound (i.e., no pre-training; 2 channel EEG) with DeepGeoCCA for three scenarios (S_1, S_2, $S_1 \rightarrow S_2$; t-max adjustment for multiple comparisons) and the KU dataset (df=53, 1e4 permutations) and the BNCI dataset (df=11, exhaustive permutations). Significant differences are highlighted in bold (p-val $\leq$ 0.05) and trends in italic (p-val $\leq$ 0.1). 154

9.6 Ablation Studies: Average (std across subjects in brackets) classification accuracy for the motor imagery task (higher is better). We considered the KU dataset with scenario 10-fold CV (sessions S_1 and S_2). 155

9.7 Comparisons between different CCAs. 156

Symbols and Acronyms

Symbols

$\odot$	the Hadamard (component-wise) product.
$\otimes$	the Kronecker product.
$\mathbf{0}$	all-zeros column vector with proper dimension.
$\mathbf{1}$	all-ones column vector with proper dimension.
$\mathbb{R}^n$	the n-dimensional Euclidean space.
$\langle \cdot , \cdot \rangle$	the inner product of two vectors.
$\| \cdot \|_{\ell_2}$	the ℓ_2-norm of a vector or matrix in Euclidean space.
$\| \cdot \|_{\mathcal{F}}$	the Frobenius norm of a vector or matrix in Euclidean space.
$\mathcal{L}^2(\Omega)$	the space of square integrable $\mathcal{L}^2$ functions on subset $\Omega \subset \mathbb{R}^n$.
$\mathcal{M}^n$	the $n-$dimensional Riemannian manifold.
$T_p\mathcal{M}^n$	the tangent space of Riemannian manifold $\mathcal{M}$ at p.
$\mathcal{S}_{++}$	the space of symmetric and positive definite matrices.
g^{AIRM}	the affine invariant Riemannian metric.
g^{LEM}	the Log-Euclidean metric.
d_g	the Riemannian distance with respect to Riemannian metric g.
$\mathcal{D}_S$ and $\mathcal{D}_T$	the source domain and the target domain.
$\mathbb{P}(X)$	the marginal probability distribution on feature space X.
$\mathbb{Q}(Y\|X)$	the conditional probability distribution of label space Y conditioned on feature space X.
$\mathcal{T}$	the learning task.
$X \in \mathbb{R}^{n_C \times n_T}$	the EEG segment with the number of channels n_C and the number

	of timestamps n_T.
$S := XX^\top \in \mathbb{R}^{n_C \times n_C}$	the EEG spatial covariance matrix with the number of channels n_C.

Acronyms

BCI	Brain-Computer Interface
EEG	Electroencephalography
fMRI	Functional Magnetic Resonance Imaging
ECoG	Electrocorticography
MI	Motor Imagery
SMR	Sensorimotor Rhythm
ERD/ERS	Event-Related Desynchronization/Synchronization
SNR	Signal-to-Noise Ratio
CSP	Common Spatial Pattern
FBCSP	Filter Bank Common Spatial Pattern
MDRM or MDM	Minimum Distance to Riemannian Mean
TSM	Tangent Space Mapping
RPA	Riemannian Procrustes Analysis
RCT/ROT	Recentering/Rotation in PRA
DL	Deep Learning
CNN	Convolutional Neural Network
RieBN	Riemannian Batch Normalization
SPDMBN	Symmetric Positive Definite Momentum Batch Normalization
GDL	Geometric Deep Learning
OT	Optimal Transport
DA	Domain Adaption
CCA	Canonical Correlation Analysis
CV	Cross Validation
LORO	Leave-One-Run-Out
LOSO	Leave-One-Subject-Out

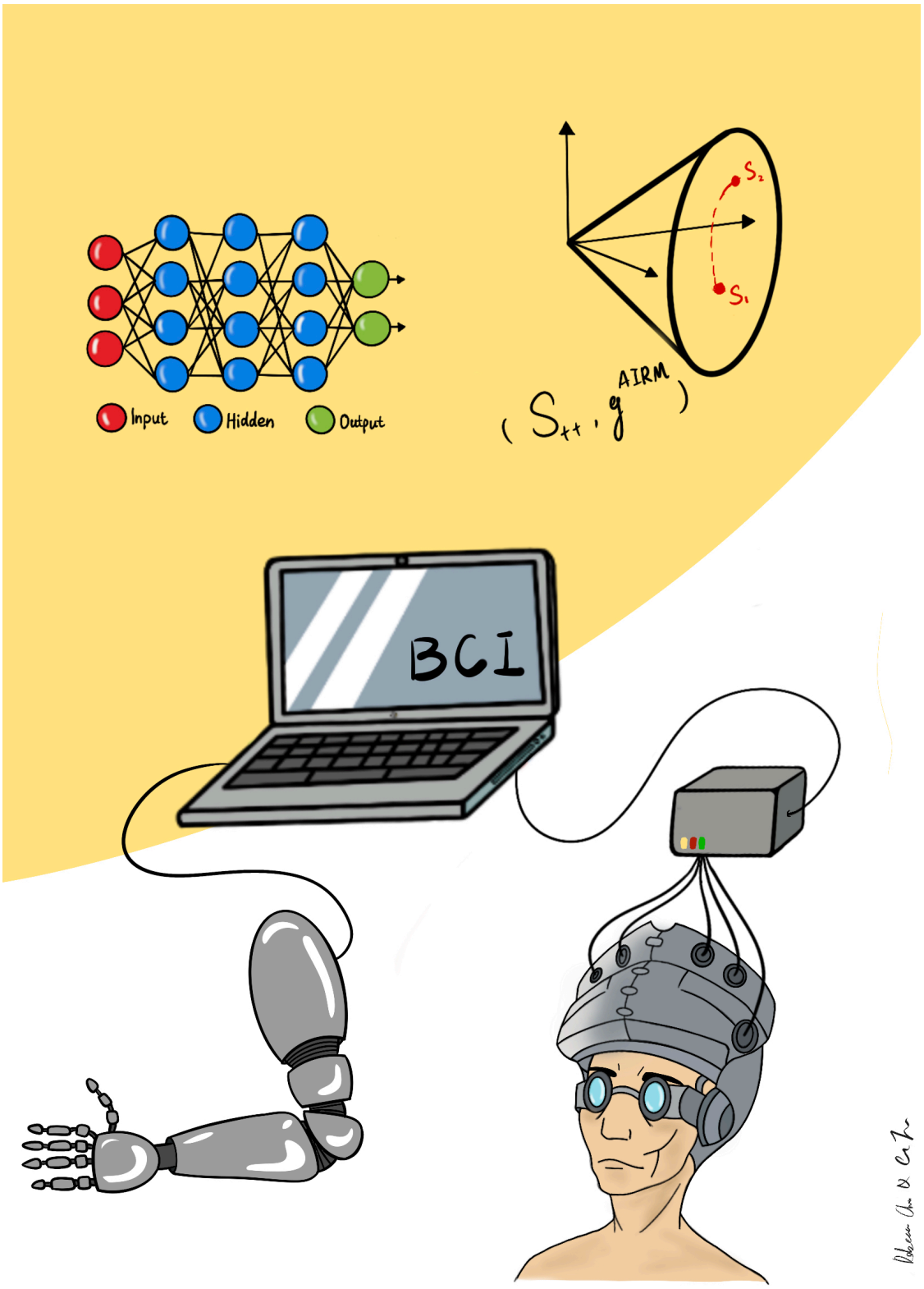


Figure 1: Illustration of a Geometric Brain-Computer Interface: This figure depicts a conceptual system architecture in which cyborgs control a robotic arm through a brain-computer interface. As cyborgs visualize actions such as grasping, neural signals generated in their brains are captured and transmitted to a computer via a signal decoder. The computer then applies geometric deep learning models on symmetric positive definite cones, equipped with the affine-invariant Riemannian metric, denoted by (S_{++}, g^{AIRM}), as proposed in this study, to govern the robotic arm's movements.

Chapter 1

Introduction

This study explores a novel pathway of neural decoding algorithms by integrating modern mathematical and statistical methods with advanced artificial intelligence models to enhance classification tasks in brain-computer interfaces.

1.1 Backgrounds and Motivations

A Brain-Computer Interface (BCI) is a technology that captures and decodes neural signals from brain activity, facilitating communication between the central nervous system and the external environment. Among the various types of BCIs, electroencephalography (EEG)-based systems are the most widely used due to their portability and cost-effectiveness. These EEG-based BCIs hold significant potential across a range of applications, including stroke rehabilitation, wheelchair control systems, and video gaming [4–9].

EEG-based BCI applications function by monitoring changes in EEG rhythms over the brain's sensorimotor areas. These changes in sensorimotor rhythms (SMRs) related to motor imagery (MI) can serve as effective control signals for BCIs [10–12]. During the planning and execution of movement, SMRs undergo amplitude changes known as event-related desynchronization/synchronization (ERD/ERS) effects [13]. ERD reflects a decrease in rhythmic activity, while ERS indicates an increase. These patterns can be reliably detected, enabling accurate classification of EEG signals. In an EEG-based MI task, the subject mentally simulates

physical movement, engaging the brain's sensorimotor systems and primary sensorimotor areas [14, 15]. The EEG device records the resulting signals, which are then interpreted by machine learning classifiers to determine the subject's intentions. The locations for signal acquisition by BCI devices (EEG/ECoG) are shown in Figure 1.1 [1].

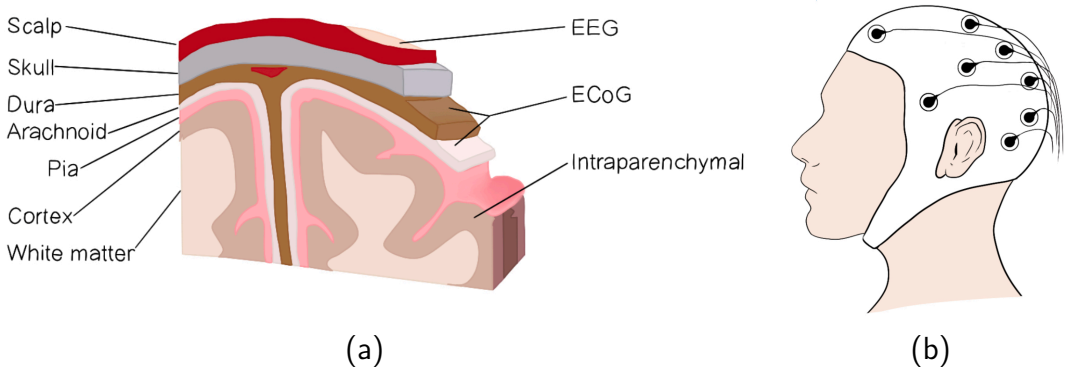


Figure 1.1: Figure (a) illustrates the signals collected by EEG/ECoG from the anatomical structures of the human head, while Figure (b) illustrates an EEG cap.

EEG-based motor imagery classification has yet to achieve satisfactory performance, despite the application of various classifiers. Several factors contribute to this suboptimal outcome. First, volume conduction, where electrical activity spreads from the brain's neural sources and is detected at distant points on the scalp or body surface, leads to the combination of signals from different sources. This results in blurred or diffused signals, making it difficult to accurately identify specific neural activities. Second, significant variability in EEG signals occurs within a session, between sessions, and across different subjects. These variations can be attributed to factors such as learning effects, differences in system calibration, the subject's mental state, electrode placement, and physiological differences like head size. Additionally, the limited size of EEG and other neuroimaging datasets poses a challenge. Due to the high cost and time-intensive nature of recording, these datasets are generally small, which limits the effectiveness of deep learning models typically trained on large-scale datasets.

[1] Figure 1.1 (a) has been adapted with the illustration redrawn and includes minor modifications, based on the original from https://www.neurotechcenter.org/research/brain-computer-interfacing.

Moreover, it's important to recognize that motor imagery classification is more complex than it initially appears. The complexity arises from various challenges and the absence of reliable measurements to assess their impact. While many existing methods are adapted from other engineering domains where similar issues have been addressed, this pragmatic approach often falls short of fully explaining the occasional inefficacy of these methods.

1.2 Main Contributions

Recently, EEG-based motor imagery classifiers have increasingly shifted toward deep learning models, resulting in significant performance improvements [16–18]. Deep learning offers robust representation capabilities, enhancing the effectiveness and resilience of neural network-based approaches [19]. Traditionally, these classifiers have relied on convolutional neural networks (CNNs) to process multi-channel EEG time series data. However, since EEG time series fundamentally differ from image data, it raises the question of whether CNNs, originally designed for image processing, are truly optimal for EEG classification.

Given the widespread use of EEG spatial covariance matrices in traditional motor imagery classifiers, integrating these matrices into deep learning models is a logical step forward. This approach introduces a new class of classifiers specifically designed for motor imagery tasks. The symmetric and positive definite nature of EEG spatial covariance matrices necessitates a distinct network architecture, different from those used in general classification tasks.

These specialized neural networks [20–22], initially developed for computer vision, are particularly well-suited to manage the unique characteristics of EEG data. By integrating the mathematical properties of symmetric and positive definite (SPD) matrices with deep learning techniques, these networks can effectively capture complex patterns within EEG signals. This integration has led to the development of innovative classifiers, known as geometric classifiers, specifically designed to the challenges of EEG-based motor imagery classification.

From a mathematical perspective, the space of SPD matrices is typically characterized as a manifold in Riemannian geometry. This geometric formulation ensures that the transformations and operations performed on SPD matrices remain within

the manifold, thereby preserving the mathematical properties of the data. Consequently, learning problems related to EEG spatial covariance matrices naturally adopt the framework of geometric statistics, which is designed to handle the statistics of manifold-valued data. This geometry effectively models EEG signal variance, specifically the spectral power of the EEG signal, which is crucial for EEG-based motor imagery classification.

This study makes contributions by introducing geometric classifiers that integrate advanced geometric methods on SPD manifolds and applying them across various scenarios to address key challenges in various neuroimaging tasks. The main contributions are summarized as follows:

- Introduction of Geometric Classifiers: Chapter 5 and 6 introduce two novel geometric classifiers that utilize neural networks on SPD manifolds to extract discriminative features from EEG spatial covariance matrices. These classifiers are designed for general EEG-based motor imagery classification.
- Addressing Cross-Session Variability: Chapter 7 addresses the challenge of cross-session variability in EEGs by extending the optimal transport-domain adaptation framework to SPD manifolds equipped with the Log-Euclidean metric. This adaptation improves the handling of data distribution shifts across sessions for manifold-valued data.
- Overcoming Limited Data: Chapter 8 explores the innovative use of score-based generative models to synthesize EEG spatial covariance matrices. This approach enhances the practical utility of limited data in motor imagery tasks, thereby improving the classification performance of geometric classifiers.
- Multimodal Fusion of Simultaneous EEG-fMRI Data: Chapter 9 presents a self-supervised framework designed to integrate multimodal covariance-based neuroimaging data, particularly simultaneous EEG-fMRI data. This framework significantly improves the performance of geometric classifiers in downstream applications by leveraging the learned representations of both modalities and maximizing geodesic correlation through the use of a geometric deep learning model.

Table 1.1 provides an overview of the mainstream EEG-based motor imagery classifiers from the past twenty years, highlighting the evolution from common spatial patterns and Riemannian-based approaches to convolutional neural network models. This study introduces a novel approach that integrates key elements from these three categories: BiMap transformation, SPD manifolds, and deep learning, as illustrated in the table. The history, mechanisms, and detailed exploration of these mainstream classifiers, along with a comprehensive discussion of the proposed innovative pathway, will be thoroughly examined in Chapters 2 and 3.

Table 1.1: Categories of EEG-Based Motor Imagery Classifiers.

Category	Proposition	Input	BiMap Transformation	SPD Manifolds	Deep Learning	Mechanism
Common Spatial Pattern	1999 [23]	Covariance	✓	✗	✗	Search for a linear projection direction that maximizes the discrimination between classes by solving Eigenvalue Problem 2.1.
Riemannian-Based Approach	2011 [24]	Covariance	✗	✓	✗	Compare the Riemannian distance, which includes the entire spectrum information obtained from Eigenvalue Problem 2.1 without directly solving it.
Convolutional Neural Network	2017 [16–18]	Time series	✗	✗	✓	Extract discriminative features using the convolutional neural network -based architecture.
Geometric Classifier (Proposed)	2022 †[2, 25]	Covariance	✓	✓	✓	Seek a nonlinear projection on Stiefel manifolds that maximizes the discrimination between classes using neural network-based models.

† To the best of our knowledge, the first instance of using SPD matrix-valued neural networks for processing EEG spatial covariance matrices in EEG-based motor imagery classification appeared in my previous study [26]. However, the broader introduction of this innovative class for general EEG-based motor imagery classification started with the publication of Tensor-CSPNet [2].

1.3 Chapter Outline

This thesis is organized into four main parts. The first part consists of three chapters that establish the foundational concepts for further exploration. Chapter 2 reviews the mechanisms of EEG-based motor imagery classification and the mainstream classifiers, providing context for a more detailed examination of these techniques. Chapter 3 introduces the concepts of covariance-based neuroimaging data, Riemannian geometry, SPD manifolds, and a new class of classifiers known as Riemannian-based approaches. Chapter 4 discusses geometric statistics and various geometric methods, including geometric deep learning, manifold learning, information geometry, and optimal transport on manifolds, all of which are crucial to the methodologies presented in this thesis. The second part, titled Geometric Classifiers and Their Applications, comprises five chapters, each focusing on one of my research contributions. The third part presents the conclusions and outlines valuable directions for future research. The final part (appendix) provides a brief overview of the diffusion model and canonical correlation analysis, which are relevant to the discussions in Chapters 8 and 9, respectively.

Figure 1.2 presents the flowchart of chapters in this thesis. Chapters 2 through 4 lay the groundwork, providing the necessary background for the research in Chapters 5 through Chapter 9. Chapters 8 and 9 apply the geometric classifiers developed in Chapters 5 and 6 to specific tasks, while Chapter 7 follows an independent trajectory. This distinction is made because Chapters 5, 6, 8, and 9 utilize the affine-invariant Riemannian metric on SPD manifolds, whereas Chapter 7 employs the Log-Euclidean metric.

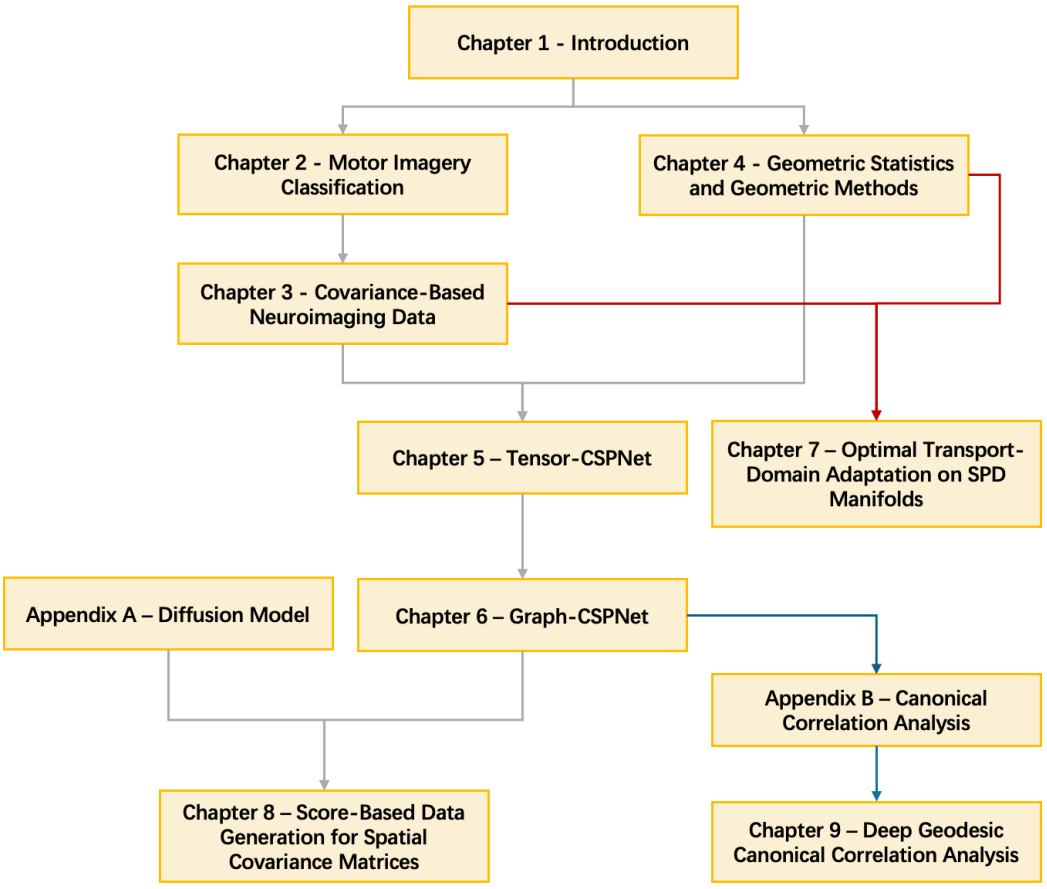


Figure 1.2: Flowchart of Chapters in this Thesis.

Part I

Preliminaries

Chapter 2

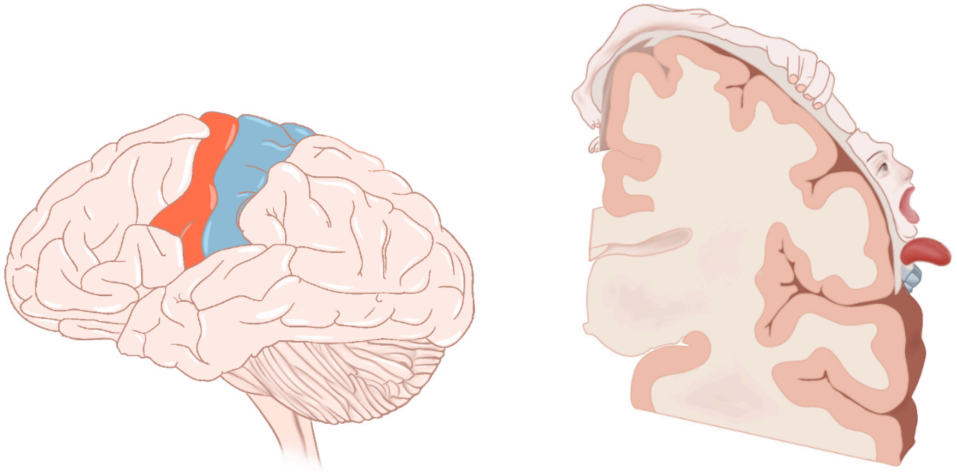
Motor Imagery Classification

Motor imagery involves the mental simulation of a specific action without physically performing it [27, 28]. This cognitive process engages the same neural regions responsible for actual movement control. Notably, motor imagery has demonstrated therapeutic benefits for individuals with neurological disorders, strokes, or spinal cord injuries by leveraging brain pathways associated with imagined movements to support recovery [29, 30]. Moreover, it enhances physical performance, making it a valuable tool in sports training and physical therapy [31, 32].

From a neurophysiological perspective, the Penfield Homunculus, shown in Figure 2.1 (b), has significantly contributed to the development of BCI technology for motor imagery tasks. Motor imagery tasks generate specific patterns of brain activity that can be accurately detected using EEG [11, 33]. This EEG data can then be used to control devices by identifying and interpreting the brain activity linked to motor imagery [34–36]. Figure 2.1 illustrates the spatial organization of the motor cortex in the brain. [1]

Motor imagery engages multiple brain regions, including the supplementary motor area, prefrontal cortex, premotor cortex, cerebellum, and basal ganglia. This has been validated by studies utilizing positron emission tomography, functional magnetic resonance imaging, and electroencephalography [14, 37, 38].

[1] The two subfigures have been adapted, with illustrations redrawn and minor modifications, based on the original from `https://www.kenhub.com/en/library/anatomy/brodmann-areas`.



(a) Primary Motor Cortex. (b) Penfield Homunculus.

Figure 2.1: Figure (a) illustrates the primary motor cortex, responsible for controlling various motor movements, including those of the fingers, hands, wrists, and facial muscles on the opposite side of the body. Figure (b) shows the Penfield Homunculus, also known as the Penfield motor cortex topographic map, which graphically represents the organization and spatial layout of the motor cortex in the brain.

Motor imagery is closely associated with event-related desynchronization (ERD) and synchronization (ERS) effects in specific frequency bands. ERD is characterized by a temporary, localized reduction in rhythmic EEG activity, often signaling increased brain activity in the relevant area. Conversely, ERS is marked by a transient, localized increase in rhythmic EEG activity, which typically indicates decreased brain activity or relaxation in that area. Both ERD and ERS are temporally linked to the event but occur out of phase and are specific to certain frequency bands [10, 12, 39].

Research consistently shows that motor imagery induces ERD/ERS effects in the Mu (8-13 Hz) and Beta (13-30 Hz) bands, each with unique temporal characteristics [40]. Mu band activity generally decreases gradually over several seconds before returning to baseline, while Beta band activity shows a burst following the ERD [40]. However, extensive research indicates that brainwaves exhibit unique spatial, temporal, and frequency characteristics for each individual [11, 40]. Therefore, the most effective approach to developing a motor imagery classifier is to account for neurophysiological differences across time, space, and frequency domains, a concept referred to in this thesis as the time-space-frequency principle.

2.1 EEG-Based Motor Imagery Classifiers

Feature engineering approaches are commonly employed in the early stages of motor imagery classifiers. This process involves designing hand-crafted features based on time-domain and frequency-domain statistics, such as band power, magnitude-squared coherence, and phase-locking value [41–43]. After extracting these features, traditional machine learning techniques, including support vector machines [44] and linear discriminant analysis [45], are often used for classification tasks.

Spatial filtering plays a crucial role in feature extraction during this phase, significantly enhancing the signal-to-noise ratio. This enhancement is essential because localized brain activities are spatially dispersed when transmitted through the skull and scalp [46]. Among the various spatial filtering methods, Common Spatial Pattern (CSP) is particularly popular [23]. CSP seeks to obtain optimal spatial features by maximizing the variance of one class while minimizing that of the other. This method utilizes average spatial covariance matrices derived from EEGs, which contain significant discriminative information for classification, including variance in the on-diagonal entries and coherence between adjacent channels in the off-diagonal entries [47].

CSP has proven highly effective in online BCI systems. Beyond its signal enhancement capabilities, one of the major strengths of CSP lies in the interpretability of its results. Unlike black-box methods, the outcomes of the optimization process of CSP can be visualized as scalp topographies, illustrating the filters and patterns of the optimized solution. This visual representation helps users understand how the algorithm processes and differentiates neural signals.

Moreover, CSP is widely used in many successful methods in the BCI Competition. Competitors have leveraged CSP-based spatial filtering to achieve impressive results, underscoring its robustness and effectiveness in practical applications. Its widespread adoption in competitive settings highlights the value of CSP in enhancing BCI system performance and its ongoing relevance in the field [48, 49].

Formally, let S^+ and $S^- \in \mathbb{R}^{n_C \times n_C}$ be the class-related estimates of spatial covariance matrices of segments in a two-class MI-EEG paradigm using the arithmetic

mean as follows,

$$S^+ := \frac{1}{|\mathcal{I}_+|}\sum_{l\in\mathcal{I}_+} X_l X_l{}^\top, \text{ and } S^- := \frac{1}{|\mathcal{I}_-|}\sum_{l\in\mathcal{I}_-} X_l X_l{}^\top,$$

where $\mathcal{I}_+$ and $\mathcal{I}_-$ are sets of segments in two classes respectively.

The CSP method is given by a simultaneous diagonalization of S^+ and S^- as follows,

$$WS^{+/-}W^\top = \Lambda^{+/-},$$

where diagonal matrices Λ^+ and $\Lambda^- \in \mathbb{R}^{n_C\times n_C}$ hold an identity constraint $\Lambda^+ + \Lambda^- = I_{n_C}$. Each column vector $w \in \text{Column}(W)$ is called a spatial filter. It is equivalent to solving a generalized eigenvalue problem as follows:

$$(S^+ + S^-)^{-1}S^{+/-}w = \lambda^{+/-}w. \tag{2.1}$$

A set of discriminative powers consists of m spatial filters $\{\mathbf{w_j}\}_{j=1}^m$ from both ends of the spectrum, i.e.,

$$z_j = \log\left(\mathbf{w_j}XX^\top\mathbf{w_j}{}^\top\right),$$

where $1 \le m \le n_C$. Once the discriminative features are extracted from EEG signals, they are commonly classified using a support vector machine (SVM). SVM is a widely used machine learning classifier renowned for its effectiveness in classification and regression tasks. The primary goal of SVM is to find an optimal hyperplane that maximally separates data points of different classes [44].

Hence, CSP yields the EEG signal analysis with the following paradigm: Let $X \in \mathbb{R}^{n_C\times n_T}$ be a segment of EEGs, where n_C is the number of channels and n_T is the number of sampled points on an epoch duration, and segment X is assumed to be band-pass filtered, centered, and scaled. The linear classifier that predicts the label for segment X can be written as follows,

$$f\left(X;\{\mathbf{w_i},\beta_i\}_{i=1}^N\right) = \sum_{i=1}^N \beta_i \log\left(\mathbf{w_i}XX^\top\mathbf{w_i}{}^\top\right) + \beta_0,$$

where N is the number of spatial filters, $\{\mathbf{w_i}\}_{i=1}^N \in \mathbb{R}^{n_C}$ are spatial filters and $\{\beta_i\}_{i=0}^N \in \mathbb{R}$ are biases.

Various variants of CSP have been developed to enhance its applicability and effectiveness in different applications, addressing specific challenges and improving overall performance across diverse scenarios. Below are brief descriptions of some well-known variants:

- Common Spatio-Spectral Pattern [50]: This variant integrates both spatial and spectral information from EEG signals by incorporating delay embedding, which enhances the ability to capture relevant temporal dynamics.
- Common Sparse Spectral Spatial Pattern [51]: This variant focuses on spectral information within specific frequency bands and imposes sparsity constraints, allowing the identification of the most relevant features with reduced complexity.
- Filter Bank Common Spatial Pattern [52]: By applying bandpass filters, this variant creates a filter bank that covers multiple frequency bands, enabling a more comprehensive analysis of EEG signal frequency components.
- Regularized Common Spatial Pattern [53]: This variant incorporates regularization techniques to mitigate overfitting, thereby improving the generalizability of the CSP method when applied to EEG data.
- Stationary Common Spatial Pattern [54]: This variant integrates stationary assumptions to ensure that the extracted features remain stable over time, which is crucial for long-term EEG monitoring and analysis.
- Wavelet Common Spatial Pattern [55]: This variant decomposes EEG signals into wavelet packets and uses fuzzy logic to select the most discriminative packets, enhancing the ability to capture non-stationary and transient EEG features.

Additionally, CSP can be expressed as a divergence maximization problem, where it directly maximizes divergence to obtain the CSP subspace, rather than solving a generalized eigenvalue problem. By incorporating regularization terms and beta divergence techniques between distributions, this approach, known as divCSP, systematically unifies various CSP variants. This method simplifies the process of obtaining the CSP subspace while increasing the flexibility and robustness of the filters through the use of regularization strategies [56].

Since 2017, numerous EEG-based motor imagery classifiers utilizing convolutional neural networks (CNNs) have been developed [16–18, 57, 58]. These classifiers can end-to-end extract essential information from the temporal, spatial, and frequency domains without the need for hand-crafted features, which have garnered significant attention due to their strong performance. A key distinction between these CNNs and traditional methods, such as those using EEG spatial covariance matrices in CSP approaches, lies in the input format. CNN models use multi-channel EEG time series as input, rather than covariance matrices. These CNN-based classifiers have consistently outperformed CSP methods across various motor imagery tasks and have become increasingly popular in addressing challenges in multi-modal EEG tasks [18, 58].

2.2 Evaluation Scenarios and Datasets

This section introduces the experimental scenarios and datasets employed for evaluation in EEG-based motor imagery classification (Chapters 5, 6, 7, 8, and 9) and the multimodal fusion of simultaneous EEG-fMRI data (Chapter 9).

2.2.1 MI-EEG Datasets

The scenarios presented are among the most commonly utilized experimental setups in EEG-based motor imagery classification, specifically designed to address non-stationarity issues, including within-session variability, and between-session/subject variability. The three primary scenarios are as follows:

- 10-Fold Cross Validation: This scenario is designed to evaluate within-session variability. The data for each subject is divided into ten equally sized, class-balanced folds. Nine folds are used for training, and one fold is reserved for testing. This cross-validation process is repeated ten times to ensure robustness.
- Holdout: This scenario focuses on assessing between-session variability. The data from the first session of each subject is used for training, while the data from the second session is reserved for testing. Typically, the EEG data

from these two sessions are collected on different days, leading to significant variability between sessions.

- Leave-One-Out Cross Validation: This scenario is aimed at evaluating between-subject variability. The dataset is divided into folds, each corresponding to a different subject. In each iteration, the data of one subject is used as the test set, while the data from the remaining subjects is used for training. This process is repeated for every subject in the dataset.

The publicly available MI-EEG datasets used for evaluation can be accessed through the MOABB [2] package or the BNCI Horizon 2020 [3] project.

All EEG signals undergo preprocessing using Chebyshev Type II filters with 4 Hz increments, designed for a maximum passband loss of 3 dB and a minimum attenuation of 30 dB in the stopband. Table 2.1 provides a summary of the selected MI-EEG datasets.

- Korea University Dataset: Also known as Lee2019MI in the MOABB package, this dataset contains EEG signals from 54 subjects who participated in a binary-class motor imagery task. EEG signals were recorded at a sampling rate of 1,000 Hz using 62 electrodes. Each trial's EEG signals were segmented from the first second to 3.5 seconds after stimulus onset, resulting in a total duration of 2.5 seconds per trial. The KU dataset is the largest publicly available motor imagery dataset for two movements. It is divided into two sessions, labeled S_1 and S_2, each containing training and testing phases with 100 trials equally split between right and left-hand imagery tasks. In total, 21,600 trials are available for evaluation, calculated as the product of 2 sessions, 54 subjects, and 200 trials per subject.

- Cho2017 Dataset: This dataset from the MOABB package involves 52 subjects performing motor imagery tasks for the left and right hands. EEG data were recorded using 64 Ag/AgCl active electrodes according to the international 10-10 system at a sampling rate of 512 Hz. The recording began at second 0, following the cue, and continued for 3 seconds until the end of the cross period. For analysis, 20 electrodes were selected from the

[2] `https://github.com/NeuroTechX/moabb`.
[3] `http://bnci-horizon-2020.eu/database/data-sets`.

motor cortex region, including FC-5/3/1/2/4/6, C-5/3/1/z/2/4/6, and CP-5/3/1/z/2/4/6. Note that the EEG data for subjects No. 32, No. 46, and No. 49 were excluded, resulting in a final dataset of 49 subjects.

- BNCI2014001 Dataset: Part of the BNCI Horizon 2020 project, also known as BCIC-IV-2a, this dataset includes 9 participants who performed an MI task with four classes: left hand, right hand, feet, and tongue. EEG data were recorded using 22 Ag/AgCl electrodes and three EOG channels at a sampling rate of 250 Hz, filtered between 0.5 and 100 Hz with a 50 Hz notch filter to eliminate line noise. The study was conducted over two separate sessions on different days, with each participant completing 288 trials (six runs of 12 cue-based trials per class). The EEG signals were segmented from the cue onset at 2.0 seconds to the end of the motor imagery period at 6.0 seconds, giving a total epoch duration of 4 seconds.

- BNCI2014002 Dataset: Also from the BNCI Horizon 2020 project, this dataset includes 13 participants who performed sustained 5-second kinaesthetic MI tasks involving their right hand and feet, as indicated by the cue. EEG data were recorded using a biosignal amplifier and active Ag/AgCl electrodes at a sampling rate of 512 Hz, with a total of 15 electrodes placed on the participants. The experimental session consisted of eight runs, with 80 trials for each class, resulting in 160 trials per participant. The EEG signals were segmented from the cue onset at 3.0 seconds to the end of the motor imagery period at 8.0 seconds, yielding a total epoch duration of 5 seconds.

- BNCI2015001 Dataset: This dataset from the BNCI Horizon 2020 project involves 12 participants performing imagery of right-hand movement versus imagery of movement in both feet. Data were recorded at a sampling rate of 512 Hz and filtered using a bandpass filter from 0.5 to 100 Hz, along with a notch filter at 50 Hz. Recording started 3.0 seconds after the prompt and continued until the end of the cross period at 8.0 seconds, resulting in a total trial duration of 5 seconds. Most studies (Subjects $1-8$) were conducted in two sessions, labeled S_A and S_B, on consecutive days. However, for Subjects 9, 10, 11, and 12, three sessions were labeled S_A, S_B, and S_C. In each session, every participant completed 100 trials per class, totaling 200 trials per participant per session.

Table 2.1: Summary of Dataset Settings for Five Selected Publicly Available MI-EEG Datasets. In the experiments, we exclusively use EEG electrodes from the primary motor cortex of the subjects. The original dataset settings are provided in parentheses for reference.

Dataset	Subject	Channel	Class	Trials/Session	Length	Imagery Period	Sampling Rate	Session
KU	54	20 (62)	left hand, right hand	200	2.5 s	1 to 3.5 s	1000 Hz	2
Cho2017	49 (52)	20 (64)	left hand, right hand	200	3 s	3 to 6 s	512 Hz	1
BNCI2014001	9	22	left hand, right hand,feet,tongue	288	4 s	2 to 6 s	250 Hz	2
BNCI2014002	14	15	right hand,feet	160	5 s	3 to 8 s	512 Hz	1
BNCI2015001	12	13	right hand,feet	200	5 s	3 to 8 s	512 Hz	2 (2 or 3)

2.2.2 Simultenous EEG-fMRI Dataset

In Chapter 9, we utilized a publicly available dataset [4], which includes simultaneous EEG-fMRI recordings from eight subjects under two resting conditions (eyes closed and eyes open). This dataset was specifically created to evaluate EEG artifact correction methods tailored for data collected within MR scanners [59], and it encompasses various experimental settings. For our analysis, we concentrated exclusively on the simultaneous EEG-fMRI condition, where EEG and fMRI data were recorded synchronously. In this setup, each subject underwent two runs, with each run consisting of continuous recordings lasting 4.5 minutes. During each run, subjects alternated between conditions based on visual cues every 30 seconds.

For comprehensive details regarding the experimental protocol, recording devices, and settings, please refer to the original dataset paper [60]. Briefly, EEG data were recorded using 30 MR-compatible electrodes placed at standardized locations on the head, with electrical activity sampled at 5 kHz using MR-compatible EEG amplifiers. The clocks of these amplifiers were synchronized with the MR scanner to facilitate artifact correction [61]. The dataset was collected using two MR scanners: four subjects were scanned with a Siemens TIM Trio, while the remaining subjects were scanned with a Siemens Verio. Both scanners used standard echo-planar imaging sequences to record fMRI activity, with identical acquisition parameters except for a slight variation in repetition time—1.95 s for the TIM Trio and 2.0 s for the Verio. Consequently, the fMRI recordings provide volume activity (i.e., discrete grids with 3 mm isotropic resolution covering the subject's head) sampled at intervals of 2.0 s (or 1.95 s for the TIM Trio).

The data from each run were preprocessed independently. For EEG data, we utilized a custom preprocessing pipeline. The dataset included both raw and MR gradient-artifact-corrected EEG data [62, 63], with the latter being used in our analysis. Each EEG channel's activity was first bandpass filtered between 0.5 and 125 Hz, followed by the attenuation of pulse artifacts [59]. Subsequently, the data were resampled to 250 Hz (with a 70 Hz cut-off frequency) and subjected to standard EEG preprocessing procedures.

In summary, we identified and interpolated faulty EEG channels [64], removed transient single-channel artifacts [65], re-referenced the channels to their common

[4] `https://www.nitrc.org/projects/cwleegfmri_data/`

average, attenuated 50 Hz line noise [66], and eliminated transient muscle artifacts [67]. The residual EEG activity was then decomposed into independent components (ICs) [68], which were classified as either artifacts or brain activity by an automated model [69]. We used lenient thresholds to retain an average of 85% (±9%) of the ICs. After removing artifact ICs, the remaining components were projected back into the original EEG channel space. We excluded six temporal and pre-frontal channels, known to primarily capture artifacts, resulting in 24 remaining EEG channels. As a final preprocessing step, the data were bandpass filtered between 1 and 36 Hz and resampled to 125 Hz to emphasize broad-band brain activity.

For fMRI data, preprocessing followed a method similar to that described by [1]. We used fmriprep [70] for standard anatomical and functional preprocessing of the raw scans. This included spatial normalization of the fMRI data to standard space MNI152NLin2009cAsym, slice-time correction, and estimation of various confounds. We selected global signal, motion parameters (six basic parameters), and the first five principal components from white matter and cerebrospinal fluid as confound regressors. Using the *nilearn* [5] package, we regressed out these confounds from the fMRI data, applied spatial smoothing with a Gaussian kernel (full width at half maximum of 3 mm), extracted activity from 128 regions of interest (using the DiFuMo atlas [71]), applied a bandpass filter (0.01 Hz to 0.2 Hz), and resampled the data to 1 Hz (i.e., 1 volume per second).

EEG and fMRI data were segmented into eight epochs (each 20 seconds long) based on triggers marking the onset of eye open and closed conditions. Following the guidelines of [72], we applied robust z-scoring to standardize the EEG channel and fMRI ROI data for each run. The resulting activity was clipped within the interval $[-20, 20]$ to mitigate the impact of outliers. The paired data from these epochs were allocated to training, validation, and test sets. We employed 10-fold stratified cross-validation, leave-one-run-out, and leave-one-subject-out cross-validation strategies, as illustrated in Figure 2.2. To increase the number of observations in each set, we extracted sliding windows (10 seconds long with a 9-second overlap) from each epoch.

[5] `https://nilearn.github.io/`

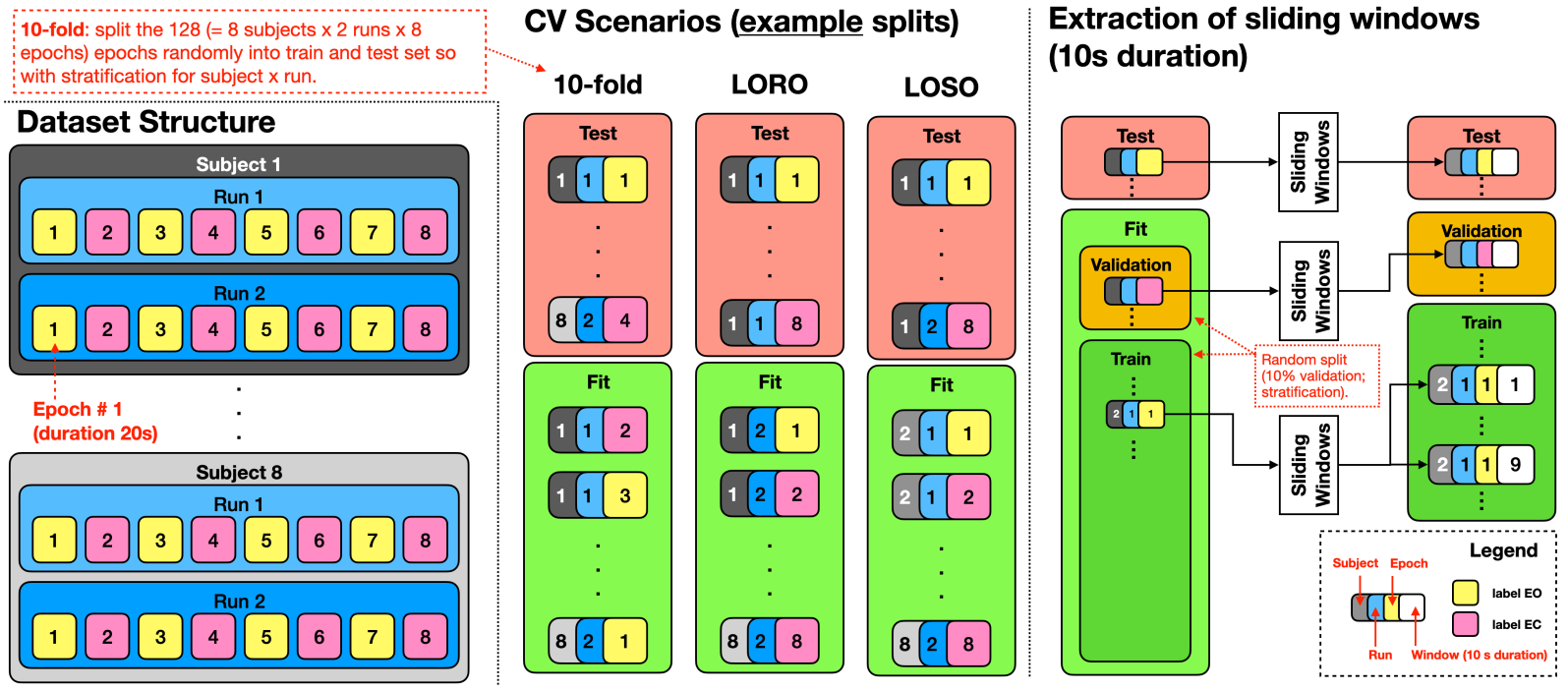


Figure 2.2: Simultaneous EEG-fMRI Dataset: Visualization of Cross-Validation Scenarios and Sliding Window Extraction. Left: The dataset includes two runs per subject, with each run consisting of eight trials lasting 30 seconds each (with no breaks in between). During these trials, subjects were either resting with their eyes open (EO) or eyes closed (EC). From each trial, we extracted one epoch of 20 seconds duration, with a 2-second offset. Middle: The extracted epochs were divided into cross-validation (CV) folds according to different scenarios. We utilized three CV strategies: 10-fold stratified, leave-one-run-out (LORO), and leave-one-subject-out (LOSO). Right: After creating the CV splits, sliding windows were extracted from each epoch.

Chapter 3

Covariance-Based Neuroimaging Data

In this chapter, we will begin by introducing the concept of covariance-based neuroimaging data, followed by an exploration of how this data is modeled within the framework of Riemannian geometry, specifically on the SPD manifold. We will then discuss two commonly used Riemannian metrics on SPD manifolds: the affine-invariant Riemannian metric and the Log-Euclidean metric. Finally, we will delve into the Riemannian-based approach, a prominent category of EEG-based motor imagery classifiers that operates on the SPD manifold.

Definition 3.1 (Covariance-Based Neuroimaging Data)**.** The covariance matrix of a neuroimaging data segment, represented by $X \in \mathbb{R}^{n_C \times n_T}$, is denoted as $S = XX^\top \in \mathcal{S}_{++}^{n_C}$, where n_C is the number of spatial dimensions and n_T is the signal timesteps.

This definition applies broadly to any multi-channel neuroimaging time series. By nature, covariance-based neuroimaging data is inherently symmetric and positive-definite. Even in cases of degeneracy, such as co-linear spatial dimensions, positive definiteness can be preserved through shrinkage regularization or dimensionality reduction. In this study, we focus specifically on two neuroimaging modalities:

- EEG spatial covariance matrix: it describes the correlations between signals from different EEG electrodes, used to analyze spatial patterns of brain activity.

- fMRI functional connectivity: it reflects static and dynamic communication between brain regions by measuring the correlations in their activity acquired by fMRI. These matrices can reveal changes in brain activity during tasks and pathological conditions [73, 74].

EEG spatial covariance matrices, in particular, contain nearly all the discriminative information necessary for motor imagery classification tasks. This suggests an equivalence between time-domain statistics, represented by spatial covariance matrices, and distinct frequency-domain statistics, such as band power and magnitude-squared coherence. The following three categories, as outlined in [75], include diverse statistical metrics derived from both time and frequency domains:

- Spatial Features: Primarily involving time-domain statistics, these features can be extracted through supervised methods like principal component analysis, independent component analysis, and common spatial patterns, or unsupervised methods like common-average reference and surface Laplacian spatial filters.
- Frequency Features: Focusing on frequency-domain statistics, this category includes measures such as band power, fast Fourier transform, and wavelet analysis.
- Similarity Features: Capturing the similarity between EEG signals, this category includes frequency-domain statistics like phase-locking value and magnitude-squared coherence.

Classifiers that use EEG spatial covariance matrices in MI-EEG classification can extract information similar to that derived from other metrics, such as phase-locking value and magnitude-squared coherence. For example, using the auto-covariance function, the off-diagonal elements of EEG spatial covariance matrices can be related to the magnitude-squared coherence without requiring normalization. Consider a wide-sense stationary real-valued random process X_t with zero mean $\mathbb{E}(X_t) = 0$. The auto-correlation function $R_X(\tau)$ of X_t is defined as $R_X(\tau) := \mathbb{E}(X_t^\top X_{t+\tau})$, and the expected (band) power P_X of X_t is given by,

$$P_X := \mathbb{E}|X_t|^2 = R_X(0) = \int_{\mathbb{R}} S_X(\omega)\, d\omega,$$

where $S_X(\omega)$ is the power spectral density. Thus, the variance of zero-centered X_t is $\sigma_X^2 := \mathbb{E}|X_t - \mathbb{E}(X_t)|^2 = P_X$. This argument applies to X_c of each channel, meaning the diagonal entries of the EEG spatial covariance matrices correspond to the variance $\sigma_{X_c}^2$, which equals the expected (band) power P_{X_c} of that channel c. This relationship has been explored in previous studies, notably in [42].

3.1 Riemannian Geometry

This section provides a concise introduction to smooth and Riemannian manifolds. For a more comprehensive understanding of Riemannian geometry, readers are encouraged to consult standard textbooks such as those by Petersen [76] and Jost [77].

Smooth Manifolds

A smooth manifold is a topological space that locally resembles Euclidean space and admits a smooth atlas, which is a collection of coordinate charts that are compatible with each other.

Definition 3.2. A topological space $\mathcal{M}$ is called a *smooth manifold* of dimension N if it satisfies the following compatibility conditions:

- $\mathcal{M}$ is a Hausdorff space: for any two distinct points $x, y \in \mathcal{M}$, there exist disjoint open neighborhoods $\mathcal{U}_x$ and $\mathcal{U}_y$ containing x and y, respectively.
- $\mathcal{M}$ has a countable basis for its topology: there is a countable collection of open sets $\mathcal{U}_i$ such that every open set in the topology can be expressed as a union of some of the $\mathcal{U}_i$.
- $\mathcal{M}$ is equipped with a smooth atlas $(\mathcal{U}_\alpha, \phi_\alpha)$, where each $\phi_\alpha : \mathcal{U}_\alpha \longmapsto \mathcal{V}_\alpha \subseteq \mathbb{R}^N$ is a homeomorphism, and the charts are smoothly compatible: For any two overlapping charts $(\mathcal{U}_\alpha, \phi_\alpha)$ and $(\mathcal{U}_\beta, \phi_\beta)$, the transition map $\phi_{\beta\alpha} = \phi_\beta \circ \phi_\alpha^{-1} : \phi_\alpha(\mathcal{U}_\alpha \cap \mathcal{U}_\beta) \longmapsto \phi_\beta(\mathcal{U}_\alpha \cap \mathcal{U}_\beta)$ is a smooth function (or infinitely differentiable) from $\mathbb{R}^N$ to $\mathbb{R}^N$.

The tangent space at a point on a smooth manifold is a vector space consisting of tangent vectors to the manifold at that point. The formal definition of the tangent space at a point p on a smooth manifold $\mathcal{M}$ is as follows:

Definition 3.3. The *tangent space* to $\mathcal{M}$ at p, denoted $T_p\mathcal{M}$, is defined as the set of all tangent vectors at p, where a tangent vector X_p is a linear map $X_p : C^\infty(\mathcal{M}) \longmapsto \mathbb{R}$ satisfying the following properties: for any smooth functions $f, g \in C^\infty(\mathcal{M})$ and any scalar $c \in \mathbb{R}$:

- $\mathbb{R}$-Linearity: $X_p(cf + g) = cX_p(f) + X_p(g)$.
- Leibniz Rule: $X_p(fg) = f(p)X_p(g) + g(p)X_p(f)$.

A vector field on a smooth manifold $\mathcal{M}$ is a smooth map $X : \mathcal{M} \longmapsto T\mathcal{M}$ that assigns to each point $p \in \mathcal{M}$ a tangent vector $X_p \in T_p\mathcal{M}$, where $T\mathcal{M}$ denotes the tangent bundle of $\mathcal{M}$. The map should satisfy the following property: for every smooth function $f : \mathcal{M} \longmapsto \mathbb{R}$, the composition $f \circ X : \mathcal{M} \longmapsto \mathbb{R}$ is also a smooth function, formally written as:

$$X : \mathcal{M} \longmapsto T\mathcal{M}, \quad X(p) = X_p \in T_p\mathcal{M}.$$

Riemannian Manifolds

To enable the measurement of distances, angles, and lengths of curves on the smooth manifold, we introduce a metric known as the Riemannian metric. This metric provides a smoothly varying assignment of inner products to the tangent spaces of the manifold, thereby endowing the smooth manifold with the structure of a Riemannian manifold $(\mathcal{M}, g)$.

Definition 3.4. A *Riemannian metric* g at any $p \in \mathcal{M}$ is a smooth assignment of an inner product $g_p : T_p\mathcal{M} \times T_p\mathcal{M} \longmapsto \mathbb{R}$ to each tangent space $T_p\mathcal{M}$, such that for any $u, v \in T_p\mathcal{M}$, the following properties hold:

- Positive-definiteness: $g_p(u, u) \geq 0$ and $g_p(u, u) = 0$ if and only if $u = \mathbf{0}$.
- Symmetry: $g_p(u, v) = g_p(v, u)$.

Moreover, we assume that g_p exhibits smooth variation. Given any two smooth vector fields X and Y, the inner product $g_p(X|_p, Y|_p)$ is a smooth function of p.

To establish a geometrically meaningful connection between local and global structures, the concept of Riemannian connection is introduced as follows:

Definition 3.5. A Riemannian connection on a smooth manifold $\mathcal{M}$ equipped with a Riemannian metric g is a linear map $\nabla : (X, Y) \longmapsto \nabla_X Y$ that satisfies the following conditions:

- $\mathbb{R}$-Linearity: for any smooth vector fields X, Y_1, Y_2 on $\mathcal{M}$ and $\alpha, \beta \in \mathbb{R}$,

$$\nabla_X(\alpha Y_1 + \beta Y_2) = \alpha \nabla_X Y_1 + \beta \nabla_X Y_2.$$

- $C^\infty(\mathcal{M})$-Linearity: for any smooth vector fields X_1, X_2, Y on $\mathcal{M}$ and $f, g \in C^\infty(\mathcal{M})$,

$$\nabla_{fX_1+gX_2} Y = f \nabla_{X_1} Y + g \nabla_{X_2} Y.$$

- Leibniz Rule: for any smooth vector fields X, Y on $\mathcal{M}$ and $f \in C^\infty(\mathcal{M})$,

$$\nabla_X(fY) = (Xf)Y + f \nabla_X Y.$$

- Torsion-free: for any smooth vector fields X, Y on $\mathcal{M}$,

$$\nabla_X Y - \nabla_Y X - [X, Y] = 0,$$

  where $[X, Y]$ denotes the Lie bracket of X and Y.

- Metric-compatible: for any smooth vector fields X, Y, Z on $\mathcal{M}$,

$$X\big(g(Y, Z)\big) = g(\nabla_X Y, Z) + g(Y, \nabla_X Z).$$

$\nabla_X Y$ is called the covariant derivative of Y in the direction of X, and the Riemannian connection is also referred to as the Levi-Civita connection, which uniquely exists.

Let $\gamma : [0, 1] \to \mathcal{M}$ be a smooth curve, we say a vector field X with respect to the Levi-Civita connection is parallel along γ if $\nabla_{\dot{\gamma}(t)} X = 0$ for all t. For any

vector $v \in T_{\gamma(0)}\mathcal{M}$, there exists a unique parallel vector field X along γ with $X(0) = v$. The terminal value $X(1)$ is called the parallel transport of v along γ, which is always interpreted as a linear map $P_\gamma : T_{\gamma(0)}\mathcal{M} \to T_{\gamma(1)}\mathcal{M}$. The Levi-Civita connection ensures that the parallel transport is independent of the choice of curve. In addition, a geodesic is a curve $\gamma : [0, 1] \to \mathcal{M}$ such that its tangent vector $\dot{\gamma}$ is parallel transported along itself, i.e., $\nabla_{\dot{\gamma}}\dot{\gamma} = 0$.

Note that $\nabla_{\gamma'(t)}\gamma'(t) = 0$ enforces $\gamma(t)$ to be curved with zero acceleration. The concept *geodesic* extends the idea of straight lines from Euclidean spaces. It is worth mentioning that every geodesic is parameterized in proportion to its arc length.

In a local chart $(\mathcal{U}, \varphi)$, the geodesic curve $\gamma(t)$ satisfies the following second-order ordinary differential equations:

$$\ddot{u_k}(t) + \sum_{ij} \Gamma^k_{ij} \dot{u_i}(t)\dot{u_j}(t) = 0, \text{ for } k = 1, ..., n,$$

where each $u_k(t) = \varphi^k \circ \gamma(t)$. For local frames $\{\partial_i\}_{i=1}^n$, Γ^k_{ij} denotes the Christoffel symbols of ∇ given by $\nabla_{\partial_i}(\partial_j) = \Gamma^k_{ij}\,\partial_k$, $\forall\, i, j, k = 1, ..., n$.

By the Picard-Lindelöf theorem, this yields the local existence and uniqueness of the solution for a geodesic: For any given $p \in \mathcal{M}$ and $v \in \mathcal{T}_p\mathcal{M}$, there exists an interval $(-\epsilon, \epsilon)$ and a unique geodesic $\gamma : (-\epsilon, \epsilon) \longmapsto \mathcal{M}$ satisfying the initial conditions $\gamma(0) := p$ and $\gamma'(0) := v$.

The Riemannian distance between two points $p, q \in \mathcal{M}$ is defined as the infimum of the lengths of all piecewise smooth curves joining p and q as follows:

$$d(p, q) := \inf_\gamma \int_0^1 \|\dot{\gamma}(t)\|_g \, dt,$$

where the infimum is taken over all piecewise smooth curves $\gamma : [0, 1] \longmapsto \mathcal{M}$ with $\gamma(0) = p$ and $\gamma(1) = q$.

To enhance the comprehension of Riemannian geodesics and their collective behavior, the exponential map is introduced as follows: Let $v \in T_p\mathcal{M}$ be a tangent vector to each point $p \in \mathcal{M}$. Then, there exists a unique geodesic γ_v satisfying $\gamma_v(0) = p$ and $\dot{\gamma}_v(0) = v$.

The exponential map is defined by $\exp_p(v) = \gamma_v(1)$. In particular, $\exp_p(\mathbf{0}_p) = p$, where $\mathbf{0}_p$ denotes the zero vector in $T_p\mathcal{M}$. There exists a neighborhood $\mathcal{U}$ of $\mathbf{0}_p$ in $T_p\mathcal{M}$ such that $\exp_p : \mathcal{U} \longmapsto \mathcal{V}$ is a diffeomorphism onto its image $\mathcal{V} \subset \mathcal{M}$. The image of $\exp_p(T_p\mathcal{M})$ is called the exponential neighborhood of p. The radius of the largest ball centered at $\mathbf{0}_p$ that is contained in $\mathcal{U}$ is called the injectivity radius of $\mathcal{M}$ at p. The inverse of the exponential map is called the logarithmic map. For any $q \in \mathcal{M}$ sufficiently close to p, the logarithmic map $\mathrm{Log}_p(q)$ is defined as the unique tangent vector $v \in T_p\mathcal{M}$ such that $\exp_p(v) = q$. The exponential and logarithmic maps are not defined globally on $\mathcal{M}$, but only on appropriate subsets.

3.2 Symmetric Positive Definite Manifolds

In this section, we focus on a particular class of Riemannian manifolds known as SPD manifolds, which are widely utilized in fields such as robotics and signal processing. Two research groups—Tom Fletcher and Sarang Joshi's group [78, 79], and Xavier Pennec, Pierre Fillard, and Nicholas Ayache' s group [80]—independently introduced the affine-invariant Riemannian metric on the space of SPD tensors for statistical analysis in diffusion tensor imaging. Initially, SPD matrices were modeled within a vector space using a standard additive matrix structure. However, it soon became evident that many operations on SPD matrices are non-convex, often leading to a loss of positive definiteness, which poses significant challenges for computational methods in these spaces [80]. Employing Riemannian geometry to model matrix spaces ensures that operations on the SPD manifold maintain both matrix symmetry and positive definiteness, and has become a natural choice in various engineering and scientific disciplines [81].

One of the well-known Riemannian metrics of SPD manifolds is the affine-invariant Riemannian metric g^{AIRM} [80]. Formally, the affine-invariant Riemannian metric is defined at any $P \in \mathcal{S}_{++}$, as follows:

$$g_P^{AIRM}(v, w) := \langle P^{-\frac{1}{2}} v P^{-\frac{1}{2}}, P^{-\frac{1}{2}} w P^{-\frac{1}{2}} \rangle_{\mathcal{F}} = \mathrm{Tr}\big(P^{-1} v P^{-1} w\big),$$

where $v, w \in T_P\mathcal{S}_{++}$. This metric yields the Riemannian distance between two SPD matrices P_1 and P_2 given by:

$$d_{g^{AIRM}}(P_1, P_2) = \left\| \log\left(P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}}\right) \right\|_{\mathcal{F}}.$$

Computations involving the affine-invariant Riemannian metric often perform slowly due to the big computational burden of matrix operations. Hence, another commonly used Riemannian metric, the Log-Euclidean metric g^{LEM}, is proposed to reduce this computational complexity [82, 83]. Formally, a commutative Lie group structure, known as the tensor Lie group, is endowed on $\mathcal{S}_{++}$ using the logarithmic multiplication $\odot$ defined by $P_1 \odot P_2 := \exp\big(\log P_1 + \log P_2\big)$, where $P_1, P_2 \in \mathcal{S}_{++}$, and exp and log are matrix exponential and logarithm respectively. The bi-invariant metric exists on the tensor Lie group, which yields the following scalar product:

$$g_P^{LEM}(v, w) := \langle D_P \log v, D_P \log w \rangle_{\mathcal{F}},$$

where tangent vectors v, w at P, and $D_P \log$ represents the differential $D \log$ at P defined in the context of Lie group and algebra. It yields a simple Riemannian distance formula $d_{g^{LEM}}(P_1, P_2) = \|\log P_1 - \log P_2\|_{\mathcal{F}}$ between two SPD matrices P_1 and P_2. The Log-Euclidean metric introduces a vector space structure to the space of SPD matrices while preserving several key properties, including invariance under inversion, logarithmic multiplication, orthogonal transformations, and scaling.

The SPD manifold equipped with either of the two metrics is a Cartan-Hadamard manifold, characterized by non-positive sectional curvature and global diffeomorphism to Euclidean space [84]. On a Cartan-Hadamard manifold, any two points are connected by a unique geodesic. In addition, for small deviations from the identity matrix, denoted as S_1 and S_2, the Log-Euclidean distance $d_{g^{LEM}}(S_1, S_2)$ can be considered as an approximation of $d_{g^{AIRM}}(S_1, S_2)$. This relationship is established using the Campbell-Baker-Hausdorff formula, which can be expressed as $d^2_{g^{AIRM}}(S_1, S_2) - d^2_{g^{LEM}}(S_1, S_2) = \mathrm{Tr}(\,[\log S_1, [\log S_1, \log S_2]]\,)(\log S_1 - \log S_2)/12 + \ldots$, where Lie bracket $[A, B] := AB - BA$.

In this thesis, the Log-Euclidean metric is employed in Chapter 7, while the remaining chapters utilize the affine invariant Riemannian metric. Table 3.1 provides the formulas for commonly used operators on SPD manifolds with these two Riemannian metrics.

Table 3.1: Operator on SPD manifolds equipped with g^{AIRM} and g^{LEM}: $S_1, S_2, P \in \mathcal{S}_{++}$ and $v, w \in \mathcal{T}_P\mathcal{S}_{++}$; exp and log are the matrix exponential and the logarithm; Tr is the trace of a matrix; $D\log$ and $D\exp$ are the derivatives of log and exp, respectively.

Operation	g^{AIRM}	g^{LEM}
$\langle v, w \rangle_P$ (Inner Product)	$\mathrm{Tr}(P^{-1}vP^{-1}w)$	$\langle D_P \log v, D_P \log w \rangle_{\mathcal{F}}$
$\exp_P S$ (Exponential)	$P^{\frac{1}{2}} \exp\left(P^{-\frac{1}{2}} S P^{-\frac{1}{2}}\right) P^{\frac{1}{2}}$	$\exp\left(\log P + D_P \log S\right)$
$\log_P S$ (Logarithm)	$P^{\frac{1}{2}} \log\left(P^{-\frac{1}{2}} S P^{-\frac{1}{2}}\right) P^{\frac{1}{2}}$	$D_{\log P} \exp(\log S - \log P)$
$d_g(S_1, S_2)$ (Distance)	$\left\|\log\left(S_1^{-\frac{1}{2}} S_2 S_1^{-\frac{1}{2}}\right)\right\|_{\mathcal{F}}$	$\|\log S_1 - \log S_2\|_{\mathcal{F}}$
$\gamma_t(S_1, S_2)$ (Geodesic)	$S_1^{\frac{1}{2}} \exp\left(t \log\left(S_1^{-\frac{1}{2}} S_2 S_1^{-\frac{1}{2}}\right)\right) S_1^{\frac{1}{2}}$	$\exp\left((1-t)\log S_1 + t \log S_2\right)$

3.3 Riemannian-Based Approaches

Inspired by research on SPD manifolds in medical image analysis, a class of motor imagery classifiers known as the Riemannian-based approach leverages SPD manifolds equipped with the affine-invariant Riemannian metric to represent the space of EEG spatial covariance matrices. This approach has garnered increasing attention within the BCI community, prompting ongoing research aimed at refining its framework and expanding its applications across various BCI tasks [47, 85].

The success of Riemannian-based methods is largely due to their ability to distinguish between task classes by leveraging Riemannian distances on SPD manifolds. Specifically, the Riemannian distance between class-related EEG spatial covariance matrices S^+ and S^- is expressed in terms of spectral powers as follows:

$$d_{g^{AIRM}}(S^+, S^-) = \sqrt{\sum_{i=1}^{n_C} \log^2\left(\frac{\lambda_i}{1-\lambda_i}\right)},$$

where each $\lambda_i \in (0, 1)$ is an eigenvalue obtained from Equation 2.1. A greater Riemannian distance between two EEG spatial covariance matrices implies a higher spectral power, which directly aligns with the classification criterion in motor imagery tasks that rely on ERD/ERS effects, using spectral power as a discriminative

feature [86].

Another key advantage of this approach lies in the affine-invariant Riemannian distance, which remains unchanged after any transformation:

$$d_{g^{AIRM}}(S_1, S_2) = d_{g^{AIRM}}(WS_1W^\top, WS_2W^\top),$$

where S_1 and S_2 are two EEG spatial covariance matrices, and W belongs to the general linear group. This property helps maintain the discriminative power between task classes.

It's important to note that the geometric classifiers proposed in this study differ fundamentally from Riemannian-based methods. While Riemannian-based approaches rely on the length of Riemannian distances for classification, the geometric classifiers seek nonlinear projections on Stiefel manifolds by maximizing class discrimination using a neural network-based approach.

Several Riemannian-based approaches addressing different applications and scenarios are introduced below:

- Minimum Distance to Riemannian Mean (MDRM, or referred to as MDM) [24]: This method uses geodesic distance on $(\mathcal{S}_{++}, g^{AIRM})$ as the key feature in classification.
- Tangent Space Mapping (TSM) [24]: This method projects data onto the tangent space at their corresponding Fréchet means on $(\mathcal{S}_{++}, g^{AIRM})$, producing a tangent vector that serves as the key feature in classification.
- Riemannian Procrustes Analysis (RPA) [87]: This method applies geometric transformations on $(\mathcal{S}_{++}, g^{AIRM})$ such as translation, scaling, and rotation to address variability between sessions/subjects. The base classifier is the MDM method. RPA consists of three steps: recentering (RCT), stretching, and rotation (ROT). RCT aligns the centroids of data to the identity matrix, stretching normalizes variances, and ROT aligns the orientation of data point clouds. RCT is supervised and suitable for semi-supervised domain adaptation, while ROT is unsupervised and can be used in unsupervised domain adaptation scenarios. In particular, RCT has been proposed as a standalone method for EEG-based motor imagery classification [88].

Chapter 4

Geometric Statistics and Geometric Methods

4.1 Geometric Statistics

Geometric statistics seeks to extend statistical concepts and tools to Riemannian manifolds, enabling the analysis of manifold-valued data. Examples include the Fréchet mean, principal geodesic analysis, and geodesic regression [78, 79, 89]. We will introduce these concepts below, as they are frequently employed in this study.

4.1.1 Fréchet Mean

The Fréchet mean was first proposed to generalize the concept of the statistical mean to a metric space [90]. It is also known as the Karcher mean, and the two terms are often used interchangeably in the literature [91]. The Fréchet mean exists and is unique when the Riemannian manifold $(\mathcal{M}, g)$ is complete and satisfies certain curvature conditions [92].

Formally, given a set of points $\{p_i\}_{i=1}^N \in (\mathcal{M}, g)$, the Fréchet mean μ is defined as the minimizer of the sum-of-squared Riemannian distances as follows,

$$\mu := \arg\min_{p \in \mathcal{M}} \frac{1}{N} \sum_{i=1}^{N} d_g^2(p, p_i).$$

It yields the Fréchet variance σ that is defined as the expectation of the sum-of-squared Riemannian distances with respect to the Fréchet mean as follows,

$$\sigma^2 := \mathcal{E}\left[d_g^2(\mu, p)\right] = \frac{1}{N}\sum_{i=1}^{N} d_g^2(\mu, p_i).$$

For manifold-valued random samples $\{p_i\}_{i=1}^N \in \mathcal{M}$, the sample covariance matrix is given by

$$S := \sum_{i=1}^{N} \log_\mu(p_i) \log_\mu(p_i)^\top.$$

where μ are Fréchet mean of samples and log is Riemannian logarithm map.

On SPD manifolds equipped with the affine-invariant Riemannian metric g^{AIRM}, the Fréchet mean is implicitly computed through a barycentric equation and solved iteratively as follows:

$$\mu_{t+1} \longleftarrow \mu_t^{\frac{1}{2}} \exp\Big(\frac{1}{N}\sum_{i=1}^{N} \log\left(\mu_t^{-\frac{1}{2}} S_i \mu_t^{-\frac{1}{2}}\right)\Big)\mu_t^{\frac{1}{2}}. \tag{4.1}$$

On SPD manifolds equipped with the Log-Euclidean metric g^{LEM}, the Fréchet mean is simplified to an explicit formula as follows:

$$\mu \longleftarrow \exp\left(\frac{1}{N}\sum_{i=1}^{N} \log\left(S_i\right)\right). \tag{4.2}$$

4.1.2 Principal Geodesic Analysis

The aim of Principal Geodesic Analysis (PGA) is to identify a series of nested geodesic submanifolds that maximize the projected variance of the data [78]. These submanifolds are referred to as the principal geodesic submanifolds.

Formally, suppose tangent vectors $\{v_i\}_{i=1}^N$ span the tangent space $\mathcal{T}_\mu\mathcal{M}$ at Fréchet mean μ. One can construct a sequence of nested subspaces V_k by taking the span of the first k vectors in the set $\{v_i\}_{i=1}^N$. The principal geodesic submanifolds are then obtained by applying the exponential map $H_k = \exp_\mu(V_k)$ to V_k.

The first principal direction is selected to maximize the projected variance along the corresponding geodesic, as follows:

$$v_1 := \arg\max_{\|v\|_g=1} \sum_{i=1}^{N} \left\| \log_\mu(\pi_H(p_i)) \right\|_g^2,$$

where $H = \exp_\mu(\text{span}(v))$ and projection map $\pi_H : \mathcal{M} \mapsto H$ is defined by

$$\pi_H(p) := \arg\min_{x \in H} d_g^2(x - p).$$

To define the remaining principal directions, a recursive approach is used as follows:

$$v_k := \arg\max_{\|v\|_g=1} \sum_{i=1}^{N} \left\| \log_\mu(\pi_H(p_i)) \right\|_g^2,$$

where $H = \exp_\mu(\text{span}(\{v_1, \cdots, v_{k-1}, v\}))$.

Although there is a closed-form solution for the PGA problem, Fletcher et al. propose approximating PGA in the tangent space at the Fréchet mean of the data [78]. To achieve this, the points $\{p_i\}_{i=1}^N$ are first mapped to the tangent space using the Riemannian logarithm map. It's worth noting that linear distances in $\mathcal{T}_\mu\mathcal{M}$ between points close to the origin are analogous to the geodesic distances between the corresponding points in $\mathcal{M}$ under the Riemannian exponential map. Therefore, if the original data is highly concentrated around the Fréchet mean, the PGA problem can be well-approximated by the principal component analysis problem of the transformed points via the Riemannian logarithm map. This approach is known as Tangent PCA.

4.1.3 Geodesic Regression

Geodesic regression is a generalization of linear regression on Riemannian manifolds that seeks a geodesic that best fits the relationship between observed values y and regressors x [93].

Formally, given $\{y_i\}_{i=1}^N \in \mathcal{M}$ with associated scalar $\{x_i\}_{i=1}^N \in \mathbb{R}$. For any point $p \in \mathcal{M}$ and tangent vector $v \in \mathcal{T}_p\mathcal{M}$, the geodesic model, which is analogous to

the multiple linear models on Riemannian manifolds, is given by

$$y = \exp_{\exp_p(xv)}(\epsilon),$$

where $\mathcal{M}$-valued random variable y is an observation, scalar $x \in \mathbb{R}$ is a regressor, and random variable $\epsilon \in \mathcal{T}_{\exp_p(xv)}\mathcal{M}$ is the error term. Geodesic regression aims to find a geodesic $\gamma : [0, 1] \longmapsto \mathcal{M}$ by minimizing the sum-of-squared Riemannian distance between all pair of x_i and y_i, for $i = 1, ..., N$, given as follows:

$$\arg\min_{p,v} \frac{1}{2} \sum_{i=1}^{N} d_g\big(\exp_p(x_i v), y_i\big)^2.$$

For specific manifolds, this problem can be solved analytically using Jacobi field techniques.

4.2 Geometric Methods

This section provides a brief overview of several geometric methods across various engineering disciplines, including geometric deep learning, manifold learning, information geometry, and optimal transport [1].

4.2.1 Geometric Deep Learning

Geometric Deep Learning (GDL) is a subfield of deep learning that focuses on developing algorithms and models capable of processing non-Euclidean structured data, particularly data represented as graphs or manifolds [96]. By leveraging the intrinsic geometric properties and topological structures of such data, GDL addresses a wide range of problems across diverse domains, including computer vision, graphics, molecular design, and medical imaging [97–99].

[1] We classify optimal transport as a geometric method due to its extensive theoretical development on general Riemannian manifolds [94, 95].

Prior Assumptions

The core concept of GDL involves generalizing convolutional neural networks (CNNs) to handle irregular, non-grid structured data. Traditional CNNs are specifically designed for images and excel at utilizing data structures with inherent prior assumptions, such as invariance, equivariance, translation invariance, and translation equivariance [100–102]. These assumptions are essential for effectively capturing patterns and features in specific types of data, particularly images. Invariance means that the network's output remains unchanged under certain transformations, while equivariance ensures that the output changes consistently with transformations. Translation invariance and translation equivariance refer to the network's ability to recognize patterns regardless of their position within the data.

Suppose square-integrable functions $\varphi(x) \in L^2(\Omega)$, where $x \in \Omega = [0,1]^d \subset \mathbb{R}^d$ is a compact domain that is a subset of $\mathbb{R}^d$. The representation $\varphi(x)$ is considered an image or signal in a supervised classification setting. The training set consists of pairs of the representation and its corresponding labels, (φ_i, y_i), where $y_i := y(\varphi_i)$ for i in a set $\mathcal{I}$, and label function y maps from the $L^2(\Omega)$ space to a discrete label space $\mathcal{Y}$.

Let $T_c\,\varphi(x) = \varphi(x-v)$ denote the translation of $\varphi \in L^2(\Omega)$ by $c \in \Omega$. We say label function y as being *invariant* if $y(T_c\,\varphi) = y(\varphi)$ holds, and as being *equivariant* if $y(T_c\,\varphi) = T_c\,y(\varphi)$ holds, for all $\varphi \in L^2(\Omega)$ and $c \in \Omega$. Let $D_\tau\,\varphi(x) = \varphi(x - \tau(x))$ denote the deformation of $\varphi \in L^2(\Omega)$ by smooth vector field $\tau : \Omega \longmapsto \Omega$. We say label function y as being *translation-invariant* if $|y(D_\tau\,\varphi) - y(\varphi)| \approx \|\nabla\tau\|$ holds, and as being *translation-equivariant* if $|y(D_\tau\,\varphi) - D_\tau\,y(\varphi)| \approx \|\nabla\tau\|$ holds, for all $\varphi \in L^2(\Omega)$ and τ.

Graph Convolutional Networks

A Graph Convolutional Network is a neural network specifically designed to handle data structured as graphs, making it well-suited for irregular data structures such as social networks and molecular structures [103, 104].

Formally, given an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with N nodes $\{v_i\}_{i=1}^N \in \mathcal{V}$ and edges $\{(v_i, v_j)\}_{i \neq j} \in \mathcal{E}$. Weighted adjacency matrix $A \in \mathbb{R}^{N \times N}$ of graph $\mathcal{G}$ records non-negative edge weights $\{a_{ij} \geq 0\}_{1 \leq i,j \leq N}$.

The architecture of a multi-layer graph convolutional network model is given by the following layer-wise propagation rule:

$$H^{(l+1)} \leftarrow \sigma\big((\bar{D}^{-\frac{1}{2}}\bar{A}\bar{D}^{-\frac{1}{2}})H^{(l)}W^{(l)}\big),$$

where $\bar{A} := A + I_N$ is the adjacency matrix with added self-connections, $\bar{D}_{ii} := \sum_{j=1}^{N} \bar{A}_{ij}$ is the diagonal matrix, $\sigma()$ is an activation function, and $W^{(l)}$ is a trainable weight matrix. $H^{(l)}$ is the matrix of activations in the l^{th} layer, and $H^{(0)}$ is the input data.

SPDNet

SPDNet [105] is the first network architecture specifically designed to process SPD matrices, ensuring that these matrices remain symmetric and positive-definite throughout the learning process. The architecture includes the following layers:

- BiMap: This layer performs the bi-map transformation $WSW^\top$ on covariance matrix S.
- ReEig: This layer performs $U \max(\epsilon e, \Sigma)U^\top$, where ϵ is a rectification threshold and e is the identity matrix.
- LOG: This layer maps S onto its tangent space at e using $U \log(\Sigma)U^\top$.

Here, transformation matrix W is typically required to have full-row rank, and the covariance matrix $S := U\Sigma U^\top$, where U is an orthogonal matrix and Σ is a diagonal matrix.

Riemannian Batch Normalization

Riemannian Batch Normalization (RieBN) is a network architecture that centers and biases batches of SPD matrices using parallel transport on SPD manifolds equipped with affine-invariant Riemannian metric [22]. Formally, given a batch of SPD matrices, the batch centering and biasing in RieBN are given by:

$$\bar{S}_i := \Gamma_{\mathcal{B}\mapsto e}(S_i) = \mathcal{B}^{-\frac{1}{2}} S_i \mathcal{B}^{-\frac{1}{2}};$$
$$\tilde{S}_i := \Gamma_{e\mapsto G}(S_i) = G^{\frac{1}{2}} \bar{S}_i G^{\frac{1}{2}},$$

where $\Gamma_{\mathcal{B}\mapsto e}$ is parallel transport from Riemannian barycenter $\mathcal{B}$ to identity matrix e, and $\Gamma_{e\mapsto G}$ is parallel transport from e to a learnable bias parameter G. The Riemannian barycenter, denoted as $\mathcal{B}$, is the Fréchet mean of that batch, typically calculated using the Karcher flow algorithm.

Furthermore, an iterative updating technique called SPD Momentum Batch Normalization (SPDMBN) has been proposed to address the inaccuracies in estimating the Fréchet mean [3]. This technique, which is employed in Chapter 9, reduces errors in estimating the Fréchet mean of latent data during training. As training progresses, SPDMBN converges to the true Fréchet mean of the latent data. Additionally, SPDMBN can learn the rescaling of SPD matrices by recording the Fréchet variance of datasets, often outperforming RieBN in various scenarios.

Riemannian Optimization

Riemannian optimization is a mathematical framework for solving optimization problems on Riemannian manifolds [106]. Mathematically, it addresses the following optimization problem:

$$\min_{x\in\mathcal{M}} f(x),$$

subject to manifold constraints, where $\mathcal{M}$ is a Riemannian manifold and smooth function $f : \mathcal{M} \longmapsto \mathbb{R}$ is the objective function.

In this study, since the model parameters of geometric deep learning models are manifold-valued, optimizers such as SPDNet and RieBN employ first-order Riemannian adaptive optimization methods [107] to optimize the manifold-valued weights of neural networks. These methods involve two key operations: retraction and parallel transport. Retraction, a first-order approximation of the exponential map on manifolds, enables efficient updates of points on the manifold. Parallel transport involves moving vectors along smooth curves within manifolds equipped with an affine connection. By utilizing these operations, neural networks achieve more precise gradient descent updates in each iteration.

4.2.2 Manifold Learning

Manifold learning is a subfield of machine learning that focuses on projecting high-dimensional data onto lower-dimensional latent manifolds. The primary goal is to visualize the data in a low-dimensional space or to learn the mapping from the high-dimensional space to a low-dimensional embedding.

The first two manifold learning algorithms were introduced in Science in 2000 [108, 109]. These algorithms aimed to reduce high-dimensional data while preserving its intrinsic nonlinear structure often conceptualized as a manifold based on geometric principles from theoretical mathematics, which are particularly relevant in scientific and engineering contexts [110, 111].

Manifold learning operates under the assumption that the data lies on or near a manifold, though the exact dimensions of this manifold are typically unknown. Linear methods like Principal Component Analysis (PCA) [112] may not perform well in such cases, as they are linear projection methods and cannot accurately capture the data's nonlinear structure.

Chronologically, key manifold learning algorithms include Isometric Feature Mapping [108], Local Linear Embedding [109], EigenMaps [113], Diffusion Maps [114], and t-distributed Stochastic Neighbor Embedding (t-SNE) [115]. Each of these algorithms has unique features for reducing data dimensionality. For instance, Isometric Feature Mapping preserves geodesic distances, Local Linear Embedding models nearby points as linear, and EigenMaps utilizes the graph Laplacian to approximate the Laplace-Beltrami operator.

In this study, we consistently employed t-SNE to visualize the dimensionally reduced latent SPD matrices (Chapters 5, 6, and 8). t-SNE is particularly effective at maintaining pairwise similarities among data points in a lower-dimensional space, with a focus on preserving small pairwise distances.

4.2.3 Information Geometry

Information geometry is an interdisciplinary field that studies statistics and probability through the framework of Riemannian geometry [116]. It provides an intuitive approach to determining optimal model parameters in parametric models by

applying Riemannian geometry to the space of these parameters. In information geometry, various dissimilarity measures are employed to capture the nuances of data-model relationships, assess goodness-of-fit, and identify deviations between models [117]. A well-known measure in the machine learning community is the Kullback-Leibler divergence, which is directly related to a core concept in information geometry—the Fisher information metric. The Fisher information metric, a specific Riemannian metric defined on statistical manifolds, represents the infinitesimal form of the Kullback-Leibler divergence [116, 118].

4.2.4 Optimal Transport

The optimal transport problem is a mathematical framework that addresses the efficient and optimal transfer of a mass distribution from one configuration to another [119, 120]. It has been widely applied across various fields, including computer graphics, image processing, and economics, among others [121–124].

Morge Formulation and Monge-Kantorovich Formulation

A separable, completely metrizable topological space is called Polish space. On two Polish spaces X and $\bar{X}$, given a Borel map $T : X \longmapsto \bar{X}$, $T_{\#}$ pushes forward $\mu \in \mathbb{P}(X)$ through T to $T_{\#}\,\mu \in \mathbb{P}(\bar{X})$, i.e., $T_{\#}\,\mu(E) := \mu(T^{-1}(E))$, for any borel $E \subset \bar{X}$.

The Monge formulation [125] of the optimal transport problem seeks to find the most efficient measure-preserving mapping T between two distributions, as determined by a designated cost function, as follows:

$$\min_{T \in S(\mu,\nu)} \int_X c(x, T(x))\, d\mu(x),$$

where $S(\mu, \nu)$ is a set of transports that pushes μ forward to ν, and cost function $c : X \times \bar{X} \longmapsto \mathbb{R}_{\geq 0}$ is a given cost function.

Furthermore, the Monge-Kantorovich formulation [126] revised the Monge formulation as follows:

$$\min_{\gamma \in \prod(\mu,\nu)} \int_{X \times \bar{X}} c(x, \bar{x})\, d\gamma(x, \bar{x}),$$

where π^X and $\pi^{\bar{X}}$ are the natural projections from $X \times \bar{X}$ onto X and $\bar{X}$ respectively, and the transportation plan is as follows:

$$\gamma \in \prod(\mu, \nu) := \{\gamma \in \mathbb{P}(X \times \bar{X}) \big| \pi^X_{\#} \cdot \gamma = \mu, \text{ and } \pi^{\bar{X}}_{\#} \cdot \gamma = \nu\}.$$

Discrete Monge-Kantorovich Formulation

Given a measurable space X and $\bar{X}$, an empirical distribution of the measure $\mu \in X$ and $\nu \in \bar{X}$ are written as $\mu := \sum_{i \in I} p_i \cdot \delta_{x_i}$ and $\nu := \sum_{j \in J} q_i \cdot \delta_{\bar{x}_i}$, where discrete random varaiable X takes values $\{x_i\}_{i \in I} \in X$, countable index set I is with weights $\sum_{i \in I} p_i = 1$. $\bar{X}$ takes values $\{\bar{x}_j\}_{j \in J} \in \bar{X}$, countable index set J is with weights $\sum_{j \in J} q_j = 1$. Dirac measure δ_s denotes a unite point mass at the point $s \in X$ or $\bar{X}$.

The discrete Monge-Kantorovich Formulation is given as follows:

$$\min_{\gamma \in \prod(\mu,\nu)} \big\langle \gamma, c(x, \bar{x}) \big\rangle_{\mathcal{F}}, \tag{4.3}$$

where transportation plan is given by:

$$\gamma \in \prod(\mu, \nu) := \{\gamma \in \mathbb{R}^{|X| \times |\bar{X}|}_{\geq 0} \big| \gamma \cdot \mathbb{1}_{|X|} = \mu, \text{ and } \gamma^\top \cdot \mathbb{1}_{|\bar{X}|} = \nu\},$$

and $\mathbb{1}_d$ is a $d-$dimensional all-one vector.

Brenier's Polar Factorization Theorem

Optimal transport makes use of the concepts of c-transform and c-concave functions to investigate the properties of optimal transport maps, such as their existence, uniqueness, and regularity.

Formally, let $c : X \times X \longmapsto \bar{\mathbb{R}}$ be a cost function and $\varphi : X \longmapsto \bar{\mathbb{R}}$. The c-transform $\varphi^c : Y \longmapsto \bar{\mathbb{R}}$ is defined as follows,

$$\varphi^c(y) := \inf_{x \in X} c(x, y) - \varphi(x),$$

where $\bar{\mathbb{R}} := \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$. A function $\phi : Y \longmapsto \bar{\mathbb{R}}$ is said to be c-concave if there exists $\varphi : X \longmapsto \bar{\mathbb{R}}$ such that $\phi = \varphi^c$.

The c-concave function is used in proving Brenier's polar factorization [127]. Brenier's polar factorization demonstrates that, under certain conditions, optimal transport maps can be expressed as the gradient of a convex function. This result is crucial for understanding the geometric properties of optimal transport maps and for developing efficient algorithms to solve optimal transport problems. A commonly used version of Brenier's polar factorization applies to Euclidean spaces and is formulated as follows:

Let μ and ν be two probability measures on $\mathbb{R}^n$ with compact support, such that μ is absolutely continuous with respect to the Lebesgue measure. Then, there exists a unique optimal transport map $T : \mathbb{R}^n \to \mathbb{R}^n$ that pushes forward μ to ν, i.e., $T_{\#}\,\mu = \nu$, and a convex function $\varphi : \mathbb{R}^n \to \mathbb{R}$ such that $T = \nabla\varphi$.

Optimal Transport-Domain Adaptation

Optimal transport has gained significant popularity in the machine learning community and is particularly useful for tasks such as domain adaptation, generative modeling, and data clustering [128–132]. The Optimal Transport-Domain Adaptation (OT-DA) framework leverages optimal transport to address the challenges of domain adaptation. This framework assumes that the transfer between source and target domains can be effectively estimated using optimal transport. It seeks a transport plan between the empirical distributions of the source domain X_s and the target domain X_t either by interpolating X_s through a barycentric mapping [133] or by estimating a mapping that, while not an exact solution to the Monge formulation, can effectively map unseen samples [134].

Moreover, Joint Distribution Optimal Transport (JDOT) [128] extends the OT-DA framework to address shifts in both marginal and conditional distributions by jointly optimizing the optimal coupling and the prediction function f for the source and target distributions as follows:

$$\min_{\gamma \in \prod(\mu,\nu)} \left\langle \gamma, c(x, y; \bar{x}, f(\bar{x})) \right\rangle_{\mathcal{F}},$$

where y is the label set of samples x, f is the prediction function, and the joint cost function is defined as follows,

$$c(x, y; \bar{x}, f(\bar{x})) := \alpha_1 d_{\ell^2}(x, \bar{x}) + \alpha_2 \mathcal{L}(y, f(\bar{x})),$$

where $\alpha_1, \alpha_2 \geq 0$ and $\mathcal{L}$ is the cross-entropy loss.

Additionally, a deep learning-based extension of JDOT, called DeepJDOT [129], has been proposed to tackle problems in computer vision with a novel loss function as follows:

$$\mathcal{L} := \mathcal{L}\big(y, f(g(x))\big) + \big\langle \gamma, c\big(g(x), y; g(\bar{x}), f(g(\bar{x}))\big)\big\rangle_{\mathcal{F}},$$

where neural network g is typically a convolutional neural network that maps images into feature space.

This deep learning model introduces a novel update method by first employing optimal transport to solve for the transportation plan γ, then updating the neural network parameters using gradient descent, and iteratively repeating these two steps until convergence.

Part II

Geometric Classifier and Its Applications

Chapter 5

Tensor-CSPNet

This section introduces the first geometric classifier, Tensor-CSPNet [1]. Tensor-CSPNet is a geometric deep learning-based classifier that utilizes neural networks on SPD manifolds to capture discriminative information across the temporal, spatial, and frequency domains.

The development of Tensor-CSPNet represents a departure from the conventional use of convolutional neural networks in deep learning-based motor imagery classifiers. This approach addresses the question posed in the introduction: Convolutional neural networks may not be the optimal architecture for motor imagery classification. Indeed, we have identified a category of geometric deep learning architectures that are more naturally aligned with the principles of the traditional CSP method.

Specifically, Tensor-CSPNet incorporates the BiMap layer, which provides a learnable left/right-multiplication transformation closely related to the CSP method. This feature makes Tensor-CSPNet an efficient and effective alternative neural network architecture.

[1] This work has been published in IEEE Transactions on Neural Networks and Learning Systems, 2022, 10.1109/TNNLS.2022.3172108. (IEEE TNNLS 2022)

5.1 Network Architecture

The network architecture of Tensor-CSPNet comprises three steps, aligning with the three rows illustrated in the neural network diagram in Figure 5.1.

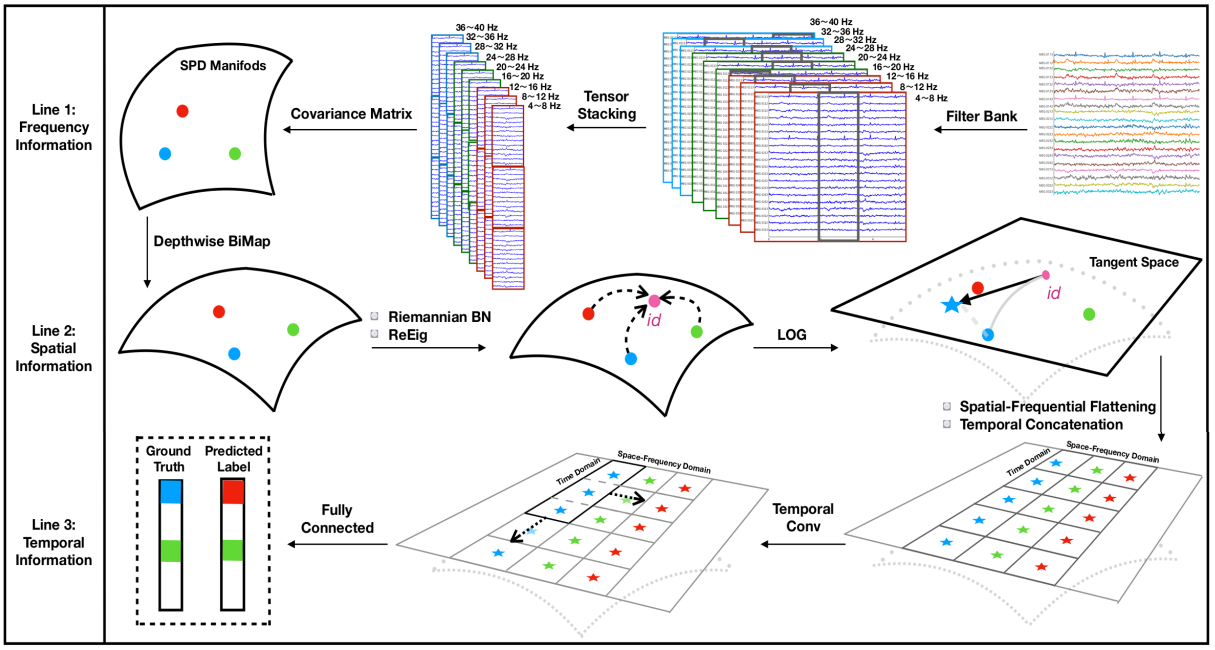


Figure 5.1: Architecture of Tensor-CSPNet: In Line 1, EEG signals undergo tensor stacking, segmenting them into time-space-frequency tensors. Line 2 employs the BiMap, RieBN, and ReEig layers to effectively capture spatial nuances. In Line 3, temporal dynamics are addressed using two-dimensional CNNs on the tangent space. The final classification is performed by fully connected neural networks with a cross-entropy loss function.

Tensor Stacking

During preprocessing, EEG signals are partitioned and guided by the task characteristics, resulting in time-space-frequency tensors.

Frequency Segmentation: This partitioning incorporates filter-bank techniques, specifically employing a bank of bandpass filters, including causal Chebyshev Type II filters, to separate the initial oscillatory EEG signals into distinct non-overlapping frequency passbands. These passbands typically range from 4 to 8 Hz, 8 to 12 Hz, and so forth, up to 36 to 40 Hz.

Temporal Segmentation: EEGs are divided into small time segments. The choice between overlapping or non-overlapping segments is dictated by the specific requirements of the task. For instance, selecting an appropriate segmentation method is crucial in cognitive tasks where rapid changes may occur in short periods. In scenarios where the characteristics of the signal are less defined, a method called the fixed time intervals is employed, where EEGs are divided into short, evenly spaced intervals of consistent duration without any overlap, as described in Appendix 5.3.

Stacking: The inputs of Tensor-CSPNet are these time-space-frequency tensors $\tilde{X} \in \mathbb{R}^{n_W \times n_F \times n_C \times \omega}$ stacked after the frequency and temporal segmentations, i.e.,

$$S^{[ij]} := \tilde{X}[i, j, :, :]\tilde{X}[i, j, :, :]^\top,$$

where n_W, n_F, n_C, and ω are the number of window slices, the number of filter banks, the number of channels, and the window length, respectively, and for $i \in \{1, \cdots, n_W\}$ and $j \in \{1, \cdots, n_F\}$. The pseudocode of the stacking is given as follows,

Algorithm 1: Tensor Stacking without Overlapping

Input : Band-filtered signal $X \in \mathbb{R}^{n_F \times n_C \times n_T}$;
The length of time window ω;
Stride s;
Padding value p.
Output: Stacked tensor $\tilde{X} \in \mathbb{R}^{\lfloor (n_T+2p-1)/s+1 \rfloor \times n_F \times n_C \times \omega}$.

for $i \leftarrow 0$ **to** $\lfloor (n_T + 2p - 1)/s + 1 \rfloor$ **do**
  for $j \leftarrow 0$ **to** n_F **do**
    $\tilde{X}[i, j, :, :] \leftarrow X[j, :, i \times s : i \times s + \omega]$.
  end
end

Common Spatial Pattern Layer

In this layer, we utilize the network architecture of neural networks on SPD manifolds to capture the spatial patterns inherent in EEGs specifically. It is important to note that, unlike combining RGB channels in an image where their information

can be jointly processed, the information from different frequency bands in EEG signals does not seamlessly merge during processing. Consequently, the BiMap layer is crucial for independently extracting features from the covariance matrices of each frequency band.

Temporal Convolutional Layer

We then employ convolutional neural networks in the subsequent layer to capture the temporal dynamics of EEGs on the tangent space. This process begins by implementing spatial-frequency flattening on the outputs from the previous layer. It is followed by a temporal concatenation, where the flattened tensors are combined along the time dimension. After completing these steps, the time-space-frequency tensor transforms into a two-dimensional tensor in $\mathbb{R}^{n_W \times (n_F \times o^2)}$ on tangent space, where n_F is the number of filter banks, and o denotes the output dimension of the common spatial pattern layer, as illustrated in Figure 5.2. Finally, two-dimensional convolutional neural networks with a width of $p \times o^2$ (where p can be either 1 or n_F) and a height of q (where $1 \leq q \leq n_W$) are employed to capture the temporal dynamics of EEGs.

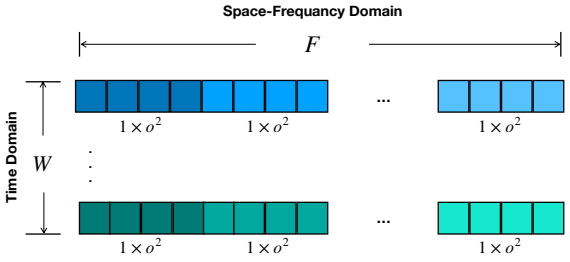


Figure 5.2: Illustration of Temporal Convolutional Layer: F blocks of $1 \times o^2$ rectangles are flattened, and W lines are concatenated, i.e., $n_F = F$ and $n_W = W$. For instance, in the case of 5-CSPNet architecture, the values of n_F, o, and n_W are 9, 20, and 5, respectively. This implies that each line is a 1×3600 flattened tensor, and the entire rectangle's shape is 5×3600.

Remark 5.1.

To mitigate the impact of diverse spatial positions of EEG electrodes on the scalp, the width of the two-dimensional CNNs on the tangent space is defined as a multiple of o^2. The two possible multiples, $p = 1$ and n_F signify that each frequency band within the array can independently and equitably contribute to the model's efficacy.

Loss Function

Tensor-CSPNet utilizes cross-entropy as the loss function, primarily to maintain simplicity.

5.2 Experimental Results

Naming Conventions for Hyperparameters

w-CSPNet$^{(m,n,l)}$ represents the Tensor-CSPNet architecture with w window slices, m banks for filters, n blocks of the common spatial pattern layer, and l-layer neural networks in the perception layer. The number of banks m is set to F and the specific frequency ranges are described in Section 5.1. The depth of the fully-connected layer l always has two options $\{1, 3\}$. The depthwise BiMap's output dimension is o. The hyper-parameters in the temporal convolutional layer are denoted as a portfolio $@(p, q, r)$, where the triple represents the width, the height, and the number of channels of two-dimensional convolutional neural networks, respectively. A summary of the notations is provided in Table 5.1.

5.2.1 Results of Classification Performance

In this section, we assess the performance of Tensor-CSPNet on the KU, BNCI2014001, BNCI2014002, BNCI2015001, and Cho2017 datasets across the 10-fold cross-validation

Table 5.1: Notations for Hyper-Parameters in w-CSPNet$^{(m,n,l)}$ $@(p, q, r)$.

Hyper-Parameters	Meaning
w	The number of time window slices.
m	The number of bandpass filters.
n	The number of the common spatial pattern layer.
l	Depth of the fully connected networks in the classification.
o	The output dimension o of the common spatial pattern layer.
(p, q, r)	The width, height, and output channels in two-dimensional CNNs of the temporal concatenation layer.

and holdout scenarios. The results for the KU and BNCI2014001 datasets are reported in [2], and the results of FBCSP, FBCNet, and Tensor-CSPNet on Cho2017, BNCI2014002, and BNCI2015001 are reported in [25].

The configuration of Tensor-CSPNet is adjusted slightly in each scenario. For instance, the output dimension of the BiMap layer is set to $o = 20$ for the KU dataset and $o = 22$ for the BNCI2014001 dataset. In the 10-fold cross-validation scenarios of two datasets, Tensor-CSPNet employs a shallow neural network structure, specifically denoted as 5-CSPNet$^{(9,1,1)}$. For the holdout scenario, we also utilize shallow neural networks but incorporate a more refined temporal segmentation strategy, specifically 10-CSPNet$^{(9,1,1)}$@$(9, 5, 2)$ and 5-CSPNet$^{(9,1,1)}$@$(9, 5, 4)$, respectively.

The dominant deep learning paradigm in motor imagery classification adopts the time-space-frequency principle. The five deep learning approaches listed in Table 5.2 fall into three distinct categories:

- SPDNet: This model focuses on exploiting the spatial patterns within EEG signals but exhibits the lowest performance among all the deep learning approaches.
- EEGNet and ConvNet: These models capitalize on temporal-spatial patterns within EEG signals and achieve performance levels comparable to FBCSP, which primarily extracts spatial-frequency patterns. This similarity in performance suggests that integrating any two types of these combinations can significantly enhance classification accuracy.
- FBCNet and Tensor-CSPNet: These models are specifically designed based on the time-space-frequency principle, achieving superior performance compared to EEGNet and ConvNet in all scenarios. This advantage can be attributed to using bandpass filters that capture frequency information effectively. Tensor-CSPNet slightly outperforms FBCNet in most scenarios.

In particular, we discuss preliminary evidence highlighting the enhanced performance of Tensor-CSPNet in non-stationary scenarios relative to other methods in Section 5.3.

Table 5.2: Aerage accuracies and standard deviations for subject-specific analyses on the KU, BNCI2014001, BNCI2014002, BNCI2015001, and Cho2017 datasets. Each entry in the table is presented as average accuracy (standard deviation), with the highest-performing value highlighted in bold.

	KU 54 subjects, 20 channels, 2 classes			BNCI2014001 9 subjects, 22 channels, 4 classes		
	CV (S1) %	CV (S2) %	Holdout (S1→ S2) %	CV (T) %	CV (E) %	Holdout (T → E) %
FBCSP	64.41 (16.28)	66.47 (16.53)	59.67 (14.32)	73.57 (15.13)	72.46 (16.02)	65.79 (14.21)
MDM	50.47 (8.63)	51.93 (9.79)	52.33 (6.74)	62.96 (14.01)	59.49 (16.63)	50.74 (13.80)
TSM	54.59 (8.94)	54.97 (9.93)	51.65 (6.11)	68.71 (14.32)	63.32 (12.68)	49.72 (12.39)
SPDNet	57.88 (8.68)	58.88 (8.68)	60.41 (12.13)	65.91 (10.31)	61.16 (10.50)	55.67 (9.54)
EEGNet	63.35 (13.20)	64.86 (13.05)	63.28 (11.56)	69.26 (11.59)	66.93 (11.31)	60.31 (10.52)
ConvNet	64.21 (12.61)	62.84 (11.74)	61.47 (11.22)	70.42 (10.43)	65.89 (12.13)	57.61 (11.09)
FBCNet	74.16 (12.60)	73.81(13.99)	67.83 (14.34)	**77.26** (14.82)	**76.58** (13.09)	72.71 (14.67)
Tensor-CSPNet	**74.95** (15.27)	**75.92** (14.63)	**69.65** (14.97)	75.98 (14.26)	74.92 (14.63)	**72.96** (14.98)
	Cho2017 49 subjects, 20 channels, 2 classes	BNCI2014002 14 subjects, 15 channels, 2 classes	BNCI2015001 12 subjects, 13 channels, 2 classes			
	CV %	CV %	CV(A) %	CV(B) %	Holdout (A → B) %	
FBCSP	61.75 (13.26)	76.07 (13.29)	79.46(14.16)	81.96 (11.14)	73.46 (14.09)	
MDM	51.62 (6.01)	57.37 (10.06)	63.13 (8.18)	62.17 (8.51)	60.9 (7.0)	
TSM	51.79 (5.28)	56.38 (9.99)	61.58 (9.46)	61.92 (7.49)	58.79 (7.41)	
SPDNet	52.88 (5.17)	57.5 (10.08)	61.62 (8.67)	60.42 (5.89)	58.29 (6.27)	
EEGNet	61.15 (3.65)	69.6 (9.26)	65.58 (5.79)	68.08 (7.21)	67.12 (4.11)	
ConvNet	62.18 (4.54)	70.8 (10.87)	67.63 (7.29)	70.54 (6.7)	70.21 (5.26)	
FBCNet	65.34 (11.14)	79.64 (12.77)	**82.62** (13.11)	84.92 (10.30)	74.50 (16.01)	
Tensor-CSPNet	**67.30** (12.94)	**80.58** (11.87)	81.29 (14.78)	**85.29** (10.54)	**79.04** (14.67)	

5.2.2 Results of Interpretability Analysis

In this section, we explore the interpretability of the patterns extracted by Tensor-CSPNet using the Deep Learning Important FeaTures (DeepLIFT) method [135].

To interpret the features extracted, we developed a straightforward visualization method. We converted the four-dimensional relevance pattern obtained from DeepLIFT into a two-dimensional rectangular format, as illustrated in Figure 5.3. Our experiments on Subject No.2 from the KU dataset yielded a test accuracy exceeding 90%. By applying DeepLIFT, we extracted a relevant pattern with an output shape of (5, 9, 20, 20). We transformed this pattern into five rectangular visualizations to enhance the interpretability of significant features. Each rectangle measured 20 grids in height, corresponding to the 20 channels, and 9 grids in width, representing the 9 frequency bands. These rectangles represented distinct time windows: $1.0 \sim 1.5$ s, $1.5 \sim 2.0$ s, $2.0 \sim 2.5$ s, $2.5 \sim 3.0$ s, $3.0 \sim 3.5$ s. The columns in each rectangle were derived from the main diagonal of the covariance matrix within the relevance pattern, which measured 20×20. The values in the heatmap's cells were normalized within the range of [0, 1] and smoothed using a Gaussian filter to clarify the visualization.

The heatmap is organized into the upper row for right-hand movement and the lower row for left-hand movement. Here is how we interpret the heatmap:

- Right-Hand Movement: We note patterns with frequencies ranging from 8 to 28 Hz, particularly highlighted around electrode C3 during the intervals of 1.0 to 1.5 seconds and 2.5 to 3.0 seconds.
- Left-Hand Movement: We see patterns with frequencies ranging from 24 to 28 Hz, specifically highlighted around electrode C4 during the same intervals of 1.0 to 1.5 seconds and 2.5 to 3.0 seconds.

The insights gained from the relevant patterns identified by DeepLIFT align with the known ERD/ERS effects during left-hand and right-hand motor imagery, particularly in the Mu band (8 to 12 Hz) and Beta band (18 to 26 Hz), which are prominent in areas C3 and C4 of the primary motor cortex. These observed active time windows correspond to the frequency changes during the motor imagery process, reinforcing the link between observed neural activity and the specific tasks.

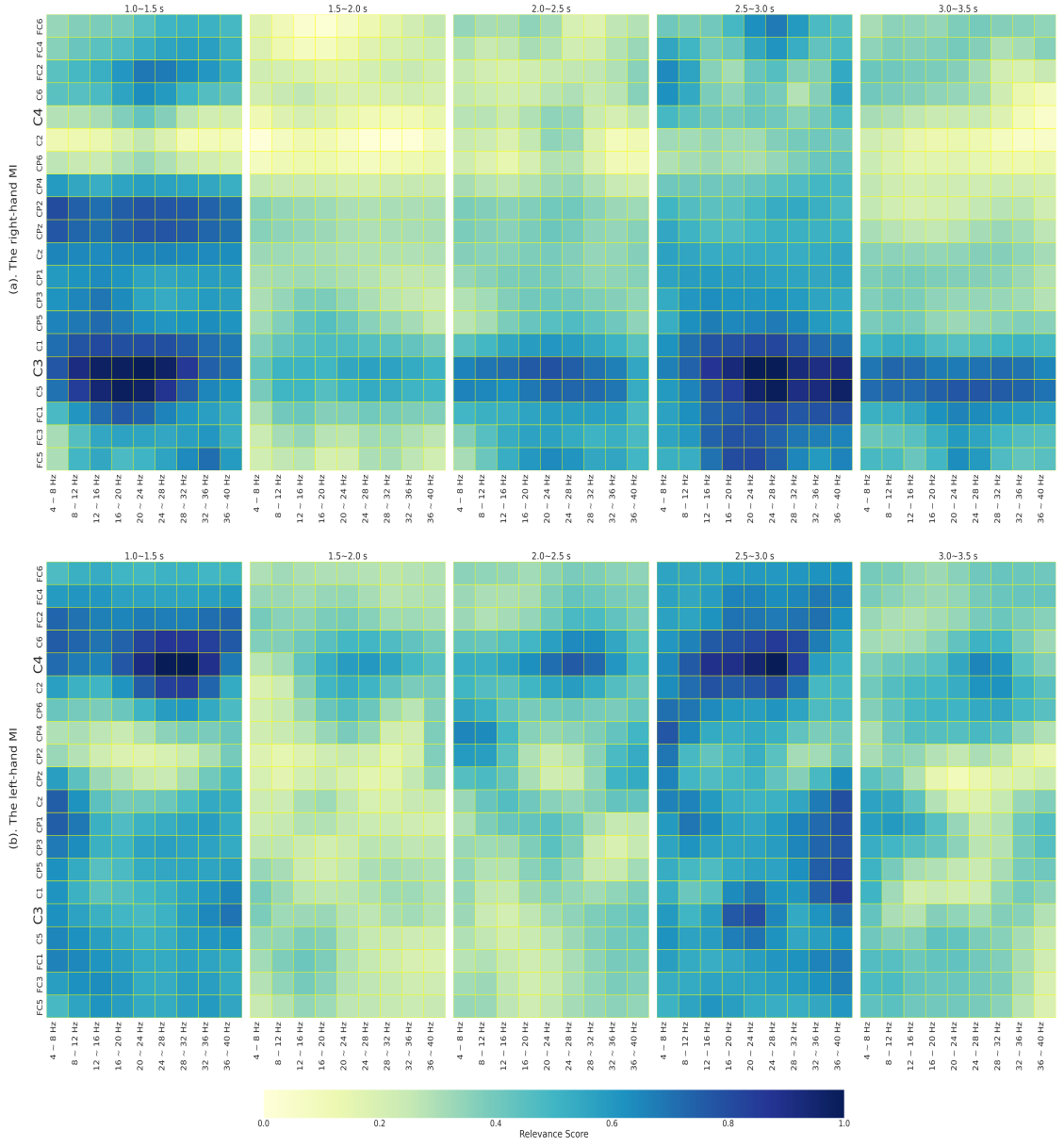


Figure 5.3: Heatmap Visualization of Relevance Patterns for 5-CSPNet$^{(9,1,1)}$.

5.2.3 Results of Visualization

This section demonstrates two-dimensional projections of the outputs from each intermediate layer of Tensor-CSPNet using t-SNE. We aim to explore the underlying mechanisms of Tensor-CSPNet by visualizing the outputs from each intermediate layer in the holdout scenario.

We selected Subject No.28 from the KU dataset, where Tensor-CSPNet achieved an impressive accuracy of over 90%. Due to tensor stacking, the training set is integrated with the validation and test sets, as depicted in Figure 5.4 (a). The common spatial pattern layer normalizes the data shape across each temporal segment of the training, validation, and test sets, resulting in fifteen segments (5× for training/validation/test sets) shown in Figure 5.4 (b). The temporal concentration and two-dimensional convolutional neural networks in the temporal concatenation layer compress these temporal segments along the time domain, as illustrated in Figure 5.4 (c). Moreover, Figure 5.4 (d) shows label-wise projections that clearly differentiate between the classes of labeled data. Class 1 and Class 2 are positioned on the lower and upper sides of the decision boundary, respectively.

5.3 Discussions

Strategy of Fixed-Interval Segmentation

We introduce a fixed-interval segmentation approach in Tensor-CSPNet, which involves dividing the signals into non-overlapping intervals of equal length. This method was applied to ensure that each time window's duration is a divisor of the total EEG signal length. Specifically, we implemented two configurations in the experiments: 5-CSPNet and 10-Cynthia, with time windows of 500 ms and 250 ms, respectively.

Additionally, we found that the optimal value for the height of the two-dimensional convolutional neural networks in the temporal concatenation layer (denoted as q) under the fixed-interval segmentation strategy is equal to the width (n_W). This

finding is corroborated by improved performance as the q value increases, as demonstrated in Table 5.3. This supports the neurobiological perspective that larger time window sizes more effectively detect ERD/ERS effects during motor imagery tasks.

Table 5.3: The outcomes in the holdout scenario of the KU dataset with varying q values in the temporal concatenation layer of 5-CSPNet and 10-CSPNet. The parameter q represents the height of two-dimensional CNNs within the temporal concatenation layer.

q	1	2	3	4	5
5-CSPNet	63.5%	65.1%	66.0%	66.8%	**67.6%**
q	6	7	8	9	10
10-CSPNet	66.8%	67.4%	67.8%	67.2%	**68.4%**

Temporal Segmentation Against Non-Stationarity

To investigate the notable performance of Tensor-CSPNet in the non-stationary (between-session/holdout) scenario, we focus on Subject No.28 from the KU dataset, whose holdout accuracy exceeds that of FBCSP by 0.3. Figure 5.5 shows a pattern where more refined temporal segmentation results in a larger overlap region covering the training, validation, and test sets within the statistical distribution space.

From a statistical perspective, temporal segmentation addresses non-stationarity, aligning with insights from a well-established theory known as segmentation techniques for nonstationary EEGs, proposed over four decades ago [136]. When implemented with neural signals, fixed-interval temporal segmentation divides EEG signals into numerous short, contiguous intervals that exhibit quasi-stationary characteristics. This segmentation effectively minimizes the impact of drift across different sessions, which is beneficial for classification performance when using a statistical classifier.

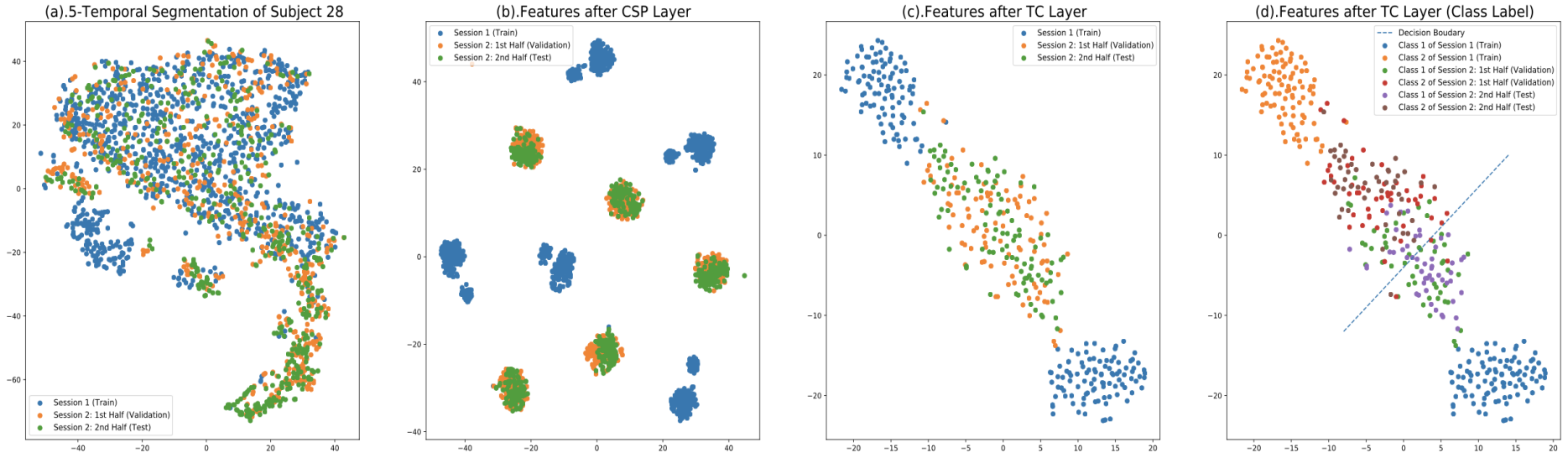


Figure 5.4: Visualization of Intermediate Outputs in 5-CSPNet$^{(9,1,1)}$ with $o = 22$ for Subject No.28 of the KU dataset Using t-SNE: The model processes EEG signals using time windows ranging from $1 \sim 1.5$ s, $1.5 \sim 2.0$ s, $2.0 \sim 2.5$ s, $2.5 \sim 3.0$ s, to $3.0 \sim 3.5$ s. Figure (a) illustrates the original EEG signals divided into 5 time windows using a segmentation strategy. Figure (b) displays the outputs from the common spatial pattern layer, where blue and yellow/green clusters correspond to two distinct motor imagery tasks. Each cluster is further subdivided into five sub-clusters corresponding to the individual time windows. Figure (c) shows the outputs following the temporal concatenation layer, where the blue and yellow/green clusters merge into two more cohesive clusters, with the yellow/green cluster positioned between the blue clusters. Finally, Figure (d) presents the outputs after further processing in the temporal concatenation layer, along with the associated class labels. Here, the blue and yellow/green clusters are separated by a decision boundary, with the two classes nearly symmetrically distributed on either side of this boundary.

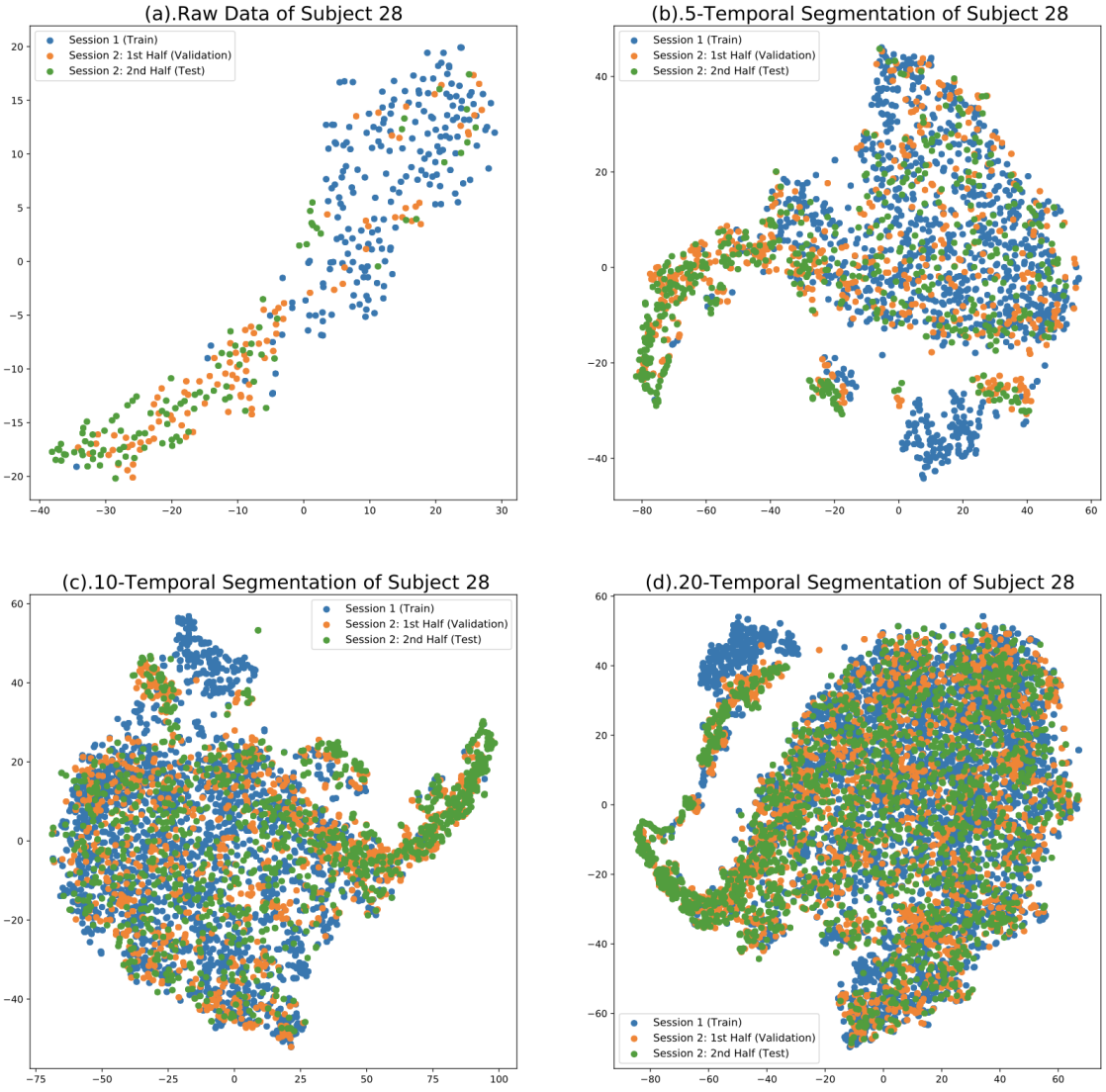


Figure 5.5: Two-dimensional Projection of Subject No.28 in the KU dataset Using t-SNE: Each subject participates in two sessions. The first session (S1) is used as the training set, while the two halves of the second session (S2) are designated as the validation and test sets. The lengths of the time windows are (a) 2500 ms, (b) 500 ms, (c) 250 ms, and (d) 125 ms, with no overlap between them. Each point in the two-dimensional projection is obtained by reducing the dimensionality of a $9 \times 20 \times 20$-dimensional point, which includes 20 electrodes in the motor cortex region and spans nine frequency bands. This format serves as the input to Tensor-CSPNet.

Chapter 6

Graph-CSPNet

The non-stationary nature of EEG spectral contents often undermines the effectiveness of traditional methods that assume statistical stationarity [137]. In such cases, conventional Fourier analysis often falls short. To overcome this limitation, time-frequency analysis techniques are utilized to effectively localize rhythmic components in real-time, thereby enhancing motor imagery classifiers [138–141]. Techniques such as the wavelet transform are examples of this approach [142–144]. These methods have been incorporated into the CSP framework to develop innovative wavelet-based classifiers, including the wavelet-CSP classifier [145].

Drawing inspiration from Gabor's time-frequency theory and Morlet's wavelet theory [146–148], this chapter aims to refine Tensor-CSPNet to more adeptly identify local oscillatory components in EEG signals. The proposed model, Graph-CSPNet [1], employs graph-based neural networks to tackle the challenges of motor imagery classification. A key innovation in this model is the novel graph BiMap layer, which moves away from the fixed-length segmentation in Tensor-CSPNet and instead adopts a flexible time-frequency resolution that effectively captures localized fluctuations, enabling nuanced analysis within the time-frequency domain.

In the graph BiMap layer of the model, each EEG spatial covariance matrix is represented as a node in a graph. This graph encapsulates the local topology through a cutting-edge time-evolution method. The connections, or edges, between the nodes in the graph signify the similarity among neighboring nodes. This similarity

[1] The work in this chapter has been published in IEEE Transactions on Neural Networks and Learning Systems, 2023, 10.1109/TNNLS.2023.3307470. (IEEE TNNLS 2023)

is quantified by applying a Gaussian kernel to the Riemannian distance between pairs of nodes located on SPD manifolds. Consequently, the task of motor imagery classification is redefined as a graph classification challenge, where the classification is based on the time-frequency distribution of EEG signals.

6.1 Network Architecture

The architecture of Graph-CSPNet is illustrated in Figure 6.1, with detailed parameters for each layer provided in Section 6.3.

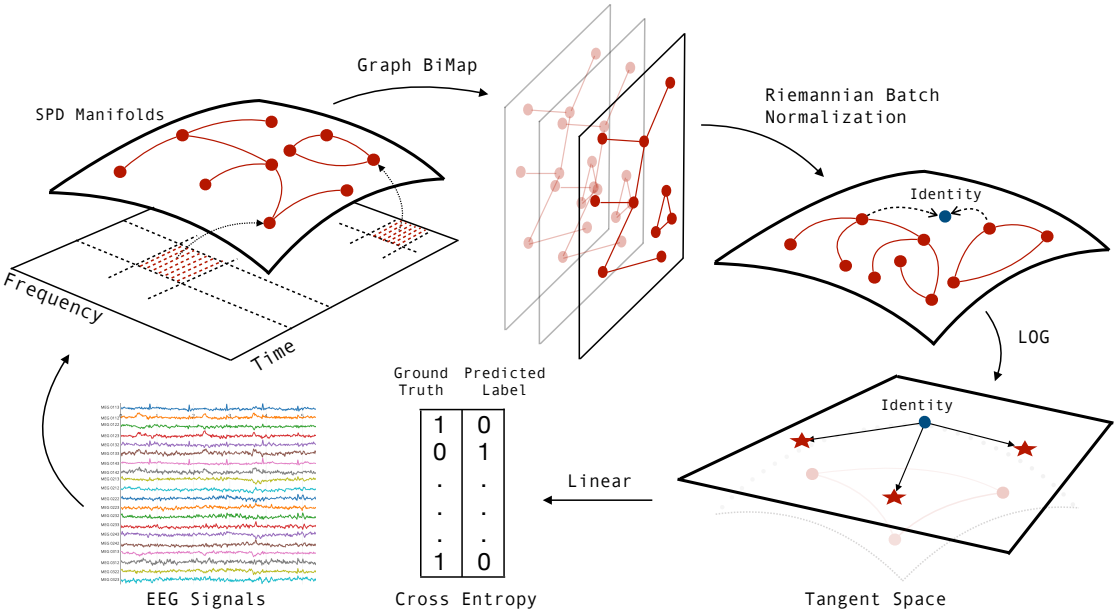


Figure 6.1: Architecture of Graph-CSPNet: In this structure, the EEG signal is segmented into multiple divisions within the time-frequency domain. The spatial covariance matrices derived from these segments form the vertices of the time-frequency graph, which is constructed using an innovative nonparametric statistical method. After establishing the time-frequency graph, an SPD matrix-valued graph convolutional network is deployed. This network includes the Graph-BiMap layer and RieBN, which together extract crucial classification information while preserving the ability to distinguish between different task classes. Subsequently, the LOG layer maps the SPD matrices onto the tangent space. These transformed matrices are then fed into the cross-entropy layer for computations.

Time-Frequency Distribution

A novel time-frequency distribution consisting of SPD matrices derived from EEG segments is constructed in the following way: given a segmentation plan on the time-frequency domain with or without overlapping $\{\Delta t_i \times \Delta f_i\}_{i\in\mathcal{I}}$, the time-frequency distribution consists of EEG spatial covariance matrices $\{S(\Delta t_i \times \Delta f_i)\}_{i\in\mathcal{I}}$, where $S(\Delta t_i \times \Delta f_i) := \bar{X}_i \bar{X}_i^\top$ is the covariance matrix of band-pass filtered EEG signal $\bar{X}_i \in \mathbb{R}^{n_C \times \Delta t_i}$ within bandwidth Δf_i. The determination of a segmentation plan $\{\Delta t_i \times \Delta f_i\}_{i\in\mathcal{I}}$ relies on changes in ongoing EEG activity, as evidenced by the appearance of the ERD/ERS effect that is induced by cognitive and motor processing. A localized decrease in amplitude characterizes the ERD effect, while an increase in rhythmic activity amplitude marks the ERS effect. Both of these effects are highly specific to the frequency band of the event. It is essential to consider the frequency discretization Δf when working with traditional frequency bands, including δ (< 4 Hz), θ ($4 \sim 7$ Hz), μ ($8 \sim 13$ Hz), β ($14 \sim 30$ Hz), and γ (> 30 Hz) activity, which aligns closely with neurophysiological mechanisms. Due to the subject/user-specific nature of event-related discrimination, the discretization of frequency bands and time intervals can vary depending on the experiment.

Remark 6.1. To illustrate, consider the task of imagining hand movements. A time resolution of $\Delta t = 125$ ms is optimal for effectively detecting the occurrence of ERD/ERS effects, particularly within the Mu and Beta frequency bands, which are crucial for accurate discrimination. Selecting the appropriate time resolution is essential to ensure that neural activity in these specific frequency bands is accurately captured and analyzed [13, 39].

Time-Frequency Graph

To characterize the time-frequency distribution mentioned above, we construct a time-frequency graph denoted as $\mathcal{G}(\mathcal{V}, \mathcal{E})$. The vertices of this graph represent EEG spatial covariance matrices in the time-frequency distribution. The edges of the time-frequency graph are created using a nonparametric statistical approach called the ϵ-neighborhoods approach. This approach involves connecting two vertices, S_i and S_j, with an edge if $d_{g^{AIRM}}(S_i, S_j)^2 < \epsilon$.

Expressly, we assume that brain activities generate a time evolution effect on the power spectrum of the EEG signals along the time axis. To capture this effect, we employ a modified ϵ-neighborhoods approach that establishes adjacency between two vertices, $S_i = S(\Delta t_i \times \Delta f_i)$ and $S_j = S(\Delta t_j \times \Delta f_j)$, if they fall within a box $\mathcal{B}_{\epsilon_1,\epsilon_2}$ in the time-frequency domain. This box is defined by a box width $\epsilon_1 \geq 0$ on the time axis and a box height $\epsilon_2 \geq 0$ on the frequency axis. Therefore, S_i and $S_j \in \mathcal{B}_{\epsilon_1,\epsilon_2}$ must satisfy the following conditions:

$$\begin{aligned} 0 \leq \text{mid}(\Delta t_i) - \text{mid}(\Delta t_j) \leq \epsilon_1; \\ |\text{mid}(\Delta f_i) - \text{mid}(\Delta f_j)| \leq \epsilon_2, \end{aligned}$$

where *mid* represents the midpoint value of an interval. Note that the time evolution in this study occurs in the forward time flow direction, making it unnecessary to take the absolute value of the difference between the midpoints of Δt in the above formula.

The adjacency matrix A of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is used to store the similarities between pairs of vertices and is defined as follows,

$$A = \begin{cases} e^{-d^2_{gAIRM}(S_i,S_j)/t}, & \text{if } S_i \text{ and } S_j \text{ are adjacent;} \\ 0, & \text{others.} \end{cases}$$

where preset Gaussian kernel width $t > 0$. These similarities are calculated using the radial basis function kernel, which incorporates the Riemannian distance between adjacent vertices. For vertices that are not adjacent, the similarity is explicitly set to zero, meaning that if S_i and S_j are non-adjacent, their similarity score is assigned a value of zero. The pseudocode for generating adjacency matrix A refers to Section 6.3.

This technique is named the Local Graph Topology (LGT) method, as it effectively captures the local interactions in the time-frequency distribution. Utilizing the LGT method gives us a more comprehensive understanding of the intricate connections between EEG spatial covariance matrices in the time-frequency distribution.

Remark 6.2. 1). Edge Weight: The weight of the edge in the time-frequency graph is calculated using a radial basis function kernel with Riemannian distance, as

the discriminative power between task classes is closely related to the Riemannian distance.

2). Connected Component: According to the LGT approach, each frequency component (θ, μ, β, and γ bands) forms a connected component in the adjacency matrix A of the time-frequency graph, as shown in Table 6.1,

$$A = \begin{pmatrix} A_\theta & & & \\ & A_\mu & & \\ & & A_\beta & \\ & & & A_\gamma \end{pmatrix}.$$

Graph BiMap Layer

This section introduces an SPD matrix-valued graph neural network to extract discriminative information from the time-frequency graph. To address this, we construct the layer-wise propagation rule for this SPD matrix-valued graph neural network as follows:

$$H^{(l+1)} \leftarrow \mathrm{RieBN}\Big(\mathrm{ReEig}\big(W^{(l)}(\bar{D}^{-1}\bar{A}^{(l)})H^{(l)}{W^{(l)}}^{\top}\big)\Big),$$

where $\bar{A}^{(l)} := A^{(l)} + I_N$, $\bar{D}_{ii} := \sum_j \bar{A}^{(l)}_{ij}$, and $W^{(l)}$ is a trainable transformation matrix with the full-row rank. The RieBN and ReEig layers are introduced in Chapter 4. $H^{(l)} \in \mathbb{R}^{|\mathcal{V}|\times n_C^2}$ is a node function in the l^{th} layer. In particular, $H^{(0)} := (S^1, \cdots, S^N)$, $A^{(0)}$ is the adjacency matrix of the time-frequency graph, and $\bar{A}^{(l)} := I_N$, for $l \geq 1$.

The proposed layer-wise propagation rule is structurally similar to the classical rule in graph convolutional networks. The linear transformation is replaced with the BiMap transformation that facilitates operations on SPD matrices. The BiMap transformation is essential as it ensures that Riemannian distances are preserved throughout the learning process while also maintaining the symmetry and positive definiteness of the matrices. This attribute is crucial for effectively performing motor imagery classification tasks. Furthermore, instead of using $\bar{D}^{-\frac{1}{2}}\bar{A}^{(l)}\bar{D}^{-\frac{1}{2}}$, we employ row-normalized $\bar{D}^{-1}\bar{A}^{(l)}$ because we aim to perform normalization only in the time direction. Additionally, we utilize the non-linear operator ReEig, which drops the smallest eigenvalues to prevent matrix degeneracy.

However, the proposed rule still differs from the classical one. In the following, we will discuss these differences in detail, specifically focusing on their relevance to our application. A perturbation analysis for the spectrum change induced by transformation $\bar{D}^{-1}\bar{A}^{(l)}$ is presented as follows,

Theorem 6.1. *Given a time-frequency graph $\mathcal{G}(\mathcal{V},\mathcal{E})$. For the l^{th} graph BiMap layer, let $A^{(l)}$ be its $|\mathcal{V}|\times|\mathcal{V}|$ adjacency matrix. For $i\in\{1,...,|\mathcal{V}|\}$, the perturbated spatial covariance matrix $\bar{S}_i\in\mathbb{R}^{n_C\times n_C}$ on $(\mathcal{S}_{++},g^{AIRM})$ is wtitten in to the original spatial covariance matrix S_i and a graph-based perturbation term, i.e., $\bar{S}_i=S_i+\Delta S_i$, where $\Delta S_i:=A^{(l)}[i,:](S_1,\cdots,S_{|\mathcal{V}|})^\top$. Then, the spectrum of the spatial covariance matrix has a perturbation ratio as follows,*

$$\big(1-N_iC_i\big)\leq\frac{\lambda(\bar{S}_i)}{\lambda(S_i)}\leq\big(1+N_iC_i\big), \tag{6.1}$$

where $C_i=\max_{(i,j)\in\mathcal{E}}\{\exp\{(\lambda_{ij})\}\}$, $N_i=\Big|\{j\Big|(i,j)\in\mathcal{E}\}\Big|$, and λ_{ij} is largest eigenvalue of $\log\Big(S_i^{-\frac{1}{2}}S_jS_i^{-\frac{1}{2}}\Big)$.

We initially establish the subsequent Lemma to prove Theorem 6.1.

Lemma 6.2. *Let $\{S_i\}_{i=1}^N$ be a set of SPD matrices. Then, any linear combination $\sum_{i=1}^N\alpha_iS_i$ is still SPD for $\alpha_i>0$.*

Proof. The symmetry is obtained by $(\sum_{i=1}^N\alpha_iS_i)^\top=\sum_{i=1}^N\alpha_iS_i^\top=\sum_{i=1}^N\alpha_iS_i$, and the positive definite is achieved by $v^\top(\sum_{i=1}^N\alpha_iS_i)v=\sum_{i=1}^N\alpha_i(v^\top S_iv)>0$, for any $v\neq0$. This concludes the proof. □

Proof of Theorem 6.1. Consider the perturbation ΔS_i on each node S_i ($1\leq i\leq|\mathcal{V}|$), which is taken as a linear combination of adjacent nodes as $S_i+\Delta S_i=S_i+\sum_{(i,j)\in\mathcal{E}}a_{ij}S_j$, where $a_{ij}:=e^{-d^2_{g^{AIRM}}(S_i,S_j)/t}\leq1$ with a large t is the similarity value for the edge between node S_i and S_j in adjacency matrix $A^{(l)}$. Each $S_i+\Delta S_i$ remains symmetric positive and definite according to Lemma 7.1.

The Riemannian distance $d_{g^{AIRM}}(S_i,S_j)$ between node S_i and its adjacent node S_j is $||\log\Big(S_i^{-\frac{1}{2}}S_jS_i^{-\frac{1}{2}}\Big)||_{\mathcal{F}}$. Notice that $\log\Big(S_i^{-\frac{1}{2}}S_jS_i^{-\frac{1}{2}}\Big)$ is a $n_C\times n_C$ symmetric matrix and bounded by an inequality according to the Rayleigh-Ritz theorem [149] as $-\lambda_{ij}I\preceq\log\Big(S_i^{-\frac{1}{2}}S_jS_i^{-\frac{1}{2}}\Big)\preceq\lambda_{ij}I$, where λ_{ij} is the largest Rayleigh quotient

of $\log\Big(S_i^{-\frac{1}{2}} S_j S_i^{-\frac{1}{2}}\Big)$. Since $\exp(\lambda_{ij})$ is the eigenvalue of $S_i^{-\frac{1}{2}} S_j S_i^{-\frac{1}{2}}$ according to the definition of exponential matrix and exp is monotone increasing, we have the following inequality:

$$-\exp(\lambda_{ij})I \preceq S_i^{-\frac{1}{2}} S_j S_i^{-\frac{1}{2}} \preceq \exp(\lambda_{ij})I.$$

Note that the pertubation $S_i + \Delta S_i$ can be written as follows,

$$S_i + \Delta S_i = S_i + \sum_{(i,j)\in\mathcal{E}} a_{ij} S_j = S_i^{\frac{1}{2}}\Big(I + S_i^{-\frac{1}{2}}\big(\sum_{(i,j)\in\mathcal{E}} a_{ij} S_j\big) S_i^{-\frac{1}{2}}\Big) S_i^{\frac{1}{2}}.$$

Then, we have $\big(1 - N_i C_i\big) S_i \preceq S_i + \sum_{(i,j)\in\mathcal{E}} a_{ij} S_j \preceq \big(1 + N_i C_i\big) S_i$, where $N_i = |\{j|(i,j) \in \mathcal{E}\}|$ and $C_i = \max_{(i,j)\in\mathcal{E}}\{\exp\{(\lambda_{ij})\}\}$. Lastly, the spectrum perturbation ratio in Equation 6.1 can be obtained using Weyl's monotonicity theorem [150]. □

Theorem 6.1 provides both a rough estimated upper and lower bounds for spectrum change ratio $\lambda(\bar{S}_i)/\lambda(S_i)$, for $1 \leq i \leq N$. This ratio depends solely on the node degree and spectrum between vertices in the time-frequency graph. As the depth of graph BiMap layers increases, the spectrum change ratio increases accordingly.

To uphold the lowest possible spectrum change ratio, $\bar{A}^{(0)}$ is designated as the adjacency matrix of the time-frequency graph, and $\bar{A}^{(l)} := I_N$, for $l \geq 1$. The visualization of the impact of spectrum changes, as stated in Theorem 6.1, is discussed in detail in Section 6.3.

Loss Function

Following the graph BiMap layers, the resulting SPD matrices are mapped onto the tangent space through the LOG layer. This mapping approach is well-defined and applicable to any SPD matrix To streamline the model, we selected the cross-entropy as the loss function for Graph-CSPNet.

6.2 Experimental Results

Evaluated Segmentation Plans: This study presents the non-overlapping and non-uniform segmentation plans, as shown in Table 6.1, to evaluate the proposed approach. The term *non-overlapping* indicates no overlap between time windows, while *non-uniform* implies that the size of the time windows varies within the frequency band. We considered the commonly used method of segmenting signals in the time domain into units of seconds, half-seconds, or quarter-seconds, which BCI researchers have widely adopted in the past years, but it is not based on extensive neurophysiological considerations. In Table 6.1, there are 60 segments of EEG signals in the KU, BNCI2014002, and BNCI2015001 datasets (i.e., 60 segments = 5 windows $\times$ 6 frequency bands + 10 windows $\times$ 3 frequency bands), 48 segments in the BNCI2014001 dataset (i.e., 48 segments = 4 windows $\times$ 6 frequency bands + 8 windows $\times$ 3 frequency bands), and 33 segments in the Cho2017 dataset (33 segments = 3 windows $\times$ 6 frequency bands + 5 windows $\times$ 3 frequency bands).

Evaluation Baselines: We will assess Graph-CSPNet by benchmarking it against established baseline methods such as FBCSP, FBCNet, and Tensor-CSPNet. It is important to highlight that our proposed method fundamentally deviates from the traditional approach of utilizing EEG device channels as graph nodes in motor imagery classifiers. Therefore, other graph-based classifiers are not included in this comparison, as they fall outside the methodological scope of this study.

Configurations of Network Architecture:

FBCNet features relatively few tunable hyperparameters. We utilized the same nine frequency sub-bands as FBCSP and configured 16 parallel spatial convolution blocks across all datasets.

For Tensor-CSPNet and Graph-CSPNet, we implemented straightforward yet efficient network architectures for different scenarios, as outlined below:

- Tensor-CSPNet: The network architecture features a two-layer BiMap block, which includes the BiMap layer, the ReEig layer, and the RieBN layer. The input and output dimensions of this block vary across different datasets. For all datasets, frequency segmentation is uniformly performed in 4 Hz bandwidth increments, ranging from 4 Hz to 40 Hz, without any overlap. However,

the approach to time segmentation is designed for each specific dataset, as outlined below: 1). For the KU dataset, the BiMap block transforms the input dimension from 20 to 30 and then back to an output dimension of 20. The network employs three temporal segmentations: 0 to 1.5 seconds, 0.5 to 2 seconds, and 1 to 2.5 seconds. 2). In the Cho2017 dataset, the BiMap block converts the input dimension from 20 to 30 and then reverts it to an output dimension of 20. The network is divided into three temporal segments: 0 to 1 second, 1 to 2 seconds, and 2 to 3 seconds. 3). For the BNCI2014001 dataset, the BiMap block transforms the input dimension from 22 to 36 and then back to an output dimension of 22. This network has two temporal segmentations: 0 to 0.75 seconds and 0.25 to 1 second. 4). For the BNCI2014002 dataset, the BiMap block increases the input dimension from 15 to 30 and then reduces it back to an output dimension of 15. This network employs five temporal segmentations: 0 to 1 second, 1 to 2 seconds, 2 to 3 seconds, 3 to 4 seconds, and 4 to 5 seconds. 5). For the BNCI2015001 dataset, the BiMap block expands the input dimension from 13 to 30 and then reduces it to an output dimension of 13. This network utilizes five different temporal segmentations: 0 to 1 second, 1 to 2 seconds, 2 to 3 seconds, 3 to 4 seconds, and 4 to 5 seconds.

- Graph-CSPNet: Graph-CSPNet shares the same neural network architecture as Tensor-CSPNet across all scenarios. However, it employs a different segmentation approach for frequency and time. The specific time-frequency segmentation plan for Graph-CSPNet is outlined in Table 6.1. In addition, we typically set the forward time flow to half their maximum possible value, as discussed in Section 6.2.2.

6.2.1 Results of Classification Performance

The evaluation is conducted using subject-specific scenario settings across two datasets, namely KU and BNCI2014001. These datasets have been reinitialized with cross-validation indices and employ different network architectures than those used in Chapter 5. As a result, the classification accuracies of each baseline may slightly differ from earlier results, with variations within an acceptable range of approximately 1%. Additionally, three other datasets are being used for the first time in this study. For each scenario, we select the best result from multiple runs.

Table 6.1: A non-overlapping and non-uniform segmentation plan for Graph-CSPNet. The table provided lists each frequency band's time-window length (seconds) as a distinct entity.

	θ band	μ band	β band				γ band		
Dataset/Freq Band (Hz)	$4 \sim 8$	$8 \sim 12$	$12 \sim 16$	$16 \sim 20$	$20 \sim 24$	$24 \sim 28$	$28 \sim 32$	$32 \sim 36$	$36 \sim 40$
KU/Cho2017	0.5	0.5	0.5	0.5	0.5	0.5	0.25	0.25	0.25
BNCI2014001	0.25	0.25	0.25	0.25	0.25	0.25	0.125	0.125	0.125
BNCI2014002/2015001	1	1	1	1	1	1	0.5	0.5	0.5

Table 6.2: Average accuracies and corresponding standard deviations derived from subject-specific analyses of the KU, Cho2017, BNCI2014001, BNCI2014002, and BNCI2015001 dataset. Each result in the table is expressed as the average accuracy accompanied by its corresponding standard deviation. Notably, the optimal outcome for each analysis is highlighted in boldface, thus providing an enhanced visual representation of the best-performing metrics.

Dataset	Scenario	FBCSP	FBCNet	Tensor-CSPNet	Graph-CSPNet
KU	10-Fold CV (S1) %	64.33 (15.43)	**73.36** (13.71)	73.28 (15.10)	72.51 (15.31)
	10-Fold CV (S2) %	66.20 (16.29)	73.68 (14.97)	74.16 (14.50)	**74.44** (15.52)
	Holdout (S1 $\rightarrow$ S2) %	59.67 (14.32)	67.74 (14.52)	69.50 (15.15)	**69.69** (14.72)
Cho2017	10-Fold CV %	61.75 (13.26)	65.34 (11.14)	67.30 (12.94)	**67.51** (12.89)
BNCI2014001	10-Fold CV (T) %	71.29 (16.20)	75.48 (14.00)	75.11 (12.68)	**77.55** (15.63)
(BCIC-IV-2a)	10-Fold CV (E) %	73.39 (15.55)	77.16 (12.77)	77.36 (15.27)	**78.82** (13.40)
	Holdout (T $\rightarrow$ E) %	66.13 (15.54)	71.53 (14.86)	**73.61** (13.98)	71.95 (13.36)
BNCI2014002	10-Fold CV %	76.07 (13.29)	79.64 (12.77)	80.58 (11.87)	**81.65** (11.74)
BNCI2015001	10-Fold CV (A) %	79.46 (14.16)	82.62 (13.11)	81.29 (14.78)	**84.62** (12.38)
	10-Fold CV (B) %	81.96 (11.14)	84.92 (10.30)	85.29 (10.54)	**88.00** (7.87)
	Holdout (A $\rightarrow$ B) %	73.46 (14.09)	74.50 (16.01)	79.04 (14.67)	**79.75** (14.63)

All motor imagery classifiers adopt the time-space-frequency principle for classification. The segmentation technique utilized by Tensor-CSPNet and Graph-CSPNet enhances their performance relative to FBCNet across three holdout scenarios. This technique segments EEG signals into short, quasi-stationary intervals, minimizing distribution shifts and enhancing classifier robustness. Among the algorithms, Graph-CSPNet exhibits a slight advantage in nine out of the eleven scenarios listed in Table 6.2. This may be largely due to its refined time-frequency segmentation technique, which provides a more precise characterization of EEG signals.

6.2.2 Ablation Study of Hyperparameters

We will explore the influence of hyperparameters on the performance of Graph-CSPNet, focusing specifically on the parameter ϵ used in the LGT method within the time-frequency graph. This analysis is depicted in Figure 6.2. To streamline the evaluation process, the network configuration of Graph-CSPNet will be simplified to include a single-layer graph BiMap layer with input and output dimensions set to 20.

- The utilization of graph topology enables Graph-CSPNet to learn discriminative patterns from the time-frequency graph effectively. This is evident from the significant improvements observed in other time-frequency graph configurations compared to the *no graph* case, i.e., Time Direction (0,0,0,0) in Figure 6.2. The *no graph* case refers to the similarity matrix $A^{(0)} := I_N$.
- Among different time directions, Time Direction (2,2,2,4) yields the best performance, as demonstrated in Figure 6.2. The corresponding transformed spectral distributions can be found in Figure 6.4 (c). Additionally, Time Direction (2,2,2,4) showed statistical significance in the 10-fold cross validation experiments of the two groups. However, it did not exhibit significance in the holdout experiment. Statistical significance was determined using the one-tailed Wilcoxon signed-rank test [151] with Bonferroni-Holm correction, with a significance level of $\alpha = 0.05$. Therefore, in our experiments, we generally set the forward time flow to the nearest integer values, which is the closest approximation to half-values.

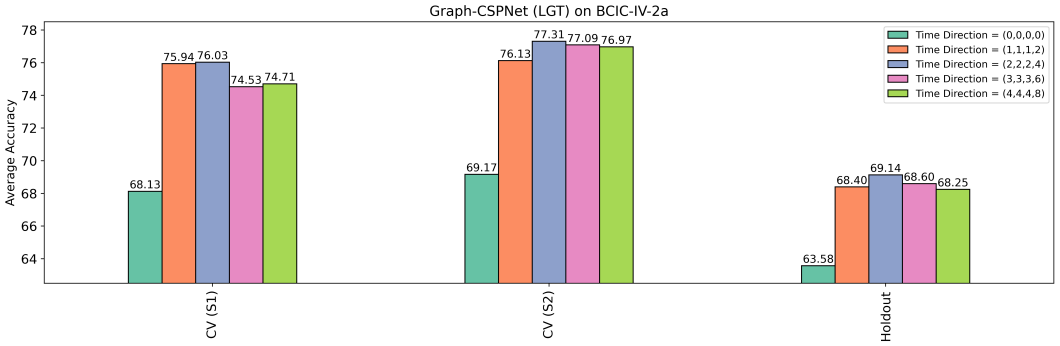


Figure 6.2: Classification performance of Graph-CSPNet with multiple time and frequency directions on the BNCI2014001 (BCIC-IV-2a) dataset. The (forward) time direction includes $0 \leq x_\theta \leq 4$, $0 \leq x_\mu \leq 4$, $0 \leq x_\beta \leq 4$, and $0 \leq x_\gamma \leq 8$. The frequency direction is preset to (1,1,4,3).

6.3 Discussions

In the discussions, we delve into various aspects of Graph-CSPNet and compare it with Tensor-CSPNet, highlighting their similarities and differences.

Preset Edge Weights

The edge weights in the graph are calculated by averaging Riemannian distances between EEG spatial covariance matrices across training samples. In a 10-fold cross-validation scenario, adjacency matrix A is initialized ten times, with minor variations between each fold. Consequently, the adjacency matrix A is specific to each individual, leading to a Graph-CSPNet architecture that is uniquely specific to each participant.

Spectrum Distribution Shift

The proposed LGT method serves as a nonparametric statistical approach for initializing the topology of the time-frequency graph. According to Theorem 6.1, both approaches will introduce changes in the distribution of the spectrum power of spatial covariance matrices.

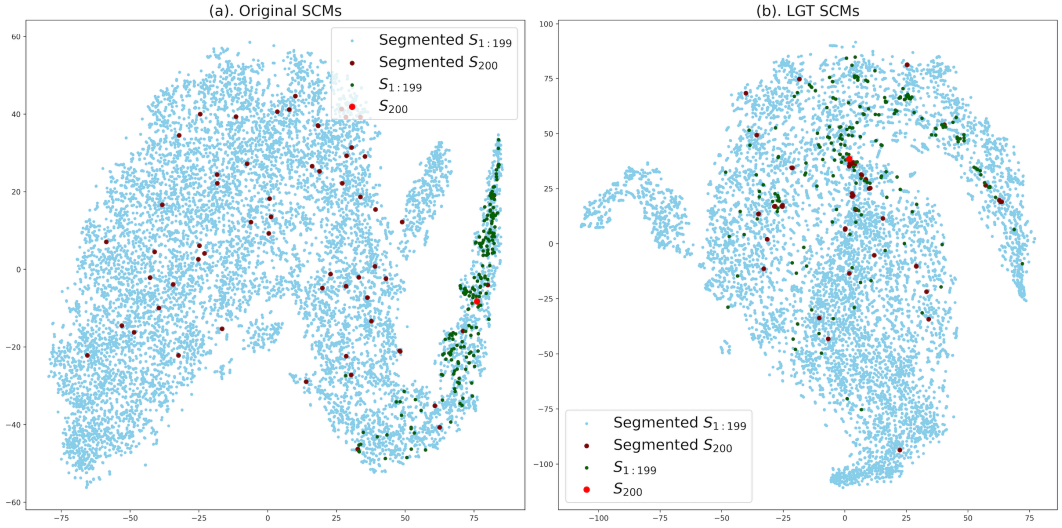
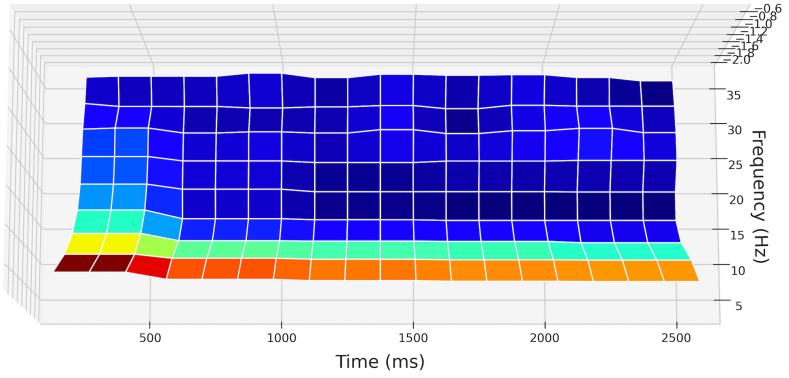


Figure 6.3: Spectrum Distribution Shift: In the KU dataset, a two-dimensional projection of Subject No.1 using t-SNE is displayed. Each session for the subject contains 200 trials in total. In each subfigure, the bright red point represents the 200th trial in that session, while 199 green points depict the first 199 trials. The points have been dimensionally reduced from 20×20-dimensional spatial covariance matrices through t-SNE. The 60 dark red points symbolize the spatial covariance matrices derived from 60 EEG segments, as specified in the segmentation plan in Table 6.1, of the last trial (represented by the bright red point), while the 11940 blue points ($11940 = 199 \times 60$) represent the spatial covariance matrices derived from the remaining 199 trials.

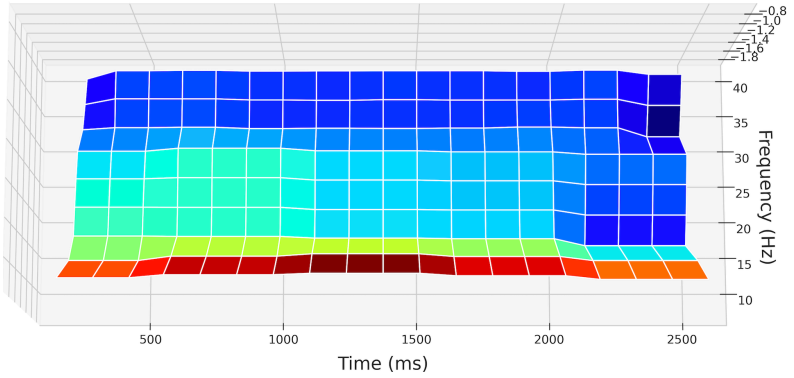
In this section, we visually examine the extent of these changes by plotting a two-dimensional projection of the transformed spatial covariance matrices using the LGT method. This allows us to observe the shifts in their spectrum distributions.

In Figure 6.3 (a), the spatial covariance matrices of the 200^{th} trial (represented by the bright red point) are located at the center of the cluster formed by the spatial covariance matrices of the first 199 trials (depicted in green). However, the segmented spatial covariance matrices of the 200^{th} trial (shown in dark red) are relatively uniformly distributed within the space occupied by the segmented spatial covariance matrices of the first 199 trials (illustrated in blue).

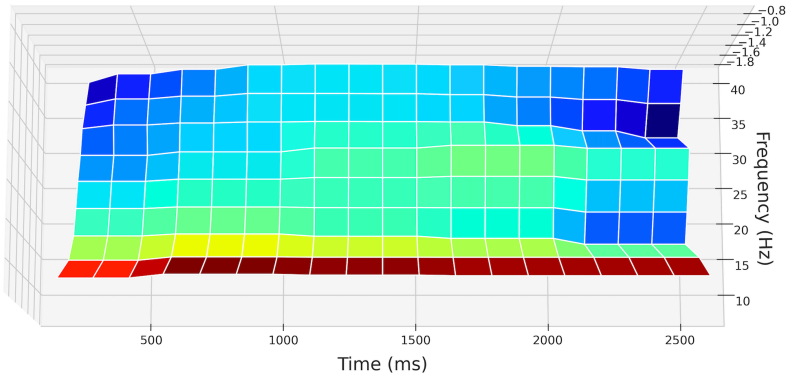
In contrast, Figure 6.3 (b) demonstrates significant changes in the distributions of the spatial covariance matrices' spectrum. The LGT method averages each point with its neighboring points based on the graph topology. As a result, the centers of each group of points are closer together compared to the centers depicted in Figure 6.3 (a).



(a) Discrete spectrogram



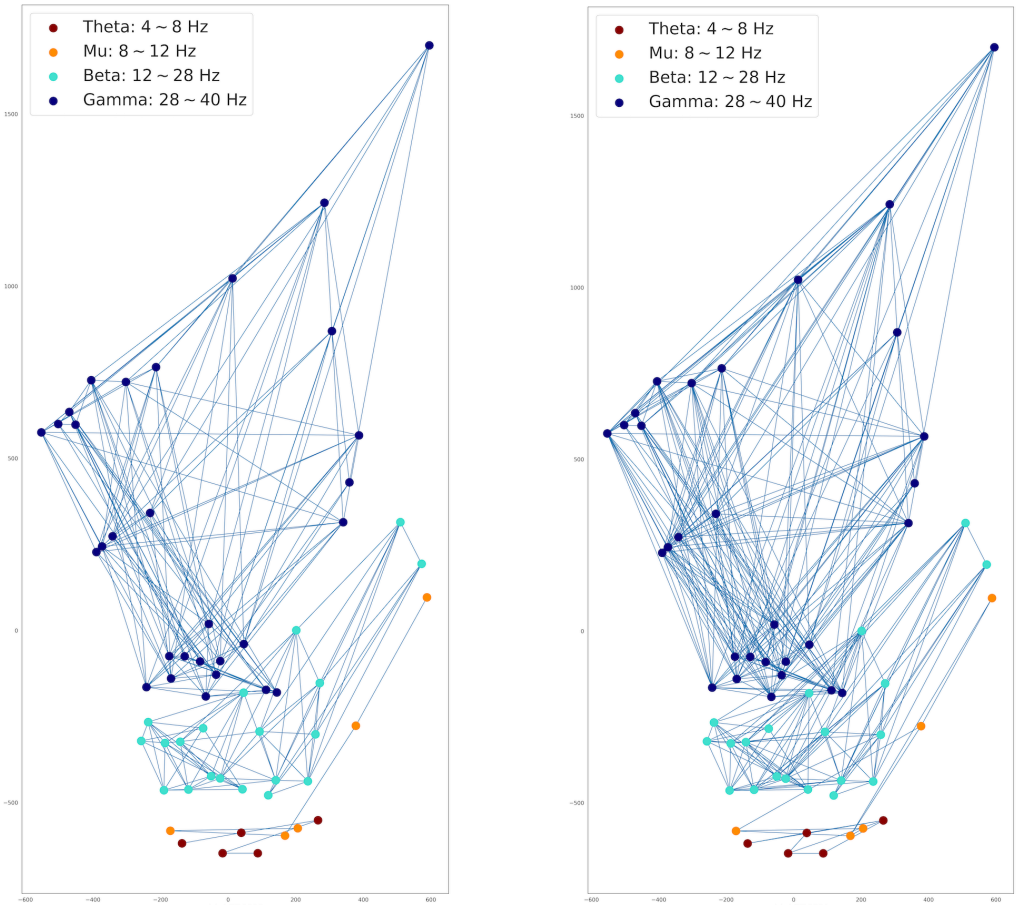
(b) Time Direction (1,1,1,2).



(c) Time Direction (2,2,2,4).

Figure 6.4: Discrete Spectrograms of Variant Configuration Time-Frequency Graphs: The LGT method has four *time direction* numbers representing the forward steps in the components of θ, μ, β, and γ. The *frequency direction* numbers are always set to Time Direction (1, 1, 4, 3).

The discrete spectrograms, ranging from (a) to (c), are calculated by evaluating the lower frequency band's spectrum power on the grids ($4 \sim 16$ Hz) across every five grids on the time axis, with a grid width of 500 ms and a height of 4 Hz. The higher frequency band ($16 \sim 40$ Hz) is calculated across each grid, with a grid width of 250 ms and a height of 4 Hz. Spectrogram (a) represents the original spectrum distribution of Subject No. 1 in the KU dataset, while spectrograms (b) and (c) are the spectrum distributions after the LGT method on (a).



(a) Time Direction (1,1,1,2).

(b) Time Direction (2,2,2,4).

Figure 6.5: Variant Configurations of Time-frequency Graphs: The time-frequency graphs (a) and (b), which are derived from spectrograms (b) and (c), respectively, in Figure 6.4, contain 60 nodes and 390 edges. Each node represents a grid or a couple of grids, with low time resolution for the low frequency (4 to 16 Hz) and high time resolution for the high frequency (16 to 40 Hz). The spectrum of a node evaluates adjacent nodes along the time axis and consists of four graph components corresponding to four frequency bands, i.e., θ, μ, β, and γ. The edge weight of each two adjacent nodes is the geodesic distance between the two points on $(\mathcal{S}_{++}, g^{AIRM})$ and is reconstructed using the multidimensional scaling algorithm.

Variant Configurations of Time-frequency Graphs

We will explore the spectrum distribution and resulting time-frequency graphs generated by different parameter configurations in the LGT method.

Figure 6.4 illustrates that the spectrum distribution, transformed by the LGT method, exhibits a localized forward diffusion from left to right along the forward time direction. This diffusion occurs independently in the θ, μ, β, and γ components due to the separation of these frequency bands in the generation Algorithm 2.

Additionally, Figure 6.5 presents the corresponding time-frequency graphs. Nodes belonging to different frequency bands are not connected. When selecting a larger scale for the forward time flow, the number of edges in the time-frequency graph will increase according to the construction rule. For instance, the time-frequency graph in Figure 6.5 (b) exhibits more connected edges compared to the one in Figure 6.5 (a).

Pseudocodes for Adjacent Matrix Generation

This section provides the pseudocode to generate the adjacency matrix A of the time-frequency graph. A lattice vector is used to compute the average EEG spectral covariance matrices within a specified time and frequency interval in the training set.

Taking the example of the BNCI2014001 dataset, with a 1000 ms signal and non-overlapping segmentations of 250 ms and 125 ms, we obtain 4 and 8 windows, respectively, as shown in Table 6.1. As a result, there are 48 EEG segments (= 4 time windows $\times$ 6 frequency bands + 8 time windows $\times$ 3 frequency bands), representing the nodes in the time-frequency graph. The lattice vector is a tensor with Dimension $= (48, 22, 22)$, where 22 is the number of electrodes. The 48 averaged spatial covariance matrices correspond to the second and third dimensions of the tensors $(i, 22, 22)$, where $1 \leq i \leq 48$.

The Riemannian distances in Algorithm 2 are calculated using the lattice vector. Specifically, the adjacency between two nodes is determined by a time-evolution mechanism. As shown in Table 6.1, the frequency bands are grouped into the θ, μ, β, and γ components, with 4, 4, 16, and 24 nodes, respectively.

The generation algorithm considers the time and frequency directions in the θ, μ, β, and γ components. For example, given Time Direction $(1, 1, 1, 2)$ and Frequency Direction $(1, 1, 4, 3)$, spatial covariance matrices evolve by one step, one step, one

step, and two steps on the θ, μ, β, and γ components, respectively, along the forward time direction, and one step, one step, four steps, and three steps along the frequency direction.

Algorithm 2: Adjacency Matrix Generation

Input : Number of Nodes $(N_\theta, N_\mu, N_\beta, N_\gamma)$;
Length of Time Horizon $(w_\theta, w_\mu, w_\beta, w_\gamma)$;
Time Direction $(x_\theta, x_\mu, x_\beta, x_\gamma)$;
Frequency Direction $(y_\theta, y_\mu, y_\beta, y_\gamma)$.
Output: Adjacency Maxtrix $A[i^*, j^*]$.

Initialization of the lattice vector and start node s $\leftarrow$ 1;

for $(N, w, x, y) \in [\theta, \mu, \beta, \gamma]$ **do**

  for $i \leftarrow \{1, ..., N\}$ **do**
    `/* Initialize the first coordinate for` A. `*/`
    $i^* \leftarrow s + i$;

    `/* Compute the largest forward time step. */`
    $T \leftarrow \min\{w - i\%w - 1, x\}$;

    `/* The time direction. */`
    if $j \in \{i + 1, ..., i + T + 1\}$ **then**
      $j^* \leftarrow s + j$;
      $A[i^*, j^*] \leftarrow \exp\Big(-d^2_{g^{AIRM}}(S_{i^*}, S_{j^*})/t\Big)$;

    `/* The frequency direction. */`
    if $j \in \{i + wy, ..., i + wy + T + 1\}$ **then**
      $j^* \leftarrow s + j$;
      $A[i^*, j^*] \leftarrow \exp\Big(-d^2_{g^{AIRM}}(S_{i^*}, S_{j^*})/t\Big)$;

  $A[j^*, i^*] \leftarrow A[i^*, j^*]$;
  s $\leftarrow$ s + N.

In the pseudocode, $(N, w, x, y) \in [\theta, \mu, \beta, \gamma]$ denotes the selection of a specific combination from the four frequency bands $(\theta, \mu, \beta, \gamma)$. For instance, selecting θ would represent the combination as $(N_\theta, w_\theta, x_\theta, y_\theta)$, focusing on the θ frequency band. To enhance clarity, we have omitted the subscripts in the presentation. The term Number of Nodes $(N_\theta, N_\mu, N_\beta, N_\gamma)$ specifies the count of graph nodes associated with each frequency component. Length of Time Horizon $(w_\theta, w_\mu, w_\beta, w_\gamma)$ refers

to the number of time windows within each frequency band, which vary due to different time resolutions leading to a diverse range of window counts. The inclusion of the maximum forward steps, indicated by T, is crucial, particularly when the distance to the rightmost position (the furthest time step) is shorter than one forward step. The operation $i\%w$ calculates the remainder of i divided by w. When assigning values along the time and frequency axes, the typical procedure involves fixing the frequency first and then progressing along the positive time direction. Within the same frequency band, higher frequencies are subsequently considered.

For detailed code examples, please refer to the related file in the GitHub repository, i.e., `https://github.com/GeometricBCI/Tensor-CSPNet-and-Graph-CSPNet`.

Architecture of Graph-CSPNet

Table 6.3 summarizes the layers and learnable network parameters in Graph-CSPNet. The total number of learnable network parameters is $No_2(o_1 + o_3) + (cN + 1)o_3^2$. For instance, in the case of BNCI2014001, the network configuration is given by $N = 48$, $c = 4$, $o_1 = 22, o_2 = 36$, and $o_3 = 22$, where o_1 and o_2 are the input and output dimensions for the first graph BiMap layer respectively, o_2 and o_3 are the input and output dimensions for the second graph BiMap layer. N is the number of the time-frequency graph nodes, and c is the number of classes. The total number of learnable network parameters is 169,444 parameters. Compared to Tensor-CSPNet, this amount is six times the parameters in 1-CSPNet$^{(9,1,1)}$ (27,104 parameters) and almost two-thirds of the parameters in 5-CSPNet$^{(9,3,1)}$ (232,360 parameters).

Table 6.3: Parameters in a two-layer Graph-CSPNet with input tensor shape (N, o_1, o_1).

Layer	Type of Parameters	Shape of Outputs	Number of Parameters
Graph BiMap Layer (1st)	Stiefel manifolds	(N, o_1, o_2)	No_1o_2
Graph BiMap Layer (2nd)	Stiefel manifolds	(N, o_2, o_3)	No_2o_3
Riemannian Batch Normalization	SPD manifolds	(N, o_3, o_3)	o_3^2
LOG	/	(N, o_3, o_3)	/
Linear	Euclidean	c	cNo_3^2
Total Number of Parameters	/	/	$N(o_1 + o_3)o_2 + (cN + 1)o_3^2$

Table 6.4: Comparison between Tensor-CSPNet and Graph-CSPNet.

Geometric Methods	Tensor-CSPNet	Graph-CSPNet
Network Input	Tensorized Spatial Covariance Matrices.	Time-Frequency Graph.
Architecture	BiMaps; Convolutional Neural Networks.	Graph-BiMaps.
Distinctive Structure	Convolutional Neural Networks for Temporal Dynamics.	Spectral Clustering for Time-Frequency Distributions.
Training Optimizer	Riemannian Optimization	Riemannian Optimization.
Underlying Space	$(\mathcal{S}_{++}, g^{AIRM})$.	$(\mathcal{S}_{++}, g^{AIRM})$.
Methodology Heritage	Common Spatial Pattern.	Common Spatial Pattern; Riemannian-Based Approach.
Design Principle	The Time-Space-Frequency Principle: *Exploitation in the time, space, and frequency domains sequentially.*	The Time-Space-Frequency Principle and the Principle of Time-Frequency Analysis: *Exploitation in the time-frequency domain simultaneously, and* then in the space domain.

Tensor-CSPNet versus Graph-CSPNet

Both Tensor-CSPNet and Graph-CSPNet utilize the time-space-frequency principle to extract discriminative information from EEG spatial covariance matrices. Graph-CSPNet further integrates time-frequency analysis to enhance its analytical capabilities. Technically, Tensor-CSPNet employs convolutional neural networks following the BiMap layers to capture temporal dynamics, while Graph-CSPNet uses the Graph-BiMap layer to concurrently capture time-frequency relationships.

In particular, Graph-BiMap layers incorporate the technique of spectral clustering into the network architecture. Spectral clustering is a multivariate statistical clustering method that leverages the spectrum of a data similarity matrix [152]. As outlined in Section 6.1, the time-frequency graph is constructed by combining classical ϵ-neighborhood methods from spectral clustering with considerations of various frequency components in EEG signals. This approach aggregates spectral power across the time-frequency domain, highlighting significant regions in EEG signals for more effective classification. Consequently, while Tensor-CSPNet focuses on capturing temporal dynamics through convolutional neural networks, the spectral clustering integrated into the Graph-BiMap layer offers a more nuanced and flexible representation of localized fluctuations. For a comparison of these two models, please refer to Table 6.4.

Spectral Clustering and Laplacian

In image processing, the Laplacian operator calculates the second-order derivative of an image, effectively highlighting areas of rapid intensity changes or edges [153]. Similarly, in the time-frequency graph, spectral clustering can be interpreted in a comparable way. By using an overlapping segmentation approach, we can gather more detailed local information compared to a non-overlapping approach. As the number of segments increases, spectral clustering emphasizes areas of significant intensity changes on the SPD manifold, which is constructed based on statistical measures in the time-frequency domain.

On SPD manifolds, given an infinite segmentation plan where spatial covariance matrices $\{S_i\}_{i=1}^{N} \in \mathcal{S}_{++}^{n_C}$ of the time-frequency graph around a test trial $\bar{S}$ are uniformly Gaussian distributed [154], the discrete graph Laplacian $L_M = I - A$ of the

time-frequency graph on a networks-based function f will converge to the continuous Laplacian $\Delta_M(f)$ with a bias term, as established by various studies [155–157], as follows:

$$\frac{1}{\epsilon}\sum_{j=1}^{N} L_M f(S_i) = \frac{1}{2}\Delta_M f(\bar{S}) + \mathcal{O}(\epsilon^{1/2}).$$

Chapter 7

Optimal Transport-Domain Adaptation on SPD Manifolds

In cross-domain problems, the distribution of source data often differs from that of target data, leading to what is widely recognized as a domain adaptation problem in the machine learning community. This issue has gained considerable attention over the past decade [158–160]. For example, signals in an online testing set frequently differ from those in the calibration set despite being generated by the same process, highlighting a common domain adaptation challenge.

In domain adaptation, a domain $\mathcal{D}$ consists of a feature space X and a marginal probability distribution $\mathbb{P}(X)$, denoted as $\mathcal{D} = \{X, \mathbb{P}(X)\}$. Meanwhile, a task $\mathcal{T}$ comprises a label space Y and a conditional distribution $\mathbb{Q}(Y|X)$, represented by $\mathcal{T} = \{Y, \mathbb{Q}(Y|X)\}$. Domain adaptation aims to build a classifier in the target domain $\mathcal{D}_T$ by leveraging the information obtained from the source domain $\mathcal{D}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ and $\mathcal{T}_S \neq \mathcal{T}_T$. In particular, $\mathcal{D}_S \neq \mathcal{D}_T$ yields either $X_S \neq X_T$ or $\mathbb{P}_S(X_S) \neq \mathbb{P}_T(X_T)$, whereas $\mathcal{T}_S \neq \mathcal{T}_T$ results in either $Y_S \neq Y_T$ or $\mathbb{Q}_S(Y_S|X_S)$ or $\mathbb{Q}_T(Y_T|X_T)$.

The domain adaptation problem can be broadly categorized into unsupervised and semi-supervised scenarios. In the unsupervised scenario, no labels are available in the target domain, while in the semi-supervised scenario, a limited number of labeled instances are available in the target domain to guide the adaptation process. For instance, the variability of EEG signals often results in unreliable and non-robust performance when transitioning from calibration to feedback phases, as

these sessions exhibit distinct distributions. This variability in EEG-based classification exemplifies a typical domain adaptation problem [161]. In response to this issue, a covariate shift adaptation approach has been proposed in machine learning society [162]. This approach assumes that the difference between domains is characterized by a change in the feature space while the conditional distributions remain unchanged.

Among various methods that have been proposed to address the domain adaptation problem across numerous engineering scenarios, Optimal Transport (OT) has emerged as a novel and effective solution for aligning source and target domains [133, 134]. The OT approach introduces regularized optimal transport, allowing a classifier to learn from labeled source data and apply this knowledge to target data. The goal of the Optimal Transport-Domain Adaptation (OT-DA) framework is to minimize the overall transportation costs between two distributions, quantified by the Wasserstein metric. It seeks to find a push-forward transformation $\varphi : X_S \longmapsto X_T$ that satisfies $\mathbb{P}_S(X_S) = \mathbb{P}_T(\varphi(X_S))$. The strengths of the OT-DA framework lie in its simplified evaluation of empirical data distribution and its enhanced ability to leverage the underlying geometry of discrete samples [133].

While broad in scope, the OT-DA framework can benefit from enhancements derived from insights into specific applications. For instance, in EEG-based motor imagery classification, EEG signals are processed as (average) spatial covariance matrices after segmentation in time and averaging over trials [23, 163]. This leads to the cross-session scenario of motor imagery classification taking place in the space of covariance matrices, which forms an SPD manifold if endowed with a Riemannian metric. In this context, Yair et al. utilized squared ℓ_2 distance as the cost function $c(x, \bar{x}) = ||x - \bar{x}||^2_{\ell_2}$ in the OT-DA framework, deriving a closed-form solution through Brenier's polar factorization [164].

Although Yair's work pioneered the OT-DA framework on SPD manifolds, it presents certain theoretical and practical shortcomings. Theoretically, their chosen cost function, based on Euclidean distance, undermines the consistency of mathematical formulation in the view of differential geometry. Practically, their methodology fails to account for disparities in conditional distributions. Additionally, the unique challenges inherent to the problem, such as the scarcity of labeled data during the feedback phase and the divergence of spatial covariance matrices from distinct frequency bands, pose significant modeling difficulties. These challenges

necessitate the development of a more adaptable OT-DA framework to effectively address problems on SPD manifolds.

To address these challenges, we consider a more applicable scenario where $X_S = X_T$, $Y_S = Y_T$, $\mathbb{P}_S(X_S) \neq \mathbb{P}_T(X_T)$, and $\mathbb{Q}_S(Y_S|X_S) \neq \mathbb{Q}_T(Y_T|X_T)$. This situation is typically handled through joint distribution adaptation, which concurrently aligns both marginal and conditional distributions across domains, even when dealing with implicit probability distributions, without relying on labeled information from the target domain [165]. To facilitate modeling of $X_S = X_T$, we formulate the space of SPD matrices using the Log-Euclidean metric g^{LEM} on SPD manifolds, as discussed in Chapter 3. In this context, parallel transport on SPD manifolds with the Log-Euclidean metric simplifies to an identity map, ensuring that the feature space of the source and target domains, which is the tangent space at the identity of SPD manifolds, are the same.

In this chapter, we propose a novel geometric deep learning model called Deep Optimal Transport (DOT) to address the joint distribution adaptation problem on SPD manifolds, utilizing a squared Log-Euclidean distance as the cost function in optimal transport, i.e., $c(x, \bar{x}) = d^2_{g^{LEM}}(x, \bar{x})$. DOT is predicated on the assumption that information transfer between domains can be effectively estimated through optimal transport. The goal of DOT is to find a transformation on SPD manifolds $\varphi : X_S, X_T \longmapsto Z \subset \mathcal{S}_{++}$ such that $\mathbb{P}_S(\varphi(X_S)) = \mathbb{P}_T(\varphi(X_T))$ and $\mathbb{Q}_S(Y_S|\varphi(X_S)) = \mathbb{Q}_T(Y_T|\varphi(X_T))$. To evaluate the effectiveness of DOT, we focus on cross-session scenarios in motor imagery classification. This problem is notably challenging and often inadequately addressed, primarily due to the extensive patterns of synchronized neuronal activity that continuously shift over time. This results in significant variability in brain signals captured by EEG devices across different sessions. In our evaluation, we calibrate the model in one session and then test it in another, encompassing both unsupervised and semi-supervised domain adaptation scenarios.

Table 7.1: History of the OT-DA Framework.

Framework	Cost function	Underlying Space	Transformation	Domain Adaptation Scenario
OT-DA[133, 134]	$c(x, \bar{x}) = \|\|x - \bar{x}\|\|^2_{\ell_2}$.	Euclidean	Affine	$X_S \neq X_T$.
JDOT-DA[128, 129]	$c(x, \bar{x}) = \|\|x - \bar{x}\|\|^2_{\ell_2}$	Euclidean	Affine	$X_S \neq X_T$, $\mathbb{Q}_S(Y_S\|X_S) \neq \mathbb{Q}_T(Y_T\|X_T)$.
OT-DA on SPD manifolds[164]	$c(x, \bar{x}) = \|\|x - \bar{x}\|\|^2_{\ell_2}$	$(\mathcal{S}_{++}, g^{AIRM})$	Bi-Map	$X_S \neq X_T$.
Proposed	$c(x, \bar{x}) = d^2_{g^{LEM}}(x, \bar{x})$	$(\mathcal{S}_{++}, g^{LEM})$	Neural Networks	$X_S = X_T$, $\mathbb{P}_S(X_S) \neq \mathbb{P}_T(X_T)$, and $\mathbb{Q}_S(Y_S\|X_S) \neq \mathbb{Q}_T(Y_T\|X_T)$.

7.1 Methodology

In the methodology section, we will first generalize the OT-DA framework on ($\mathcal{S}_{++}$, g^{LEM}) and propose a novel geometric deep learning approach to address the problems formulated on this framework.

7.1.1 Optimal Transport-Domain Adaptation

We first calculate the gradient of any squared distance function on general Riemannian manifolds.

Lemma 7.1. *Let $(\mathcal{M}, g)$ be a connected, compact, and C^3-smooth Riemannian manifold without a boundary. Consider a compact subset $\mathcal{U} \subset \mathcal{M}$ and a fixed point $q \in \mathcal{U}$. Let $f(p) = \frac{1}{2} d_g^2(p, q)$ denote the squared distance function. Then, for p not on the boundary of $\mathcal{M}$, the gradient of $f(p)$ is given by $\nabla f(p) = -\log_p(q)$.*

Proof. Consider a point $p \in \mathcal{M}^\circ$ without loss of generality. There exists an open neighborhood $\mathcal{N}$ of p and $\epsilon > 0$ such that the exponential map $\exp_p$ maps the ball $\mathcal{B}(0, \epsilon) \subset T_p\mathcal{M}$ diffeomorphically onto $\mathcal{N}$ [76, Corollary 5.5.2.]. Let $c(s)$ be a geodesic on $\mathcal{N}$, parameterized with constant speed, with $c(0) := q \in \mathcal{N}$ and $c(1) := p$ with $\dot{c}(0) = \log_q(p)$. By reversing the direction of the geodesic $c(1-s)$, we have $\dot{c}(1) = -\log_p(q)$. We proceed by linearizing the exponential maps around $0 \in T_p\mathcal{M}$ and $\dot{c}(0) \in T_q\mathcal{M}$ and derive the derivative of a function f as follows:

$$
\begin{aligned}
& f\big(\exp_p(v)\big) \\
=& \frac{1}{2} d_g^2\big(\exp_p(v), q\big) \\
=& \frac{1}{2} \big| \log_q\big(\exp_p(v)\big) \big|_q^2 \\
=& \frac{1}{2} \big| \dot{c}(0) + (D\log_q)_{\dot{c}(0)}\big((D\exp_p)_0 v\big) + o(|v|_p) \big|_q^2 \\
=& \frac{1}{2} \big| \dot{c}(0) \big|_q^2 + g_q(\dot{c}(0), (D\log_q)_{\dot{c}(0)} v) + o(|v|_p) \\
=& \frac{1}{2} d_g^2(p, q) + g_p(\dot{c}(1), v) + o(|v|_p),
\end{aligned}
$$

where the differential $(D\exp_p)_q(\,\cdot\,) : T_q(T_p\mathcal{M}) \longmapsto (T_p\mathcal{M})$ is the differential of the exponential map at q with respect to p. The fourth equality follows from

$(D\exp_p)_0 = I$ on $T_p\mathcal{M}$, and the fifth equality follows from Gauss' lemma [166, Section 3.3.5.]. Hence, by the definition of the gradient on Riemannian manifolds, we have $df(p)(v) = g_p\big(\nabla f(x), v\big)$, which implies that $\nabla f(p) = \dot{c}(1) = -\log_p(q)$. □

We say a function $\psi : \mathcal{M} \longmapsto \mathbb{R} \cup \{-\infty\}$ is *c-concave* if it is not identically $-\infty$ and there exists $\varphi : \mathcal{M} \longmapsto \mathbb{R} \cup \{\pm\infty\}$ such that

$$\psi(p) = \inf_{q\in\mathcal{M}} \big\{c(p,q) + \varphi(q)\big\},$$

where $c(p,q)$ is a nonnegative cost function that measures the cost of transporting mass from q to $p \in \mathcal{M}$. Brenier's polar factorization on general Riemannian manifolds is given as follows:

Lemma 7.2 (Brenier's Polar Factorization [167])**.** *Given any compactly supported measures $\mu, \nu \in \mathbb{P}(\mathcal{M})$. If μ is absolutely continuous with respect to Riemannian volume, there exists a unique optimal transport map $T \in S(\mu,\nu)$ as follows,*

$$T(x) = \exp_p\big(-\nabla\psi(p)\big),$$

where $\psi : \mathcal{M} \longmapsto \mathbb{R}\cup\{\pm\infty\}$ is a c-concave function with respect to the cost function $c(p,q) = \frac{1}{2}d_g^2(p,q)$, and $\exp_p(\,\cdot\,)$ is the exponential map at p on $(\mathcal{M}, g)$.

On $(\mathcal{S}_{++}, g^{LEM})$, suppose source measure μ and target measure $\nu \in \mathbb{P}(\mathcal{S}_{++})$ that are accessible only through finite samples $\{S_i\}_{i=1}^N \sim \mu$ and $\{\bar{S}_j\}_{j=1}^M \sim \nu$. Each sample contributes to its respective empirical distribution with weights $\frac{1}{N}$ and $\frac{1}{M}$. We assume there exists a linear transformation on $(\mathcal{S}_{++}, g^{LEM})$ between two discrete feature space distributions of the source and target domains that can be estimated with the discrete Monge-Kantorovich problem 4.3, i.e., there exists

$$\gamma^\dagger := \arg\min_{\gamma\in\prod(\mu,\nu)} \big\langle \gamma, d^2_{g^{LEM}}(S,\bar{S})\big\rangle_{\mathcal{F}}, \tag{7.1}$$

where $\prod(\mu,\nu) := \{\gamma \in \mathbb{R}^{N\times M}_{\geq 0} \big| \gamma\mathbb{1}_N = \frac{1}{N}$, and $\gamma^\top\mathbb{1}_M = \frac{1}{M}\}$. The pseudocode for the computation of the transport plan is presented in Algorithm 3.

In the following paragraph, we will employ discrete transport plans to construct c-concave functions and utilize Lemma 7.1 and 7.2 to obtain continuous transport plans. According to Algorithm 3, the new coordinates for target samples on SPD

Algorithm 3: Discrete Monge-Kantorovich Algorithm on $(\mathcal{S}_{++}, g^{LEM})$.

Input : Source samples $\{S_i\}_{i=1}^N \sim \mu$, target samples $\{\bar{S}_j\}_{j=1}^M \sim \nu$.

Output: Transport plan γ, and new coordinates for target samples $\{\bar{S}_j^\dagger\}_{j=1}^M$.

`/* 1. Compute ground metric matrix for OT */`

$D(i,j) \leftarrow d_{g^{LEM}}(S_i, \bar{S}_j)$.

`/* 2. Compute transport plan` γ `using the earth movers distance-based OT algorithm. */`

$\gamma^\dagger \leftarrow OT^{EMD}(N, M, D(i,j))$.

`/* 3. Compute new coordinates using transport plan on` $(\mathcal{S}_{++}, g^{LEM})$. `*/`

$\{\bar{S}_j^\dagger\}_{j=1}^M \leftarrow \exp\Big\{\big(\gamma^\dagger[j,:]\{\log(S_i)\}_{i=1}^N\big)\Big\}$.

manifolds are given by the following expression:

$$\{\bar{S}_j^\dagger\}_{j=1}^M := \exp\Big\{\big(\gamma^\dagger[j,:]\{\log(S_i)\}_{i=1}^N\big)\Big\}.$$

We define $\varphi(\bar{S}) := +\infty$ for $\bar{S} \notin \{\bar{S}_j^\dagger\}_{j=1}^M$, and set the cost function $c(S,\bar{S}) := \frac{1}{2} d^2_{g^{LEM}}(S,\bar{S})$. This leads to the formulation of the corresponding c-concave function as follows:

$$\psi(S) = \min_{j\in\{1,\dots,M\}} \frac{1}{2} d^2_{g^{LEM}}(S, \bar{S}_j^\dagger) + \varphi(\bar{S}_j^\dagger).$$

For a fixed S, we suppose $j^* \in \{1,\dots,M\}$ achieves the minimum. By invoking Lemma 7.1 and 7.2, we promptly derive the unique optimal transport map as follows,

$$T(S) = \exp_S\big(-\nabla_S \psi(S)\big) = \bar{S}_{j^*}^\dagger \in \{\bar{S}_j^\dagger\}_{j=1}^M,$$

which implies the following theorem:

Theorem 7.3. *On $(\mathcal{S}_{++}, g^{LEM})$, consider the source measure μ and target measure $\nu \in \mathbb{P}(\mathcal{S}_{++})$, which are only accessible through finite samples $\{S_i\}_{i=1}^N \sim \mu$ and $\{\bar{S}_j\}_{j=1}^M \sim \nu$. Each sample contributes to the empirical distribution with weights $\frac{1}{N}$ and $\frac{1}{M}$, respectively. Assuming the existence of a linear transformation between the feature space distributions of the source and target domains, which can be estimated using the discrete Monge-Kantorovich Problem 7.1. Therefore, the values of the continuous optimal transport $T(S)$ lie in the set $\Big\{\exp\Big\{\big(\gamma^\dagger[j,:]\log(S_i)_{i=1}^N\big)\Big\}\Big\}_{j=1}^M$.*

Theorem 7.3 extends [133, Theorem 3.1] to $(\mathcal{S}_{++}, g^{LEM})$ in the OT-DA framework. Notably, our framework is designed to incorporate any supervised neural network-based transformation, distinguishing it from the conventional use of affine

transformations, and it employs the squared Riemannian distance. In the discussion part, we clarify the connection between our proposed approach and another similar OT-DA framework on SPD manifolds proposed in [164], which utilizes the squared ℓ_2-Euclidean distance in its cost function.

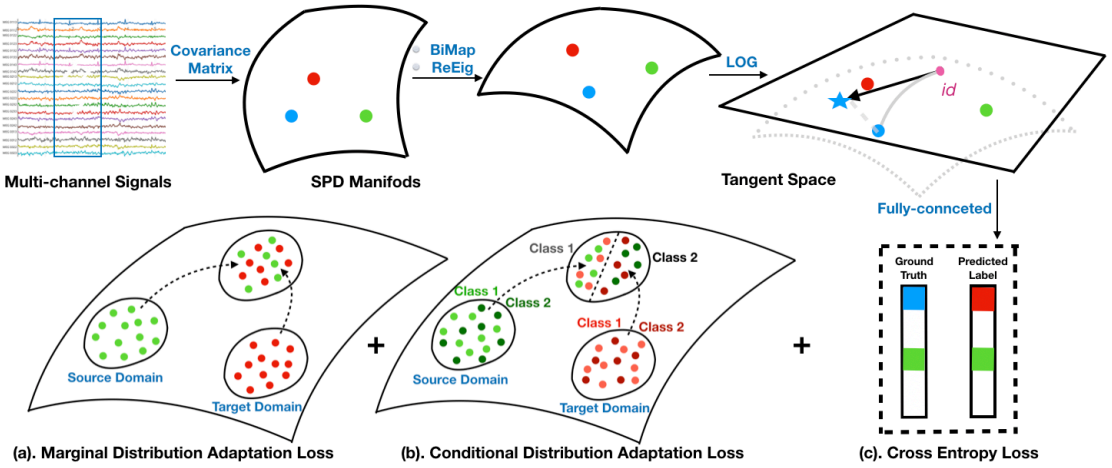


Figure 7.1: Illustration of Deep Optimal Transport: Multi-channel signals are first transformed into covariance matrices, placing these matrices on $(\mathcal{S}_{++}, g^{LEM})$. The SPD matrix-valued data is then processed through the BiMap, ReEig, and LOG layers, mapping it to the tangent space at the identity. The loss function incorporates cross-entropy loss, marginal distribution adaptation, and conditional distribution adaptation.

7.1.2 Deep Optimal Transport

We propose a novel geometric deep learning model, Deep Optimal Transport (DOT) on $(\mathcal{S}_{++}, g^{LEM})$, to address the joint distribution adaptation problems involving both marginal distribution shifts and conditional distribution shifts. We assume that samples from the source and target distributions can be aligned via optimal transport The proposed architecture is depicted in Figure 7.1.

Marginal Distribution Adaptation (MDA)

MDA aims to adapt the marginal distributions $\mathbb{P}_S(X_S)$ and $\mathbb{P}_T(X_T)$ on $(\mathcal{S}_{++}, g^{LEM})$. We measure the statistical discrepancy between them using the Riemannian distance $d_{g^{LEM}}(\overline{w}(\mathcal{B}_S), \overline{w}(\mathcal{B}_T))$ between the Log-Euclidean Fréchet means of each batch of source and target samples $\mathcal{B}_S$ and $\mathcal{B}_T$, respectively. The MDA loss is formulated as $\mathcal{L}_{MDA} := ||\log\left(\overline{w}(\mathcal{B}_S)\right) - \log\left(\overline{w}(\mathcal{B}_T)\right)||_{\mathcal{F}}$.

Conditional Distribution Adaptation (CDA)

CDA aims to address the adaptation of conditional distributions $\mathbb{Q}_S(Y_S|X_S)$ and $\mathbb{Q}_T(Y_T|X_T)$ on $(\mathcal{S}_{++},\ g^{LEM})$. It necessitates pseudo labels predicted by a baseline algorithm for the target domain [165]. We consider class-conditional distribution $\mathbb{Q}(X|Y)$ instead of posterior probability $\mathbb{Q}(Y|X)$ and introduce the use of pseudo labels $\hat{y}$ for the unsupervised setting. To quantify the statistical difference between $\mathbb{Q}_S(Y_S|X_S)$ and $\mathbb{Q}_T(Y_T|X_T)$, we continue to employ the Riemannian distance $d_{g^{LEM}}(\overline{w}(\mathcal{B}_{S^\ell}), \overline{w}(\mathcal{B}_{T^\ell}))$ between the Log-Euclidean Fréchet means of each batch of source and target samples $S^\ell := \{x_S|\hat{y}_S = \ell\}$ and $T^\ell := \{x_T|\hat{y}_T = \ell\}$, respectively, for class $\ell \in \{1, ..., L\}$. The CDA loss is formulated as $\mathcal{L}_{CDA} := ||\log\left(\overline{w}(\mathcal{B}_{S^\ell})\right) - \log\left(\overline{w}(\mathcal{B}_{T^\ell})\right)||_{\mathcal{F}}$.

Overall Objective Function

The overall objective function of DOT is composed of the cross-entropy loss $\mathcal{L}_{CE}$ and the joint distribution adaptation as follows:

$$\mathcal{L}_{DOT} := \alpha_1\mathcal{L}_{CE} + \alpha_2\mathcal{L}^2_{MDA} + \alpha_3\mathcal{L}^2_{CDA},$$

where α_1, α_2, and $\alpha_3 \geq 0$.

Remark 7.1. 1). $\mathcal{L}_{MDA}$ and $\mathcal{L}_{CDA}$ should be in the squared form, as required by Brenier's formulation and the associated theoretical results [167].

2). The affine invariant Riemannian metric is not a suitable choice for our framework. Because the calculation of the Fréchet mean on $(\mathcal{S}_{++},\ g^{AIRM})$ requires iterative methods. This is not feasible as a loss function for deep learning-based approaches.

3). In practice, we substitute Equation 4.2 into $\mathcal{L}_{MDA}$ and $\mathcal{L}_{CDA}$, resulting in the simplified expressions,

$$\mathcal{L}_{MDA} = ||\sum_{S_i\in\mathcal{B}_S}\frac{\log(S_i)}{|\mathcal{B}_S|} - \sum_{S_j\in\mathcal{B}_T}\frac{\log(S_j)}{|\mathcal{B}_T|}||_{\mathcal{F}};$$

$$\mathcal{L}_{CDA} = ||\sum_{S^\ell_i\in\mathcal{B}_{Sl}}\frac{\log(S^\ell_i)}{|\mathcal{B}_{S^\ell}|} - \sum_{S^\ell_j\in\mathcal{B}_{T^\ell}}\frac{\log(S^\ell_j)}{|\mathcal{B}_{T^\ell}|}||_{\mathcal{F}}.$$

These simplified expressions have two distinct interpretations in domain adaptation as follows: First, they can be seen as an empirical maximum mean discrepancy approach defined as follows:

$$\mathcal{L}_{MMD} := ||\sum_{i=1}^{n} \varphi(S_i)/n - \sum_{j=1}^{m} \varphi(S_i)/m||_{\mathcal{H}},$$

where $n = |\mathcal{B}_S|$, $m = |\mathcal{B}_T|$, $\varphi(x) := \log(x)$ is the feature map, $\mathcal{H} := \mathbb{R}^d$ is the reproducing kernel Hilbert space, and d is the output dimension; Second, since each matrix S in $\mathcal{B}_S$ or $\mathcal{B}_T$ is a covariance matrix of multichannel signals, the expressions can also be viewed as a correlation alignment approach [168, 169]. This approach aligns the second-order statistics of the source and target data distributions to minimize the drift between their statistical distributions.

4). For multi-source adaptation, the loss function can be modified by incorporating multi-joint distribution adaptions between each pair of source and target domains on $(\mathcal{S}_{++}, g^{LEM})$, as follows:

$$\mathcal{L} := \alpha_1^i \mathcal{L}_{CE} + \sum_i \left(\alpha_2^i \mathcal{L}_{MDA,i} + \alpha_3^i \mathcal{L}_{CDA,i}\right),$$

where the subscript i is used to represent the i^{th} source domain. This modification is based on the theory of multi-marginal optimal transport on Riemannian manifolds [170] and multi-source joint distribution optimal transport [171].

5). DOT does not directly solve an OT problem. Instead, it uses transportation costs as a guiding principle to direct the neural network's learning process. Consequently, the trained neural network maps the source and target domains onto a submanifold of the SPD manifold, thereby implementing a subspace method in transfer learning.

6). DOT calculates this cost for the centroid points and addresses the issue of conditional marginal adaptation without requiring alternating parameter updates. In contrast, DeepJDOT in Chapter 4 computes the transportation costs for all points and necessitates alternating parameter updates, as the transport plan is a deterministic transfer.

7.2 Experimental Results

7.2.1 Deep Optimal Transport-Based Classifier

We model the space of EEG spatial covariance matrices in $(\mathcal{S}_{++}, g^{LEM})$ and employ DOT to address the issue of EEG statistical characterization shifting during the feedback phase By using event-related desynchronization/synchronization methods to classify motor imagery classes, specific EEG frequency components within certain bands are identified, providing optimal discrimination between imagined movements. Therefore, it is reasonable to assume that features in the source and target feature spaces remain consistent, i.e., $X_S = X_T$, after transformation into a latent space using the BiMap layer. This assumption justifies modeling the space of EEG spatial covariance matrices using SPD manifolds with the Log-Euclidean metric. Furthermore, the BiMap layer offers a transitive group action on SPD manifolds, enabling any transition between two points on the manifolds. In Fig. 7.2, we illustrate how the DOT-based MI-EEG classifier functions in a domain adaptation problem.

Specifically, EEGs are represented by $X \in \mathbb{R}^{n_C \times n_T}$, where n_C and n_T denote the number of channels and timestamps, respectively. The corresponding EEG spatial covariance matrix, $S := S(\Delta f \times \Delta t) \in \mathcal{S}_{++}^{n_C}$, represents an EEG segment in the frequency band Δf and time interval Δt. Let $\mathcal{F}$ and $\mathcal{T}$ represent the frequency and time domains, respectively. A given segmentation $Seg(\mathcal{F}, \mathcal{T})$ is written as $\{\Delta f \times \Delta t | \Delta f \subset Range(\mathcal{F}),$ and $\Delta t \subset Range(\mathcal{T})\}$, on the time-frequency domain of EEG signals. To ensure the theoretical convergence of this classifier in a real-world context, the following three assumptions are necessary:

Assumption 7.1. There exist linear transformations between the feature space distributions on the source and target domains for each segment of $Seg(\mathcal{F}, \mathcal{T})$, which can be estimated using the Kantorovich Problem 7.1.

Assumption 7.2. The weights for the Log-Euclidean Fréchet mean of samples on each time-frequency segment $\Delta f \times \Delta t \in Seg(\mathcal{F}, \mathcal{T})$ in both the source and target empirical distributions are uniform and equal $1/|Seg(\mathcal{F}, \mathcal{T})|$.

Assumption 7.3. The Riemannian distance between the Log-Euclidean Fréchet means of source and target samples, both under the same time-frequency segment, is minimized, that is:

$$d_{gLEM}\left(\overline{w}(\mathcal{B}_S), \overline{w}(\mathcal{B}_T)\right) \leq d_{gLEM}\left(\overline{w}(\mathcal{B}_S), \overline{w}(\mathcal{B}'_T)\right),$$

where $\mathcal{B}'_T$ is a batch of samples on the target domain that is generated by the other time-frequency segments $\Delta f' \times \Delta t' \in Seg(\mathcal{F}, \mathcal{T})$, where $\Delta f' \neq \Delta f$ or $\Delta t' \neq \Delta t$.

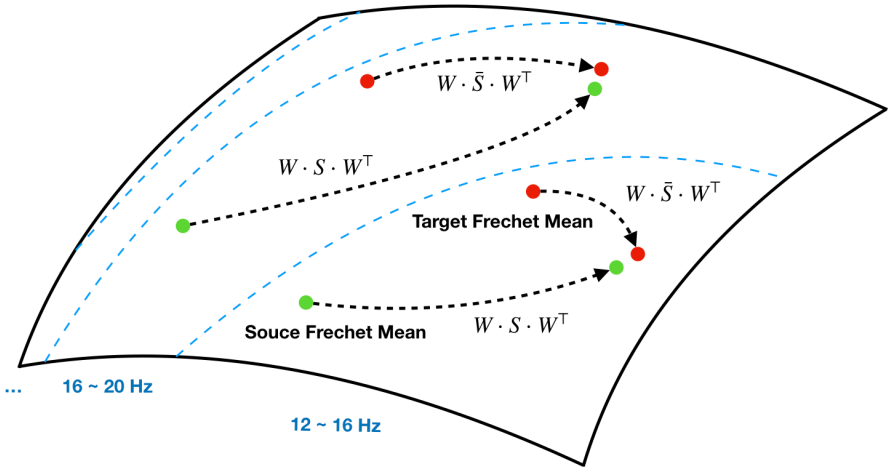


Figure 7.2: Illustration of DOT-Based Motor Imagery Classifier: The proposed classifier simultaneously transports the source and target Fréchet means in each frequency band to a common subspace within the same frequency band. Assumption 7.2 ensures that each frequency component contributes equally to the classification. Assumption 7.3 allows each pair of transformations to occur within the same EEG frequency band.

Assumption 7.1 is a necessary condition for modeling the OT-DA problem. Assumption 7.2 is straightforward because, in cases where we are uncertain about which EEG frequency band holds greater importance for a specific task, we typically assume that each frequency band has equal priority for the classification task. Assumption 7.3 ensures that features extracted from different bands do not overlap.

Corollary 7.4. *Suppose EEG signals in the source and target domain have the same segmentation, Seg($\mathcal{F}, \mathcal{T}$). Assuming we model EEG spatial covariance matrices in ($\mathcal{S}_{++}^{n_C}$, g^{LEM}) and that Assumption 7.1, 7.2, and 7.3 exist, then the values of the continuous optimal transport $T(S)$ reside in the set $\left\{\log(S_i)\right\}_{i=1}^{N}$.*

Proof. In this case, the transport plan γ equals the identity matrix. Consequently, Theorem 7.3 leads directly to this corollary. □

7.2.2 Evaluation Settings

Evaluation Dataset

The selection criteria for the three evaluation datasets, KU, BNCI2014001, and BNCI2015001, require each subject to collect at least two session recordings on different days.

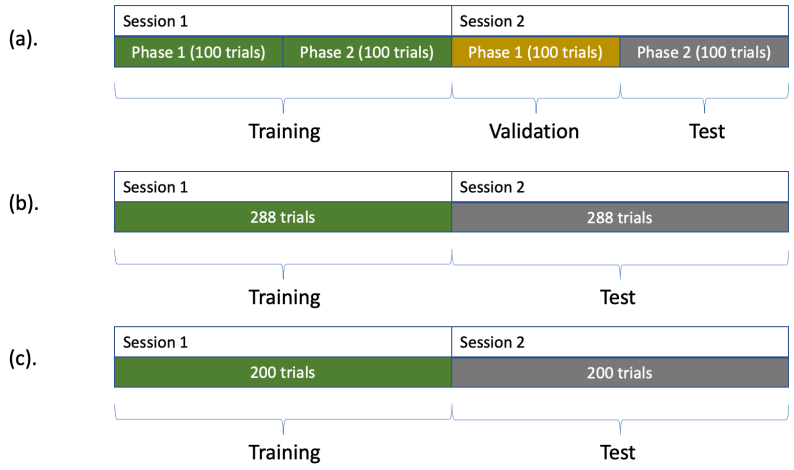


Figure 7.3: Illustrations for experimental settings on three datasets: (a). KU; (b). BNCI2014001; (c). BNCI2015001.

Evaluation Scenarios

The experimental settings for domain adaptation on the three evaluated datasets are described as follows and illustrated in Figure 7.3:

- Semi-supervised Domain Adaptation: For the KU dataset, Session 1 was used for training, and the first half of Session 2 was used for validation, while the second half was reserved for testing. Early stopping was adopted during the validation phase.
- Unsupervised Domain Adaptation: For the BNCI2014001 and BNCI2015001 datasets, Session 1 was used for training, and Session 2 was used for testing. No validation set was used, and a maximum epoch value (e.g., 500) was set for stopping in practice.

We specified that knowledge transfer must always occur from Session 1 (source domain) to Session 2 (target domain), i.e., $S1 \longrightarrow S2$ for each dataset. This is because there is no neurophysiological meaning if negative transfer is used in the learning process. The former and latter sessions typically correspond to the calibration and feedback phases.

Evaluation Baselines

The geometric methods in this study can be broadly divided into four categories: parallel transport, deep parallel transport, optimal transport, and deep optimal transport. Each category encompasses various transfer methods, applied to three types of base machine learning classifiers: support vector machine, minimum distance to Riemannian mean, and SPDNet. For SPDNet, we utilize a single-depth Bi-Map layer with both the input and output dimensions set to the number of electrodes on the primary motor cortex: 20 (KU), 22 (BNCI2014001), and 13 (BNCI2015001).

Since computations on the SPD manifold are tied to Riemannian metrics, we will highlight the specific Riemannian metrics associated with each method. The various transfer methods across categories, according to different Riemannian metrics, are presented in the first three columns of Table 7.2.

Category of Parallel Transport

This category refers to the utilization of parallel transport to perform a transfer of training and test data distributions. One representative method in this category is Riemannian Procrustes Analysis (RPA), as discussed in Chapter 3.

Category of Deep Parallel Transport

This category refers to domain adaptation approaches that utilize deep learning and parallel transport on SPD manifolds. One representative work in this area is Riemannian Batch Normalization (RieBN), an SPD matrix-valued network architecture that employs parallel transport for batch centering and biasing in SPD neural networks, as discussed in Chapter 4.

Category of Optimal Transport

This category involves treating the distribution of the covariance matrix obtained from cross-session data as an OT problem and solving the transport plan using different variants of Wasserstein distances [172]. The classifier is trained in one session and tested in a shifted session, making this an unsupervised method. Three different types of Wasserstein distances are utilized: Earth Mover's Distance (EMD), Sliced-Wasserstein Distance (SPDSW), and Log Sliced-Wasserstein Distance (logSW).

Category of Deep Optimal Transport

This category refers to the approach proposed in this study, which uses SPD neural networks on $(\mathcal{S}_{++}, g^{LEM})$ to transfer both source domain and target domain to a common space induced by optimal transport.

7.2.3 Results of Synthetic Dataset

We synthesize a source domain dataset consisting of 50 covariance matrices of size 2×2, drawing a Gaussian distribution on the SPD manifold proposed in a

generative approach [154] with a median of 1 and a standard deviation of 0.4. We utilize the following transformation matrix in the generation,

$$W := \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix},$$

and data in the target domain can be expressed as $WSW^\top$.

Subfigure (a) shows red points sourced from the source domain and blue points originating from the target domain. In Subfigure (b), the blue points represent the relocation of the red points using the EMD method from OT. The associations made by this transformation are represented by thin blue lines. Notably, the position of these transformed blue points is very close to the points in the target domain in Subfigure (a). This proximity can be attributed to the OT-DA method, which essentially computes a weighted average of the source and target domain data points based on weights learned to minimize transportation costs.

The transformed blue points in Subfigures (c) and (d) were derived from mapping the red points via neural network weights learned through the DOT and DeepJDOT methods, respectively. Corresponding point pairs are connected by thin blue lines. These figures reveal a shared fundamental characteristic of the DOT and DeepJDOT methods: both project the source and target domains onto a subspace learned via the network, which, in these two figures, is represented by a straight line. In differential geometry, this is known as a submanifold, a lower-dimensional manifold embedded in a higher-dimensional one. In this case, the two-dimensional matrices are represented by a line, effectively projecting the higher-dimensional data onto a lower-dimensional space, reducing complexity while preserving essential information.

7.2.4 Results of Real-World EEG Datasets

As shown in Table 7.2, we classify all approaches into three categories according to Riemannian metrics. The first category does not require a Riemannian metric for computation. The second employs g^{AIRM}, while the third uses g^{LEM}.

The experimental results indicate that metrics strongly influence the results. Specifically, in the table, EMD+SVM achieved 50.89 (g^{AIRM}) and 68.33 (g^{LEM}) on the

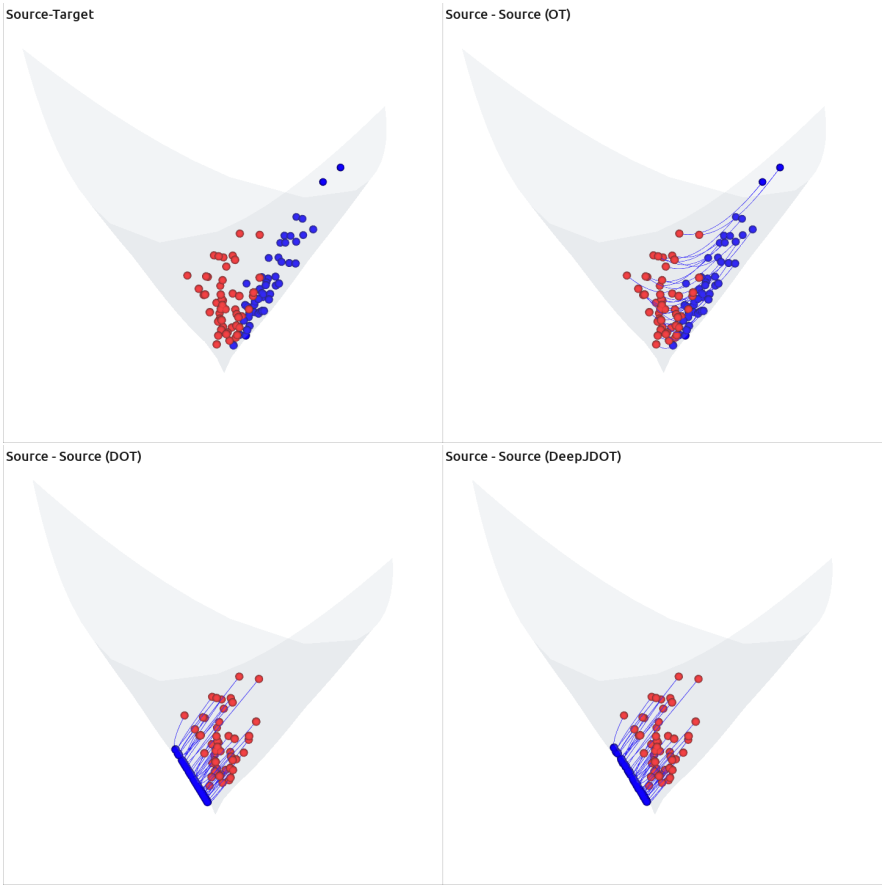


Figure 7.4: Synthetic Data of 2-dimensional SPD Cone: The subfigures from left to right and top to bottom are labeled as (a) to (d). In subfigures (a) to (d), the red points represent the source domain, while the blue points indicate their locations after the transformation.

BNCI2014001 dataset. Moreover, several observations are introduced as follows: 1). Participant Number: The improvement is related to the number of participants. On datasets with fewer participants, the enhancement brought by transfer methods is more significant, with several scenarios showing changes of more than 5%. 2). Base Classifier Results: The extent of improvement is linked to the base classifier's performance. Lower initial results from the base classifier provide more opportunities for transfer methods to achieve significant gains. 3). Methodological Factors: Improvements are associated with the classifier neural network architecture, parameter selection, and data preprocessing. Therefore, the conclusions of this study may not be universally applicable.

Furthermore, we summarize the observations of all methods affected by various Riemannian metrics as follows:

- $(\mathcal{S}_{++}, g^{AIRM})$ includes categories of OT, PT, and DPT. Combining the results from all three datasets, the RCT method performs well, especially when paired with the MDM or SPDNet classifier. The EMD method shows the largest variation across datasets. The second step (ROT) in the RPA method, which is limited to semi-supervised scenarios, performs worse than the first step (RCT) of RPA. The RieBN method provides slight improvements but does not outperform RCT across the board.

- $(\mathcal{S}_{++}, g^{LEM})$ includes two categories of OT and DOT. The EMD method, when paired with the SVM classifier, performs significantly better compared to its pairing with the SPDNet classifier. SPDSW and logSW methods have obvious improvements compared to their base classifier (SVM), among which SPDSW performs better than logSW by 1% to 2% on all three datasets. The MDA and CDA methods perform similarly, except for the BNCI2014001 dataset. The improvements on the other two datasets are between 1% to 2%, while the MDA method is slightly better than the CDA method or a mixture of the MDA and CDA methods. Notably, the use of pseudo-labels in the CDA method affects its results. The DeepJDOT method's performance across three datasets falls short of our proposed approach. The primary reason for this could be attributed to our decision to replace the task of computing transportation costs for each data point with merely computing transportation costs for the centers of each frequency band's dataset. Minimizing the transportation costs does not necessarily imply improved classification accuracy. Thus, the proposed approach reduces computational cost while enhancing the model's potential.

Additionally, neural network-based approaches generally achieve better classification outcomes compared to other methods, such as EMD, SPDSW, and logSW. To compute the CDA loss in DOT, we used the pseudo-label method, with pseudo labels derived from the Graph-CSPNet model discussed in Chapter 6, which currently performs best for subject-specific tasks. Specifically, the results for SPDSW and logSW on the BNCI2014001 dataset were obtained using the data processing procedure and source codes provided by their respective GitHub repositories, and they outperformed our results. This is likely because the performance of SVM-based methods heavily depends on the selection of handcrafted features.

Table 7.2: Comparative experiment table for three public datasets: All results in the table are in percentages (%), and the comparison methods include four categories, nine methods, and the corresponding three base classifiers. The original results of the base classifiers without considering transfer methods are at the bottom of the table. In the table, we use the symbol "/" to indicate that this method does not exist in the scenario constructed by this dataset.

Metric of $\mathcal{S}_{++}$	Category	Method	Classifier	KU (54 Subjects) $S1 \longrightarrow S2$ Semi-supervised		BNCI2014001 (9 Subjects) $T \longrightarrow E$ Unsupervised		BNCI2015001 (12 Subjects) $S1 \longrightarrow S2$ Unsupervised	
				Accuracy	Increment	Accuracy	Increment	Accuracy	Increment
g^{AIRM}	OT	EMD	SVM	65.24 (14.81)	−0.07	50.89 (16.54)	−14.27	82.17 (11.80)	+6.25
	PT	RCT	MDM	52.69 (6.39)	+0.34	55.13 (12.48)	+0.89	77.67 (13.23)	+5.46
		ROT	MDM	51.39 (5.89)	−0.96	/		/	
		RCT	SPDNet	69.91 (13.67)	+2.15	72.11 (12.99)	+3.51	81.25 (10.64)	+4.75
		ROT	SPDNet	63.44 (14.94)	−4.32	/		/	
	DPT	RieBN	SPDNet	67.57 (15.27)	−0.19	69.33 (13.49)	+0.73	78.83 (14.56)	+2.33
g^{LEM}	OT	EMD	SVM	67.70 (15.14)	−0.06	68.44 (15.79)	+3.28	82.17 (11.80)	+6.25
		EMD	SPDNet	64.22 (14.30)	−3.54	62.15 (14.69)	−6.45	78.33 (13.80)	+1.83
		SPDSW	SVM	66.67 (14.54)	+1.36	67.82 (16.71)	+2.66	82.42 (11.74)	+6.50
		logSW	SVM	66.26 (15.00)	+0.95	65.82 (16.56)	+0.66	80.75 (12.94)	+4.83
	DOT	DeepJDOT	SPDNet	67.38 (14.67)	−0.38	61.00 (11.40)	−7.60	77.63 (15.12)	+1.13
		MDA	SPDNet	69.19 (14.37)	+1.43	67.78 (12.46)	−0.82	78.33 (14.57)	+1.83
		CDA	SPDNet	68.80 (14.29)	+1.04	65.12 (11.34)	−3.48	78.25 (14.59)	+1.75
		MDA+CDA	SPDNet	68.61 (14.79)	+1.05	64.85 (11.54)	−3.75	78.29 (11.99)	+1.79
Base Classifier			SVM	65.31 (14.42)		65.16 (17.00)		75.92 (16.29)	
			MDM	52.35 (6.73)		54.24 (13.31)		72.21 (14.67)	
			SPDNet	67.76 (14.55)		68.60 (13.03)		76.50 (14.41)	

7.3 Discussions

The experimental results indicate that all transfer learning methods demonstrate effectiveness. However, it is important to note that while domain shift is a recognized factor in these tasks, classification performance is not solely influenced by it. As a result, the improvements achieved by transfer learning methods are generally limited. Currently, there is no effective method to quantify the intensity of domain shift or a system to correlate this intensity with classifier performance, making it difficult to fully assess the effectiveness of these methods.

In the remainder of this section, we will discuss two issues related to the proposed approach.

Theorem 7.3 with cost function $c(x, \bar{x}) = ||x - \bar{x}||^2_{\ell^2}$

We claim that Theorem 7.3 with cost function $c(x, \bar{x}) = ||x - \bar{x}||^2_{\ell^2}$ is equilvant to [133, Theorem 3.1]. It was first observed and proved by Yair et al.[164], who use the matrix vectorization operator to establish an isometry between the affine transformation and the Bi-Map transformation. We follow their arguments and revise a bit of their proof.

Theorem 7.5. *[133, Theorem 3.1] Let μ and ν be two discrete distributions on $\mathbb{R}^n$ with N Diracs. Given a strictly positive definite matrix A, bias weight $b \in \mathbb{R}^n$, and source samples $\{x_i\}_{i=1}^N \sim \mu$, suppose target samples $\bar{x}_i := Ax_i + b$, for $i = 1, ..., N$, and the weights in both source and target distributions' empirical distributions are $1/N$. Then, transport $T(x_i) = Ax_i + b$ is the solution to the OT Problem provided with a cost function $c(x, \bar{x}) := ||x - \bar{x}||^2_{\ell^2}$.*

To expose the relationship between the Bi-Map transformation in Theorem 7.3 and affine transformation in Theorem 7.5, we introduce the matrix vectorization operator $vec(\ \cdot\ )$ to stack the column vectors of matrix $A = (a_1 | a_2 | \cdots | a_n)$ below on another as follows,

$$vec(A) := \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}.$$

Then, we have the following lemma.

Lemma 7.6. *$vec(WSW^\top) = (W \otimes W)vec(S)$, where $\otimes$ is Kronecker product.*

Proof. Suppose S is an $n \times n$ matrix, and $W^\top = (w_1^\top | w_2^\top | \cdots | w_m^\top)$ is the transpose of $m \times n$ matrix W with each column vector $w_i^\top \in \mathbb{R}^n$ (i=1, ..., m). The k-th column vector of $WSW^\top$ can be written as $(WSW^\top)_{(:,k)} = WSw_k^\top = (w_k^\top \otimes W)vec(S)$. Hence, we achieve the lemma as follows,

$$\begin{aligned} vec(WSW^\top) &= \begin{pmatrix} (WSW^\top)_{(:,1)} \\ \vdots \\ (WSW^\top)_{(:,m)} \end{pmatrix} \\ &= \begin{pmatrix} w_1^\top \otimes W \\ \vdots \\ w_m^\top \otimes W \end{pmatrix} vec(S) \\ &= (W \otimes W)vec(S). \end{aligned}$$

□

Lemma 7.7. *Suppose $W \in \mathbb{R}^{n \times n}$ is positive definite, then $W \otimes W$ is also positive definite.*

Proof. This is given by a fact that suppose $\{\lambda_1, ..., \lambda_n\}$ are the eigenvalues of W, then the eigenvalues for $W \otimes W$ is $\{\lambda_1^2, ..., \lambda_n^2\}$. Hence, $W \otimes W$ is positive definite. □

According to Lemma 7.6 and 7.7, the Bi-Map transformation $W \cdot S \cdot W^\top$ turns to be affine transformation $A = W \otimes W$, and thus Theorem 7.3. Hence, Theorem 7.3 endowed with the cost function $c\big(vec(x), vec(y)\big) = ||vec(x) - vec(y)||_{\ell^2}^2$ is equilvant to Theorem 7.5.

Parallel Transport

Parallel transport on Riemannian manifolds transfers a tangent vector at one point along a curve on the manifold to another point, such that the transported vector

remains parallel to the original vector [76]. Under two Riemannian metrics on SPD manifolds g^{AIRM} and g^{LEM}, expressions of parallel transport exhibit distinctly and determine two distinct modeling approaches for the problems [164, 173]. Specifically, for any S_1 and S_2 on $(\mathcal{S}_{++}, g^{AIRM})$, parallel transport from S_1 to S_2 for $s \in T_{S_1}\mathcal{S}_{++}$ can be expressed as $\Gamma_{S_1 \to S_2}(s) = EsE^\top$, where $E = (S_1 \cdot S_2^{-1})^{\frac{1}{2}}$. For any S_1 and S_2 on $(\mathcal{S}_{++}, g^{LEM})$, parallel transport is given by $\Gamma_{S_1 \to S_2}(s) = s$.

Keep in mind that DOT seeks a neural networks-based transformation $\varphi : X_S, X_T \longmapsto Z \subset \mathcal{S}_{++}$ such that $\mathbb{P}_S(\varphi(X_S)) = \mathbb{P}_T(\varphi(X_T))$ and $\mathbb{Q}_S(Y_S|\varphi(X_S)) = \mathbb{Q}_T(Y_T|\varphi(X_T))$, where $\varphi(X_S)$ and $\varphi(X_T)$ are transformation φ on the source and target feature spaces. Neural networks-based transformation φ extracts discriminative information from feature spaces. After the extraction, it is reasonable to view the feature space of the source domain as the same as it of the target domain, as they will be transformed under neural networks with the same weights, so are their transformed feature spaces, i.e., $X_T = X_S$ and $\varphi(X_T) = \varphi(X_S) \subset Z$, and thus we utilize the Log-Euclidean metric to formulate the space of covariance matrice.

Assumption 3

In this section, we will justify that Assumption 7.3 is reasonable from the numerical results. Table 7.3 records average Log-Euclidean distances in the same frequency band, and diagonal numbers are consistently the smallest among all the numbers in each row for the two datasets. This suggests that the mapping shifts the center of each frequency band in the source domain to the corresponding frequency band center in the target domain. It aligns with our intuition that the difference between the same frequency bands should be smaller since they capture similar signal characteristics in the same frequency range, and the difference between different frequency bands tends to be larger.

The procedure of calculating the average Log-Euclidean distance is specifically given as follows: assuming source and target samples are represented in a format of [trials, frequency bands, channels, channels], with frequency bands fixed (in the 2^{nd} dimension), we first calculate the Fréchet mean by averaging all covariance matrices along the trial dimension. This results in the shape of [frequency bands, channels, channels]. Next, we directly compute the average Log-Euclidean distance for each pair of frequency bands.

Table 7.3: Average Log-Euclidean distances between the Fréchet means of EEG spatial covariance matrices generated from different frequency bands in the source and target domains for the KU (54 subjects), BNCI2014001 (9 subjects), and BNCI2015001 (12 subjects) datasets. The shortest average Log-Euclidean distance in each row is highlighted in boldface.

KU: S1 \ S2	4∼8 Hz	8∼12 Hz	12∼16 Hz	16∼20 Hz	20∼24 Hz	24∼28 Hz	28∼32 Hz	32∼36 Hz	36∼40 Hz
4∼8 Hz	**4.69**	4.98	5.82	6.74	7.19	7.64	8.17	8.78	9.28
8∼12 Hz	5.09	**4.21**	4.82	5.81	6.25	6.70	7.22	7.82	8.31
12∼16 Hz	5.84	4.76	**4.01**	4.62	4.98	5.34	5.78	6.30	6.75
16∼20 Hz	6.66	5.61	4.37	**3.96**	4.08	4.33	4.64	5.02	5.40
20∼24 Hz	7.08	6.01	4.66	3.97	**3.86**	3.99	4.27	4.63	4.97
24∼28 Hz	7.52	6.45	4.98	4.11	3.86	**3.75**	3.91	4.21	4.52
28∼32 Hz	8.04	6.96	5.40	4.37	4.07	3.82	**3.73**	3.87	4.11
32∼36 Hz	8.65	7.56	5.93	4.77	4.42	4.13	3.88	**3.74**	3.82
36∼40 Hz	9.16	8.07	6.40	5.16	4.77	4.43	4.11	3.82	**3.74**

BNCI2014001: T \ E	4∼8 Hz	8∼12 Hz	12∼16 Hz	16∼20 Hz	20∼24 Hz	24∼28 Hz	28∼32 Hz	32∼36 Hz	36∼40 Hz
4∼8 Hz	**2.81**	4.25	5.14	5.88	6.70	8.30	9.64	10.97	12.46
8∼12 Hz	4.31	**2.37**	3.98	5.45	6.28	8.07	9.44	10.77	12.26
12∼16 Hz	5.16	3.91	**2.29**	3.53	4.23	5.83	7.15	8.42	9.87
16∼20 Hz	5.84	5.49	3.44	**2.17**	2.76	4.08	5.29	6.56	7.98
20∼24 Hz	6.59	6.24	4.18	2.75	**2.18**	3.31	4.54	5.79	7.24
24∼28 Hz	8.11	8.02	5.72	4.01	3.14	**2.20**	2.96	4.03	5.38
28∼32 Hz	9.42	9.38	7.06	5.21	4.39	2.83	**2.21**	2.86	4.05
32∼36 Hz	10.67	10.63	8.25	6.43	5.57	3.87	2.72	**2.22**	3.05
36∼40 Hz	12.16	12.12	9.69	7.85	7.02	5.18	3.85	2.70	**2.22**

BNCI2015001: $S_1 \setminus S_2$	4∼8 Hz	8∼12 Hz	12∼16 Hz	16∼20 Hz	20∼24 Hz	24∼28 Hz	28∼32 Hz	32∼36 Hz	36∼40 Hz
4∼8 Hz	**1.07**	12.39	12.35	10.36	9.71	8.46	5.73	2.25	6.24
8∼12 Hz	12.04	**1.06**	1.49	2.68	3.22	4.40	7.13	12.43	18.19
12∼16 Hz	12.10	1.60	**1.09**	2.52	3.13	4.32	7.08	12.41	18.18
16∼20 Hz	10.19	2.99	2.67	**1.19**	1.59	2.45	5.06	10.40	16.22
20∼24 Hz	9.64	3.40	3.15	1.53	**1.22**	1.90	4.51	9.84	15.66
24∼28 Hz	8.36	4.51	4.30	2.34	1.80	**1.19**	3.21	8.52	14.34
28∼32 Hz	5.72	7.17	7.01	4.94	4.31	3.06	**1.28**	5.73	11.54
32∼36 Hz	2.52	12.43	12.31	10.24	9.58	8.29	5.44	**1.30**	6.17
36∼40 Hz	6.41	18.22	18.12	16.09	15.42	14.14	11.27	5.88	**1.25**

Chapter 8

Score-Based Data Generation for Spatial Covariance Matrices

In EEG-based motor imagery classification, a significant challenge is the limited availability of training data for deep learning techniques [26]. To address this limitation, researchers have turned to generative modeling, a rapidly evolving field in machine learning, to generate synthetic EEG time series through a process known as data augmentation [174]. This technique involves creating plausible samples that were not present in the original dataset, thereby expanding the training data with unseen examples. As a result, developing generative models for spatial covariance matrices with neurocognitive relevance presents a promising approach to enhancing the efficacy of geometric classifiers, ultimately delivering tangible benefits to BCI research.

In this chapter, we generate spatial covariance matrices using a cutting-edge generative modeling technique known as score-based generative modeling. [1] Score-based generative modeling generates samples from noise by gradually increasing noise in the data, which is then reversed by estimating the score function, representing the gradient of the log-density function relative to the data. This noise perturbation is described as a forward diffusion process modeled by a stochastic differential equation [175–177]. This approach has been successfully applied to generating images,

[1] The work in this chapter has been published in 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society, 10.1109/EMBC40787.2023.10340899. (IEEE EMBC 2023)

audio, and point clouds. Unlike three-channel RGB images, which have pixel intensities ranging from 0 to 255, spatial covariance matrices are generally preprocessed as multi-channel square matrices that are symmetric, positive semidefinite, and consist of decimal values. We evaluate our approach using the KU dataset, the largest EEG-based motor imagery dataset for two-class motor imagery classification.

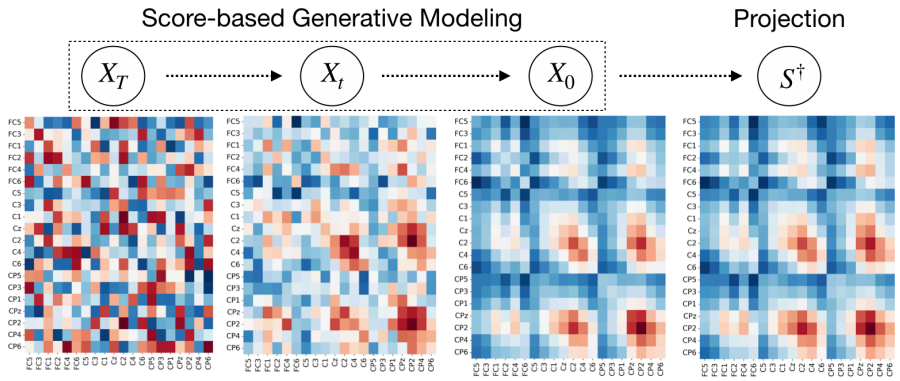


Figure 8.1: The sampling procedure can be depicted subsequently: Initially, we possess a noise matrix, represented as X_T. By employing the score-based generative modeling technique, we synthesize the spatial covariance matrix, X_0, with an intermediate state denoted as X_t. The generated spatial covariance matrix X_0 is approximated as being nearly SPD due to the acquired knowledge. To counterbalance numerical inconsistencies, we ensure the projection of X_0 as SPD by imposing a threshold upon all eigenvalues, denoted as $\epsilon = 1e - 4$. The arrangement of spatial covariance matrice channels proceeds sequentially from beginning to end: FC-5/3/1/2/4/6, C-5/3/1/z/2/4/5, and CP-5/3/1/z/2/4/6.

8.1 Methodology

This section proposes a two-step process for generating EEG spatial covariance matrices utilizing score-based generative modeling. The mathematical foundations of score-based generative modeling are presented in Appendix A. The process of sampling is depicted in Figure 8.1.

- **Score-based Generative Modeling**: During training, raw EEG signals undergo filtration and segmentation in frequency and temporal domains, employing methodologies delineated in [2, 25]. Explicitly, a collection of

bandpass filters dissect the EEGs into multiple-frequency passbands. Subsequently, a temporal segmentation scheme is executed to partition the EEGs into diminutive segments, with or without overlap. For a segment within T duration $X \in \mathbb{R}^{n_C \times n_T}$, the spatial covariance matrix is denoted as $S := X \cdot X^\top \in \mathbb{R}^{n_C \times n_C}$, where n_C and n_T represent the quantity of channels and timestamps, correspondingly. In the terminal procedure of this phase, spatial covariance matrices undergo scaling by division with their respective ℓ_2-norms, indicated as $\bar{S} := S/||S||_{\ell_2}$. Utilizing score-based generative modeling, the unknown prior distribution $p_{data}(S)$ is approximated through score matching, generating samples within specific frequency bands and temporal intervals of EEGs, employing either Langevin dynamics or time-reversal SDEs. The model is concurrently fitted for all frequency bands. During the sampling process, the generated samples consist of $n_C \times n_C$ matrices, albeit generally lacking symmetry and positivity.

- **Projection**: In this step, we approximate the generated matrices to preserve symmetry and positivity. Specifically, suppose a generated matrix $X \in \mathbb{R}^{n_C \times n_C}$, then the projected spatial covariance matrice as follows,

$$S^\dagger := \sum_{i=1}^{n_C} \max\{\lambda_i, \epsilon\} u_i u_i^\top,$$

where $\epsilon > 0$ is a preset threshold, eigenvalues $\{\lambda_i\}_{i=1}^{n_C}$ and corresponding orthonormal eigenvectors $\{u_i\}_{i=1}^{n_C}$ are crafted from symmetric matrix $\frac{1}{2}(X + X^\top)$.

8.2 Experimental Results

8.2.1 Evaluation Settings

In this research, we will utilize the KU dataset. By the subject-specific study of the KU dataset in Chapter 6, the subjects were divided into two groups: a *training subject group* and a *test subject group*. The criteria for inclusion in the *training subject group* were that the accuracies of 10-fold cross-validation on both S_1 and S_2 must be higher than 70% (criterion level). A total of 21 subjects met this criterion

and were included in the training subject group: Subjects No. 2, 3, 5, 6, 8, 12, 17, 18, 21, 22, 28, 29, 32, 33, 35, 36, 37, 39, 43, 44, and 45. The remaining 33 subjects were included in the *test subject group*: Subjects No. 1, 4, 7, 9, 10, 11, 13, 14, 15, 16, 19, 20, 23, 24, 25, 26, 27, 30, 31, 34, 38, 40, 41, 42, 46, 47, 48, 49, 50, 51, 52, 53, and 54.

Hyperparameters and Strategies in Training Models: For the score-based generative modeling, the Variance Exploding SDE approach, incorporating the NCSN++ model architecture [2], was selected for evaluation. We independently train two generative models to produce the left and right-hand samples using 4,200 trials in the *training subject group*, comprising 21 subjects over 2 sessions with 100 trials/per class each. The signal from each trial was initially transformed into a covariance matrix and scaled. The noise parameters were set to $\sigma_{max} = 10$ and $\sigma_{min} = 0.01$. The training process was performed over 100,000 iterations, utilizing a batch size of 128. Notably, the CNN filter size within NCSN++ was set to 20×20 due to the reason discussed in Section 8.3. We pick $\epsilon = 1e - 4$ in the projection step. To alleviate the discrepancy between the raw and generated distributions from the generative methods, we normalize each covariance matrix by zero-centering the means and scaling the variances to unity before visualization and quantitative analysis.

8.2.2 Results of Visualization

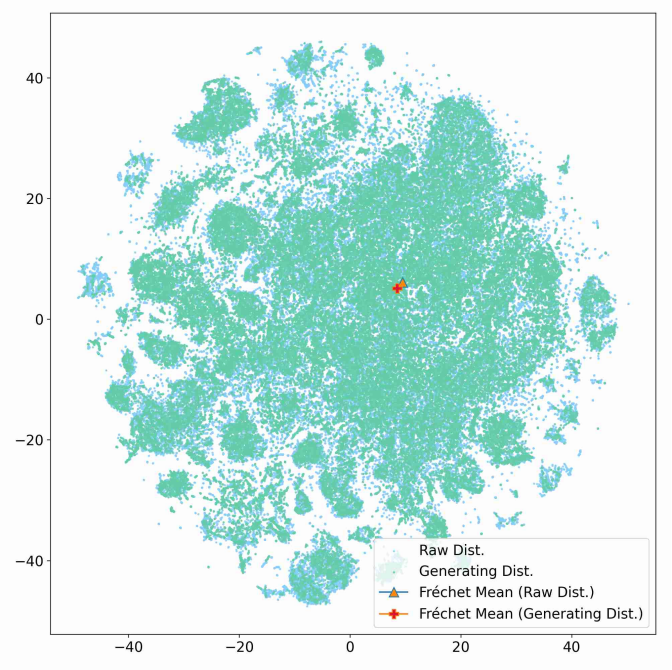
In Figure 8.2, there are in total of 151,200 points (i.e., 9 frequency bands $\times$ 8400 trials $\times$ raw and generating points) in Subfigure 8.2a and 84,000 points (i.e., 5 frequency bands $\times$ 8400 trials $\times$ raw and generating points) in Subfigure 8.2b, where the nine frequency bands include $4 \sim 8$ Hz, $8 \sim 12$ Hz, $12 \sim 16$ Hz, $16 \sim 20$ Hz, $20 \sim 24$ Hz, $24 \sim 28$ Hz, $28 \sim 32$ Hz, $32 \sim 36$ Hz, and $36 \sim 40$ Hz. Each two-dimensional point is projected from its associated 20×20 covariance matrices using t-SNE [115].

Figure 8.2a presents two-dimensional projections of the 8,400 raw spatial covariance matrices from the *training subject group* and the generated 8,400 covariance matrices (both left and right-hand classes), while Figure 8.2b provides a more

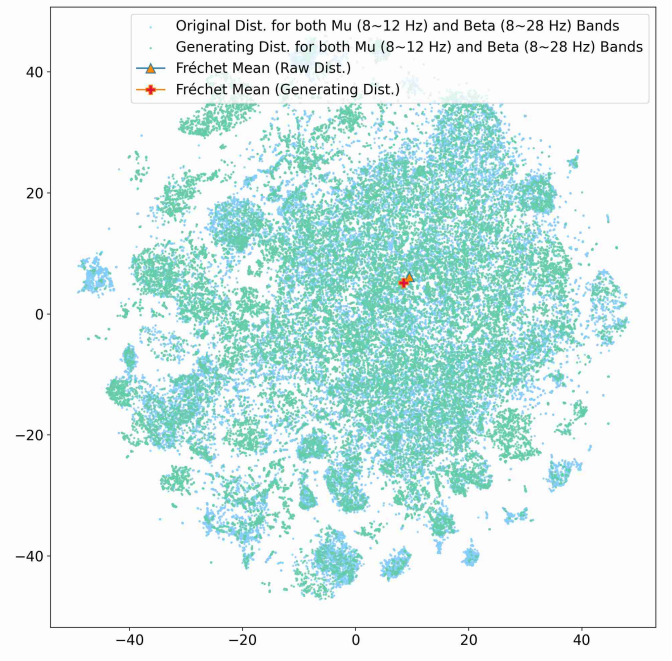
[2] The PyTorch implementation for Score-Based Generative Modeling refers to the GitHub repository (`https://github.com/yang-song/score_sde_pytorch`).

detailed view of the projections within the Mu and Beta frequency bands. Figures 8.3a and 8.3b illustrate the Fréchet means of covariance matrices for the nine frequency bands of the raw and generating distributions, respectively. We notice that the two distributions are nearly coincident and the Riemannian distance between their Fréchet means is relatively small. It indicates that the center of the learned distribution closely matches the raw distribution.

Figures 8.4 and 8.5 illustrate the Fréchet means of EEG spatial covariance matrices differentiated between the left and right-hand classes within nine frequency bands. Subfigures 8.4a and 8.4b display the spatial covariance matrices with key regions highlighted, corresponding to neurophysiological findings. Specifically, the highlighted entities in subfigures 8.4a and 8.5a (Mu and Beta bands) are situated in the regions of FC4, C4, and CP4 across the scalp, while those in subfigures 8.4b and 8.5b are located in the regions of FC3, C3, and CP3 across the scalp. To provide a more comprehensive visual representation of the texture of the generated samples, Figure 8.6 offers a closer examination of the spatial covariance matrices derived from actual EEGs and those generated from the proposed methodology for the two categories.

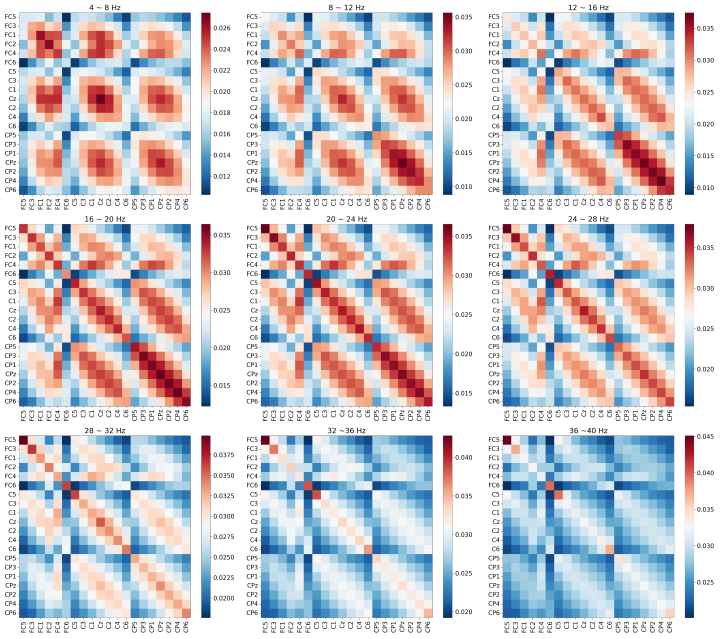


(a) Raw and generating distributions for all the nine frequency bands.

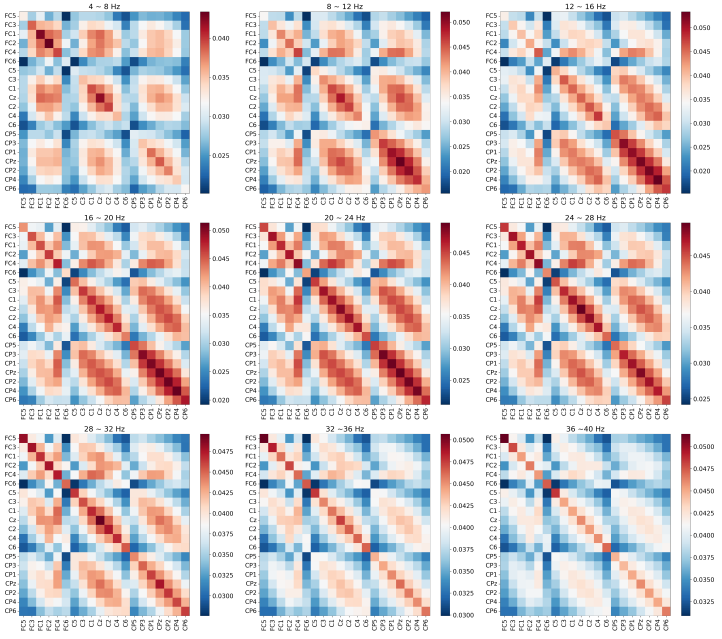


(b) Raw and generating distributions for the Mu and Beta frequency bands.

Figure 8.2: Illustration of the raw and generating distributions of two-dimensional projections of EEG spatial covariance matrices. The Fréchet means of both distributions are marked with triangle and cross signs.

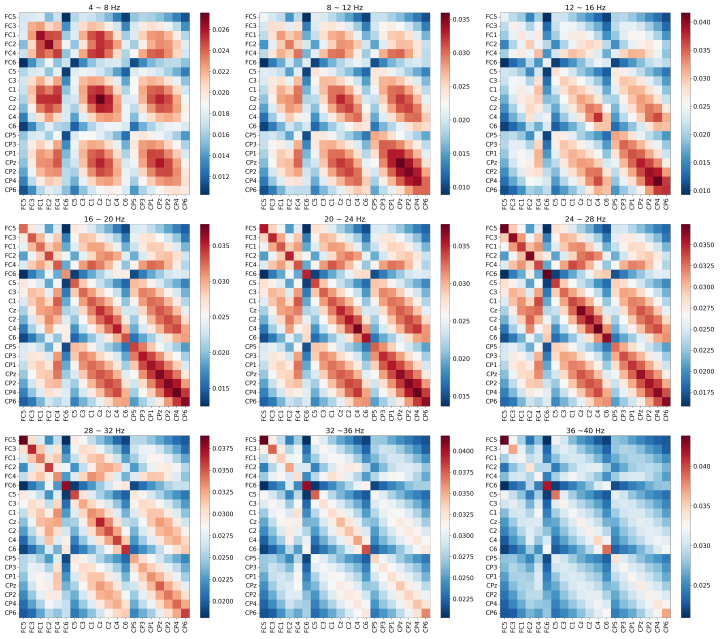


(a) Fréchet means of spatial covariance matrices in the raw dataset. (Triangle sign in Subfigure 8.2a)

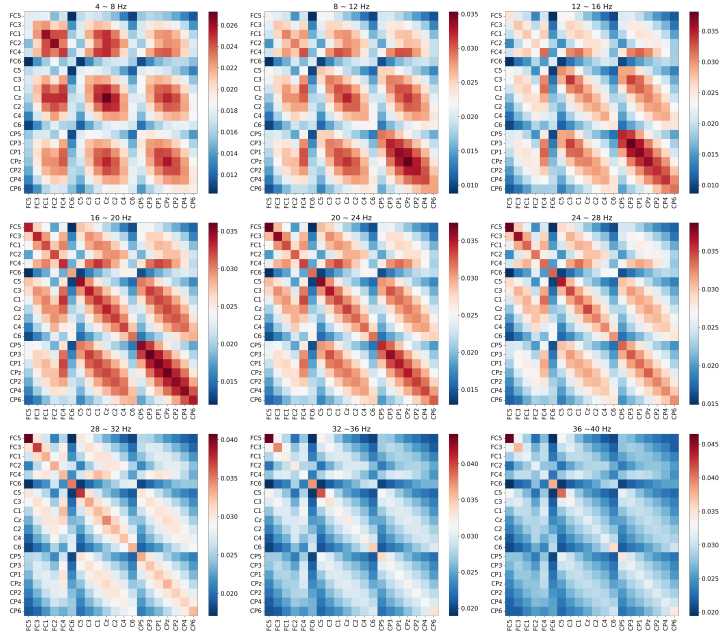


(b) Fréchet means of spatial covariance matrices in the generating dataset. (Cross sign in Subfigure 8.2b)

Figure 8.3: Illustration of Fréchet means of spatial covariance matrices in the raw dataset within all the nine frequency bands.

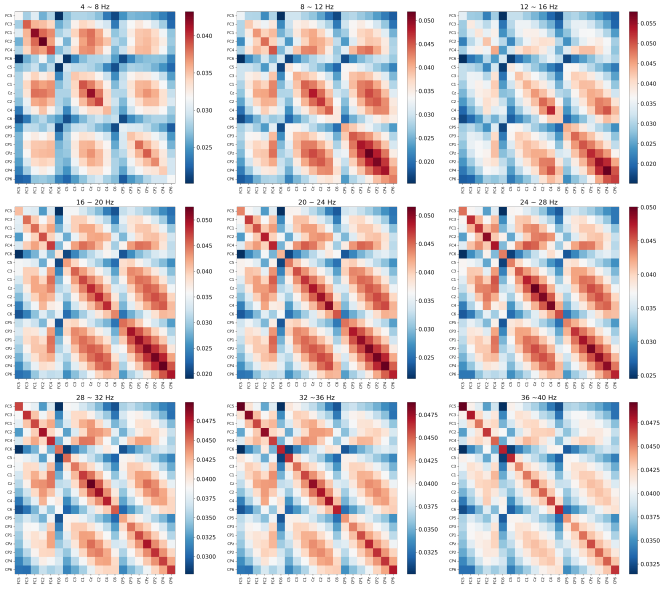


(a) Fréchet means of spatial covariance matrices derivated from the left-hand-class trials in the raw dataset.

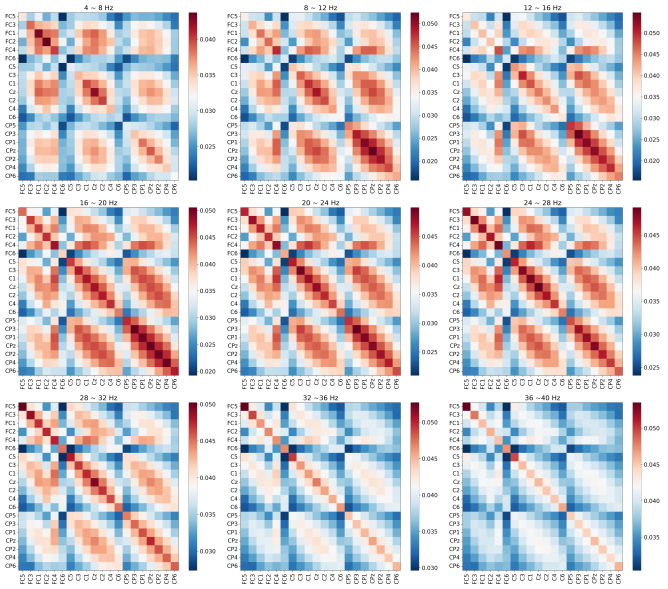


(b) Fréchet means of spatial covariance matrices derivated from the right-hand-class trials in the raw dataset.

Figure 8.4: Illustration of Fréchet means of spatial covariance matrices derived from the left and right-hand trials in the raw dataset within the nine frequency bands. The highlight entities of spatial covariance matrices in subfigure 8.4a (Mu and Beta bands) are located in the regions of FC4, C4, and CP4 over the scalp, while those in subfigure 8.4b fall in the regions of FC3, C3, and CP3.

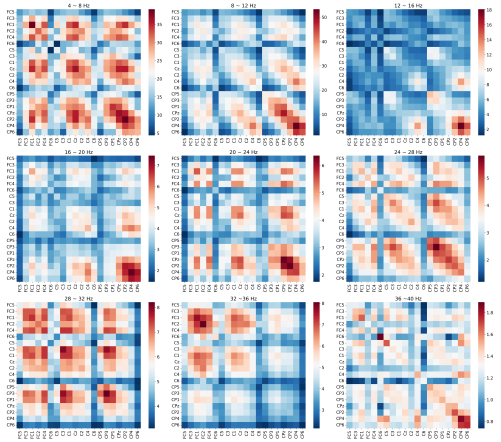


(a) Fréchet means of spatial covariance matrices derivated from the left-hand-class trials in the generating dataset.

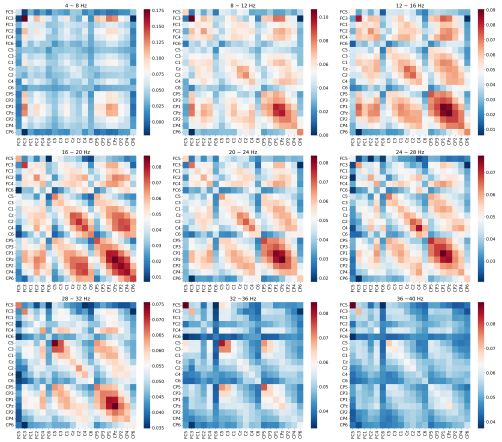


(b) Fréchet Mean of spatial covariance matrices derivated from the right-hand-class trials in the generating dataset.

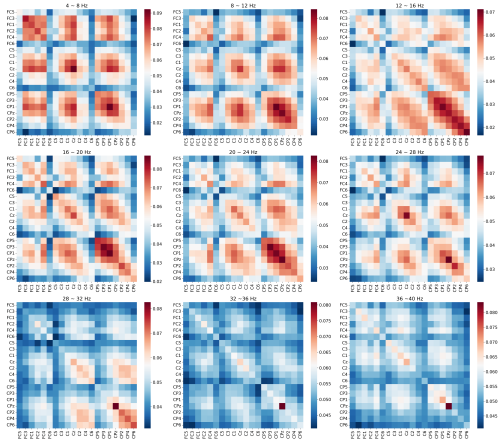
Figure 8.5: Illustration of Fréchet means of covariance matrices within the nine frequency bands for the left and right-hand trials in the generating dataset within the nine frequency bands. The highlight entities of spatial covariance matrices in subfigure 8.5a (Mu and Beta bands) are located in the regions of FC4, C4, and CP4 over the scalp, while those in subfigure 8.5b fall in the regions of FC3, C3, and CP3.



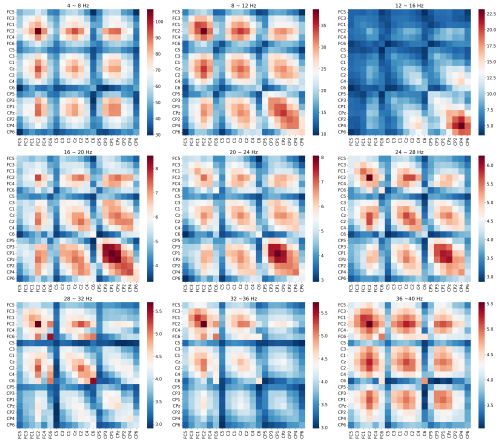
(a) Real spatial covariance matrices, left-hand (Trial No.27 of Subject No.1).



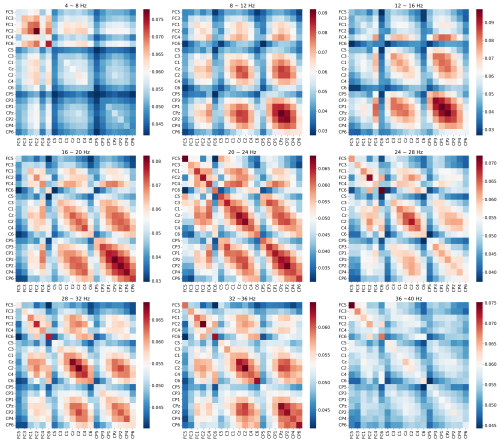
(b) A sample of the generated spatial covariance matrices, left-hand.



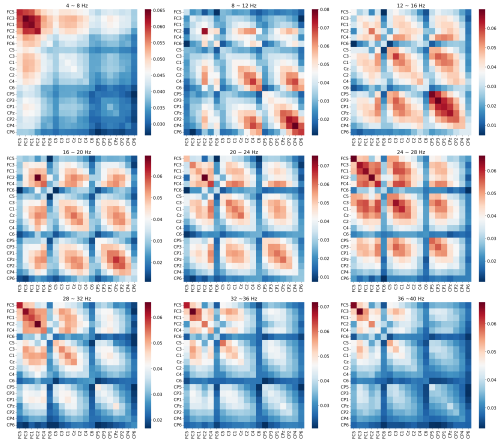
(c) A sample of the generated spatial covariance matrices, left-hand.



(d) Real spatial covariance matrices, right-hand (Trial No.8 of Subject No.1)



(e) A sample of the generated spatial covariance matrices, right-hand.



(f) A sample of the generated spatial covariance matrices, right-hand.

Figure 8.6: Conditional EEG spatial covariance matrice generation: In each line, we plot a picked spatial covariance matrix derived from an actual EEG segment for a category and another two generated samples within the same class.

8.2.3 Results of Classification Performance

To assess the generated samples' performance, we classify them using the pre-trained Tensor-CSPNet model on all subjects (two sessions) in the training subject group, which comprises a total of 8400 trials. The model architecture adopts a simplified 2500 ms time window and incorporates two-level BiMap layers, transforming the input dimension of 20 to 30 and back to an output dimension of 20. There are 8400 balanced generated samples, with each class containing 4200. The pre-trained classifier predicts an accuracy of 84.30% over all samples, and the confusion matrix is as follows:

Table 8.1: Confusion Matrix: Predicted labels in a total of 8400.

True \ Predicted	Right-hand	Left-hand
Right-hand	3730 (44.4%)	470 (5.6%)
Left-hand	849 (10.1%)	3351 (39.9%)

In this study, we conducted an additional experiment in a cross-session setting where one session of trials was utilized for training, the first half of another for validation, and the second half for testing, which is also known as the holdout scenario. This task presents a significant challenge due to the signal variability across sessions, and many state-of-the-art algorithms, including geometric methods [2, 25] performed poorly, yielding accuracy rates below 70%. The proposed generative method was applied to generate spatial covariance matrices using all subjects (two sessions) in the training subject group. The classifier, Tensor-CSPNet, was trained using the first session, and the generated samples were validated in the first half of the second session and evaluated in the second half for testing. Table 8.2 shows the cross-session classification accuracies, where each column of *None*, 20, 40, 60, 80, 100, 120, 140, 160, 180, and 200 represents the number of added generative samples to the training set. The *None* column results are the typical cross-session outcomes but applied to normalized spatial covariance matrices and without segmenting the time interval, thus slightly different from those in [25]. We selected Subjects 30 and 42 as representatives from the *testing subject group*. In the case of Subject 30, the average accuracy, calculated over ten runs, increased by 3.1% after adding 180 generated samples. Conversely, Subject 42 saw a substantial improvement of 8.7% in average accuracy after incorporating 200 generated samples in each trial.

Table 8.2: Cross-session classification with data augmentation approach: Each column depicts the number of samples incorporated into the training session. The samples are divided equally between the two classes: left-hand and right-hand. The selected cross-session scenario originates from the training and evaluation sessions in the KU dataset. The initial session of 200 trials and the added samples serve as the training data, while the first half of the second session, comprising 100 trials, is utilized for validation and the latter half, consisting of 100 trials, for testing purposes. The results (%) presented encompass the mean of 10 times runs across all scenarios and the optimal performance.

Argumentation Samples	None	20	40	60	80	100	120	140	160	180	200
Subject No.30											
Avg.(Std.)	55.2(3.9)	52.1(3.9)	55.5(6.8)	56.2(2.9)	54.7(5.1)	53.8(4.4)	56.7(4.8)	57.2(4.3)	56.8(6.4)	**58.3**(5.3)	57.6(4.5)
Best	61.0	59.0	**71.0**	63.0	64.0	62.0	64.0	66.0	66.0	66.0	67.0
Subject No.42											
Avg.(Std.)	59.2(3.5)	59.1(6.2)	63.2(4.6)	62.5(3.4)	65.6(4.6)	65.1(4.1)	64.8(3.6)	65.8(2.8)	65.4(3.6)	61.8(4.1)	**67.9**(2.5)
Best	63.0	66.0	69.0	67.0	72.0	**73.0**	71.0	70.0	72.0	69.0	72.0

8.3 Discussions

This study uses score-based generative modeling with the SDE approach to explore a new method for generating spatial covariance matrices for BCI applications. The generated samples are analyzed through both visual and quantitative evaluations. Visually, the samples produced by the proposed method have a comparable appearance to the spatial covariance matrices obtained from actual EEG recordings. Furthermore, the generated samples' center (Fréchet mean) aligns with neurophysiological findings that event-related desynchronization and synchronization occur on electrodes C3 and C4 within the Mu and Beta frequency bands during motor imagery processing. From a quantitative standpoint, 84.3% of the samples can be accurately predicted by a pre-trained Tensor-CSPNet, and holdout experiments on two subjects (Subject No.30 and No.42) show an improvement of up to 8.7% in the average accuracy of 10-times runs.

Although our findings are promising, the absence of evaluation metrics for generative models in the MI-EEG classification, akin to those commonly employed in the computer vision domain, such as Inception Score [178] and Fréchet Inception Distance [179], precludes our study from providing more detailed outcomes, such as individual participant results. It is crucial to recognize that not all participants exhibit noticeable improvements after incorporating generative samples. No established criterion exists for determining which subjects may benefit from this technique. Furthermore, the current approach has room for enhancement in several aspects, which we will address in the following:

Non-Euclidean Nature : In the experiments, the spatial covariance matrice channels are ordered from start to end as FC-5/3/1/2/4/6, C-5/3/1/z/2/4/5, and CP-5/3/1/z/2/4/6. The score-based generative model employs a CNN-structured architecture to capture local information from adjacent channels in this sequence. However, this order fails to reflect the correlations between EEG channels with respect to their spatial locations, a phenomenon referred to as the non-Euclidean nature, which results in limited performance. To tackle this problem, we propose a heuristic approach that sets the filter size to 20×20, which corresponds to the total size of the spatial covariance matrices. It may not readily apply to complex scenarios, as capturing the signal granularity with a large filter size can be challenging.

Randomness: Some generated samples may contain valuable discriminatory information for classification, while others may not. The randomness introduced by the sampling process in the score-based generative modeling may compromise the classifier's performance. Additionally, as we just mentioned, this randomness also leads to the ineffectiveness of conventional evaluation methods, which is due to the varying generated samples used for assessment each time, yet there is no standard evaluation metric for the generative models in the EEG-BCI classification. After all, the texture of EEG spatial covariance matrices is not even present in the general image recognition databases.

Cross-frequency Coupling: A potential explanation for the limited performance could be attributed to the diversity of the generative model. Since each spatial covariance matrix over a specific frequency band is independently generated from random noise, the composite spatial covariance matrix generated from these independent spatial covariance matrices may lack neurophysiological significance and have yet to be previously observed. In simpler terms, real spatial covariance matrices derived from the EEGs where changes in brain activity occur during cognitive and motor processing, resulting in event-related desynchronization and synchronization. However, a generative spatial covariance matrix may not have this same origin, even though it may appear similar. For instance, the spatial covariance matrix within the frequency range of 32 to 36 Hz, as depicted in Subfigure 8.6e, highlights a novel instance of the typical occurrence of high-intensity activities within the Mu and Beta bands.

Distribution Shift: Even though the generative samples may contain ample discriminatory information, the limited performance observed may still stem from the disparity between the prior and learned distributions. This incongruence can result in variations in the numerical ranges of pixels or entities within the spatial covariance matrices. To mitigate this challenge, we utilize a simple heuristic normalization technique for the covariance matrices by zero-centering the means and scaling the variances to unity. This approach results in well-overlapping raw and generated distributions, but it may not always be reliable in complex scenarios.

Chapter 9

Deep Geodesic Canonical Correlation Analysis

In human neuroimaging, it is common to use multi-modal imaging techniques to improve our understanding of the dynamic relationships among brain networks and their alterations in pathology [180–183]. For instance, combining non-invasive EEG measurements with fMRI can offer a high-resolution view of neurophysiological events relevant to treating mental disorders like epilepsy [184]. However, achieving a sense of consistency between EEG and fMRI frequently presents significant challenges due to a range of complications stemming from their disparities [185, 186].

A classical statistical learning approach to identify paired subspaces for consistency through correlation is Canonical Correlation Analysis (CCA) [187]. Its versatility is evident in a range of scenarios even when the sample size is limited relative to data dimensionality or when data dimensionality exceeds human interpretability [188]. Beyond classical CCA, extensions to study phase-amplitude coupling and amplitude-amplitude coupling proved useful in uni-modal [189] and multi-modal neuroimaging [190, 191]. For instance, Deligianni et al. [191, 192] used inverse modeling to align paired, resting-state EEG and fMRI signals spatially. After alignment, they used a CCA variant to link the covariance-based EEG and fMRI data and assessed the goodness-of-fit with a geometry-aware evaluation criterion.

When it comes to paired covariance-based data, employing conventional CCA methods directly leads to a loss of mathematical structure, including properties

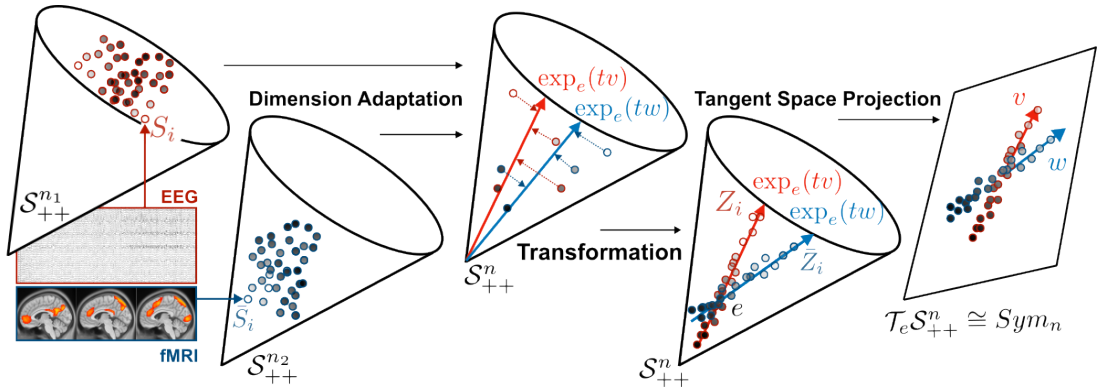


Figure 9.1: Deep Geodesic CCA (DeepGeoCCA) framework. We consider paired views of multivariate time-series data (e.g., simultaneously recorded EEG and fMRI data). Individual views are transformed (e.g., covariance matrices) to form paired views on SPD manifolds $\{(S_i, \bar{S}_i)|S_i \in \mathcal{S}_{++}^{n_1}, \bar{S}_i \in \mathcal{S}_{++}^{n_2}\}_{i=1}^{N}$. A *Dimension Adaptation* module transforms the views so that they reside within $\mathcal{S}_{++}^{n}$, a cone in $\mathbb{R}^{n\times n}$. On this cone, a *Transformation* module learns to center the data around e and project them onto geodesics $\exp_e(tv)$ and $\exp_e(tw)$ to form latent SPD representations $\{(Z_i, \bar{Z}_i)\}_{i=1}^{N}$. Our primary objective is to maximize geodesic correlation, which measures correlation along the speeds v and w, after *Tangent Space Projection* at the identify matrix e.

like symmetry and positive definiteness. Modeling techniques based on Riemannian geometry have become the primary approach to address such constraints in covariance-based neural data. In the case of CCA, extensions generalized the concept of subspace projection in Euclidean vector space to non-linear projections on geodesic submanifolds for general Riemannian manifolds [193] and SPD manifolds [193, 194].

In the context of multi-view deep representation learning [195], CCA inspired a family of self-supervised learning (SSL) approaches [196–199], as categorized in Balestriero et al. [200]. Methods like DeepCCA [201] and VICReg [199] proposed to learn the correlation relationship between two latent representations by analyzing their covariance and cross-covariance structure. However, these methods do not preserve their geometric properties while learning correlation relationships for covariance-based neuroimaging data with geometric characteristics.

In this chapter, we propose a geometric deep learning framework named Deep Geodesic Canonical Correlation Analysis, or DeepGeoCCA. [1] It belongs to the

[1] The work in this chapter has been published in the twelfth International Conference on Learning Representations, 2024. (ICLR 2024, Spotlight)

SSL-based deep representation learning family [202] as well as geometric modeling techniques to learn latent SPD representations with high correlation consistency among paired views in Figure 9.1. To characterize this correlation consistency, we present a new correlation measure called geodesic correlation, which is applied to SPD manifolds equipped with the affine-invariant Riemannian metric. Intuitively, geodesic correlation essentially measures the correlation of paired data along geodesics passing through the identity matrix e.

DeepGeoCCA introduces a novel geodesic loss function and SPD matrix-valued neural network models to maximize this measure to learn latent SPD representations. Specifically, we devise a relaxation method, called $\varepsilon-$geodesic constraint, that enables geometric classifiers to map covariance-based data onto or near geodesics within a double-cone region around e.

Using theoretical analysis and experiments with simulations and real data, we systematically study our framework, detailed in Section 9.2, and show that DeepGeoCCA:

- generates latent representations that maintain the properties of symmetry and positive definiteness.
- supports a *controllable* projection region. Specifically, we prove that adjusting ε in our relaxation approach can control the deviation of the mapped points from the geodesics.
- allows *flexible* unit speeds for predefined geodesics in the sense that they can be pre-defined by characteristics of downstream tasks or driven by data.
- effectively learns latent representations exhibiting significant geodesic correlation for previously unseen data while preserving pertinent information.

9.1 Prelimnary

We will introduce the idea of orthogonal projection to clarify the geodesic constraints.

Given that the SPD manifold $(\mathcal{S}_{++}, g^{AIRM})$ is geodesically complete, meaning that all geodesics are defined for all time $t \in \mathbb{R}$, any geodesic $\exp_e(tu)$ on $\mathcal{S}_{++}$ is solely determined by $u \in \mathcal{T}_e\mathcal{S}_{++}$ and varies smoothly with respect to u.

Given an unit speed $u \in Sym_n$, the orthogonal projection of $S \in \mathcal{S}^n_{++}$ on arc-length geodesic $\exp_e(tu) : \mathbb{R} \longmapsto (\mathcal{S}^n_{++}, g^{AIRM})$ is $S^\dagger = \exp_e(t^\dagger u)$ [2] with the following optimality condition:

$$t^\dagger := \arg\min_{t\in\mathbb{R}} d^2_{g^{AIRM}}\big(S, \exp_e(tu)\big). \tag{9.1}$$

When considering the Riemannian distance $d_{g^{AIRM}}$, Equation 9.1 becomes into the following expression:

$$t^\dagger := \arg\min_{t\in(-\epsilon,\epsilon)} \mathrm{Tr}\Big(\log_e^2\big(S^{-1}\exp_e(tu)\big)\Big). \tag{9.2}$$

Take the derivative of Equation 9.2 with respect to time t, set it to zero, and we obtain:

$$\begin{aligned} 0 &= \partial \mathrm{Tr}\Big(\log_e^2\big(S^{-1}\exp_e(tu)\big)\Big)\Big/\partial t \\ &\overset{(*)}{=} 2\mathrm{Tr}\Big(\log_e\big(S^{-1}\exp_e(tu)\big)\exp_e(-tu)\,\partial\exp_e(tu)/\partial t\Big), \end{aligned}$$

which yields an equality as follows,

$$\mathrm{Tr}\Big(\log_e\big(S^{-1}\exp_e(tu)\big)\exp_e(-tu)u\exp_e(tu)\Big) = 0. \tag{9.3}$$

Equality $(*)$ is valid by virtue of the following proposition [203]:

Proposition 1. Consider a real matrix-valued function $S(t)$ defined on time $t \in \mathbb{R}$. Assume that $S(t)$ is an invertible matrix and does not have eigenvalues on the closed negative real line. Then, we have the following equality:

$$\partial \mathrm{Tr}\Big(\log_e^2 S(t)\Big)\Big/\partial t = 2\mathrm{Tr}\Big(\log_e S(t)S^{-1}(t)\,\partial S(t)/\partial t\Big).$$

[2] A covariance matrix S marked with a $\dagger$ symbol in the upper right corner denoted as $S^\dagger$, signifies that it represents the orthogonal projection of S on the geodesic.

Suppose $t := t^\dagger$ is optimal and thus $S^\dagger := \exp_e(t^\dagger u)$. Hence, Equality 9.3 becomes to be as follows,

$$\mathrm{Tr}\Big(\log_e\big(S^{-1}S^\dagger\big)S^{\dagger^{-1}}uS^\dagger\Big) = 0. \tag{9.4}$$

9.2 Methodology

Like previous CCA variants, our framework resides within the SSL family, eliminating the need for labels. Using established criteria, we consider paired views of neural data from the same or different modalities. For example, we pair EEG and fMRI views based on their recording time. We use SPD neural networks to learn latent SPD representations by maximizing geodesic correlation. Formally, without loss of generality, we define the term geodesic correlation at the identity matrix $e \in \mathcal{S}^n_{++}$, as illustrated in Figure 9.2.

Definition 9.1. On $(\mathcal{S}^n_{++}, g^{AIRM})$, given paired covariance-based data $\{S_i\}_{i=1}^N$ and $\{\bar{S}_i\}_{i=1}^N$, each with Fréchet mean e, and two arc-length geodesics $\exp_e(tv)$ and $\exp_e(tw) : Sym_n \longmapsto \mathcal{S}^n_{++}$ with $t \in \mathbb{R}$ and unit speed $v, w \in \mathcal{T}_e\mathcal{S}^n_{++} \cong Sym_n$, respectively. Geodesic correlation with respect to v and w is defined as the correlation between their Riemannian logarithm of the orthogonal projections on the tangent space at e as follows,

$$\mathrm{corr}(\{S_i\}_{i=1}^N, \{\bar{S}_i\}_{i=1}^N) := \frac{\sum_{i=1}^N g_e^{AIRM}\big(\log_e(S_i^\dagger), \log_e(\bar{S}_i^\dagger)\big)}{\sqrt{\sum_{i=1}^N \Big\|\log_e(S_i^\dagger)\Big\|_e^2}\sqrt{\sum_{i=1}^N \Big\|\log_e(\bar{S}_i^\dagger)\Big\|_e^2}}, \tag{9.5}$$

where $S_i^\dagger$ and $\bar{S}_i^\dagger$ represent the orthogonal projections on the given geodesics with the following geodesic constraints:

$$\mathrm{Tr}\Big(\log_e\big(S_i^{-1}S_i^\dagger\big)S_i^{\dagger^{-1}}vS_i^\dagger\Big) = 0;$$
$$\mathrm{Tr}\Big(\log_e\big(\bar{S}_i^{-1}\bar{S}_i^\dagger\big)\bar{S}_i^{\dagger^{-1}}w\bar{S}_i^\dagger\Big) = 0,$$

for all $i \in \{1, ..., N\}$.

Geodesic correlation is better suited for use in the geometric deep learning framework for SPD manifold-valued data compared to conventional correlation and

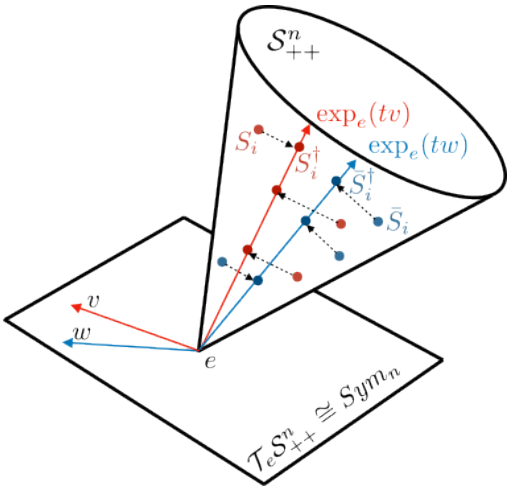


Figure 9.2: Geodesic Correlation on $\mathcal{S}^n_{++}$.

Riemannian correlation, illustrated in Figure 9.2. In particular, incorporating geodesic correlation and introducing geodesic constraints can be reformulated as a constrained nonlinear optimization problem, and its Lagrangian duality naturally serves as a loss function for the neural network-based solutions, as explained in Remark 9.2.

Remark 9.1. In this study, we employ parallel transport to relocate the Fréchet means of each data cluster to the identity matrix e. This process preserves the length since the Levi-Civita connection is metric-compatible. Hence, the Fréchet variance of the set of covariance data can be simplified as follows:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} \|\log_e(S_i)\|_e^2 ,$$

which is given by $d_{g^{AIRM}}(e, S) = \|\log_e(S)\|_e$.

Correlation, Riemannian Correlation, and Geodesic Correlation

We provide a detailed explanation of the distinctions between correlation, Riemannian correlation, and geodesic correlation. Figure 9.3 illustrates the distinctions

between the three terms. All of them are utilized to elucidate statistical relationships between two random variables, whether causal or not. However, their definitions encompass different scopes, as outlined below:

- Correlation: it is a general statistical term for any random variable [204].
- Riemannian Correlation: it is a generalized term of correlation for general Riemannian manifolds. It is quantified by computing the correlation between tangent vectors, which represent manifold-valued data projected onto the respective geodesic submanifold of the tangent spaces at the Fréchet mean of each modality [193].
- Geodesic Correlation: it is a generalized term of correlation for covariance-based neuroimaging data. It is quantified by calculating the correlation between tangent vectors, representing manifold-valued data projected onto the respective geodesics of the tangent spaces of each modality at the identity matrix e.

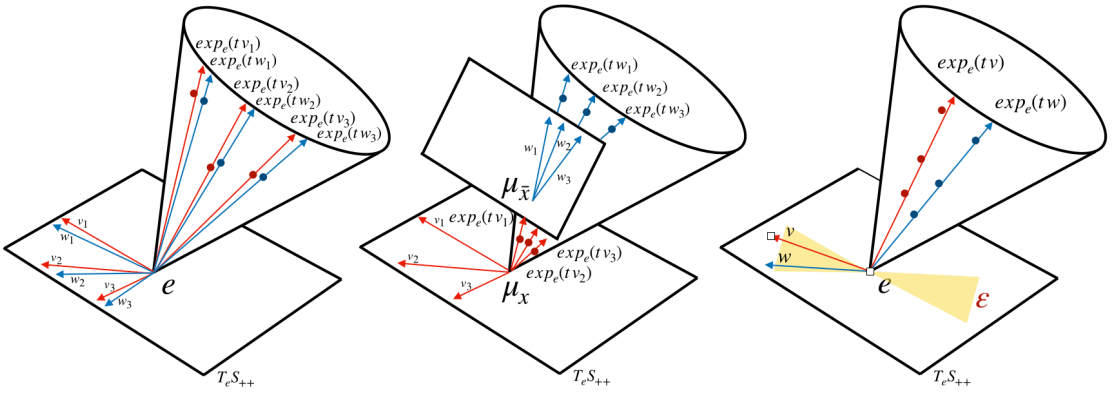


Figure 9.3: Illustration of Correlation, Riemannian Correlation, and Geodesic Correlation: From left to right, the concepts of Correlation, Riemannian Correlation, and Geodesic Correlation are depicted with specific descriptions as follows: 1). Correlation: This measures the relationship between paired data without requiring proximity between the pairs.; 2). Riemannian Correlation: This measure computes the correlation between paired projection data on the tangent spaces of each modality, considering their respective Fréchet means. The projection data is obtained by projecting the raw data onto geodesic submanifolds; 3). Geodesic Correlation: This measure calculates the correlation between paired projection data in the tangent space at the identity point e. The projection data is obtained by projecting the raw data onto or near geodesics.

Since $(\mathcal{S}_{++}, g^{AIRM})$ is a Cartan-Hadamard manifold and geodesically complete, every geodesic can be extended throughout the entire $\mathbb{R}$. These favorable geometric properties facilitate the simplification of Riemannian correlation within the SPD cone. We leverage this fact as follows:

- Geodesics: We simplify the constraint of a geodesic submanifold in Riemannian correlation to a geodesic.
- Base Point e of Tangent Space: We exclusively compute the correlation on the tangent space at the identity point e rather than the Fréchet mean for each modality. In particular, defining geodesic correlation at e is equivalent to defining it at any point $s \in \mathcal{S}_{++}$. This equivalence is established through parallel transportation.
- Correlation Measurement: We use the Riemannian metric to measure the correlation between two tangent vectors on the tangent space at the identity point e. In contrast, Riemannian Correlation relies on the inner product of scalar values of tangent vectors based on different Fréchet means.

Given these simplifications, geodesic correlation fulfills the requirements of a neural network-based solution. We summarize the primary distinctions from the four aspects discussed earlier in Table 9.1 as follows:

Table 9.1: Difference between Correlation, Riemannian Correlation, and Geodesic Correlation

Correlation	Region	Base Point	Correlation Measurement	Solver
Correlation	Line	0	Dot Product	CCA
Riemannian Correlation	Geodesic Submanifolds	Two Fréchet Means	Scalar Dot Product	RieCCA
Geodesic Correlation	Geodesics	e	Riemannian Metric	DeepGeoCCA

In fact, geodesic correlation has a relationship with correlation according to the following proposition:

Proposition 2. Suppose v and w are tangent vectors on $\mathcal{T}_e\mathcal{S}^n_{++}$ of $(\mathcal{S}^n_{++}, g^{AIRM})$, then we have

$$g_e^{AIRM}(v, w) = \langle\, vec(v), vec(w) \,\rangle_{\ell^2}.$$

Proof. This proposition arises from the following derivation,

$$g_e^{AIRM}(v, w) := \mathrm{Tr}(vw) = \sum_{i=1}^{N} v[i, :]w[:, i] = \langle\, vec(v^\top), vec(w) \,\rangle_{\ell^2} \overset{(*)}{=} \langle\, vec(v), vec(w) \,\rangle_{\ell^2},$$

where $(*)$ is derived from the property that the tangent space of $(\mathcal{S}^n_{++}, g^{AIRM})$ is a symmetric space. □

This proposition demonstrates that maximizing the geodesic correlation at the base point e is equivalent to maximizing the correlation if we view the covariance-based data as flattened.

9.2.1 Neural Network-Based Solution

As outlined in Figure 9.1, we transform paired views of neural data into latent SPD representations centered around e. Within our framework, we use them in three conceptual modules. The first module, known as *dimension adaptation* network, is responsible for converting input data of varying dimensions to a consistent dimension. The second module, referred to as *transformation* network, learns non-linear transformations to center the data around the identity matrix e and project them close to geodesics passing through e. Lastly, *tangent space projection* maps the observations to the tangent space at e to form inputs for our proposed loss terms. In particular, to enable computations to be performed at e, we employ Batch Normalization on $(\mathcal{S}^n_{++}, g^{AIRM})$ with specially tuned step momentum parameter to parallel transport Fréchet means of SPD data distribution to e.

9.2.2 Relaxed Orthogonal Projection

In this section, we propose a relaxation method for the geodesic constraints of geodesic correlation, which is particularly suited for the neural network-based solution. In intuitive terms, this relaxation assesses whether the projection $Z \cong S^\dagger$ on tangent space $\mathcal{T}_e\mathcal{S}^n_{++}$, after being normalized, falls within the double cone centered at identity e, with the unit geodesic speed u as the symmetric line, by calculating the cosine value between it and u, defined as follows,

$$\cos(\Theta) := \frac{g_e^{AIRM}\big(\log_e(Z), u\big)}{\|\log_e(Z)\|_e \, \|u\|_e}. \tag{9.6}$$

We introduce a relaxation parameter $\varepsilon \in [0, 1]$ to control how much the projections can deviate from the geodesic by $\varepsilon \leq |\cos(\Theta)|$. We refer to this new constraint as

the $\varepsilon-$geodesic constraint. Furthermore, we offer the following theorem to estimate the discrepancies between projections precisely on the geodesic and those within the tubular neighborhood of the geodesic $\exp_e(tu)$:

Theorem: Suppose $\{Z_i\}_{i=1}^N$ are a set of latent representations on $(\mathcal{S}_{++}^n, g^{AIRM})$, and a δ-width tubular neighborhood of arc-length geodesic $\exp_e(tu)$ with unit speed u is defined as $\{\exp_e(tu+v)\big|v \in (\mathcal{T}_e\mathcal{S}_{++}^n)^\perp, \|v\|_e \leq \delta, t \in \mathbb{R}\}$, where $(\mathcal{T}_e\mathcal{S}_{++}^n)^\perp$ represents the orthogonal complement of $\mathcal{T}_e\mathcal{S}_{++}^n$. Then, the width δ of this tubular neighborhood is upper bounded by the following inequality,

$$\delta \leq \sqrt{\frac{1-\varepsilon^2}{\varepsilon^2}} \max_{i=\{1,...,N\}} g_e^{AIRM}\big(\log_e(Z_i), u\big),$$

where relaxation parameter $\varepsilon \in (0, 1]$ regulates the deviation of the projections from the geodesics by $\varepsilon \leq |\cos(\Theta)|$. In particular, when $\varepsilon \to 1$, we have $\delta \to 0$.

Proof. The δ-width tubular neighborhood of a geodesic $\exp_e(tu)$ on $(\mathcal{S}_{++}^n, g^{AIRM})$ always exists inside its normal bundle according to general tubular neighborhood theorem [205].

For any predefined unit speed $u \in \mathcal{T}_e\mathcal{S}_{++}^n$, let the subspace of speed vectors $\mathcal{U} := \{tu\big|t \in \mathbb{R}\} \subset \mathcal{T}_e\mathcal{S}_{++}^n$. We can define the orthogonal complement of $\mathcal{U}$ as $\mathcal{U}^\perp := \{u^\perp\big|g_e^{AIRM}(u^\perp, u) = 0\} \subset \mathcal{T}_e\mathcal{S}_{++}^n$. Note that the orthogonal complement $\mathcal{U}^\perp$ is not an empty set, as the space of skew-symmetric matrices is the orthogonal complement of the space of symmetric matrices.

For each $i \in \{1, ..., N\}$, suppose $\log_e(S_i)^\perp$ are orthogonal components of $\log_e S_i$ with respect to u as follows,

$$\log_e(S_i)^\perp = \log_e(S_i) - g_e^{AIRM}(\log_e(S_i), u)u \in \mathcal{U}^\perp.$$

Assuming $\delta \in (0, 1]$ represents the deviation magnitude, it is defined as a magnitude with the direction that is perpendicular to the vector u that passes through $\log_e(S)$ in the tangent space. The proposed approach requires that its projections fall within a δ-width tubular neighborhood around u. This is expressed as:

$$u_i^\perp := \delta \frac{\log_e(S_i)^\perp}{\|\log_e(S_i)^\perp\|_e} \in \mathcal{U}^\perp, \quad \forall\, i = 1, ..., N.$$

Since the relaxation parameter $0 \leq \varepsilon < 1$, we measure the cosine value $\cos(\Theta)$ between $u^{\perp} + g_e^{AIRM}(\log_e(S), u)u$ and u, and get the following inequality:

$$\frac{|g_e^{AIRM}\left(u_i^{\perp} + g_e^{AIRM}(\log_e(S_i), u)u, u\right)|}{\left\|u_i^{\perp} + g_e^{AIRM}(\log_e(S_i), u)u\right\|_e} \geq \epsilon, \quad \forall\, i = 1, ..., N,$$

which implies a formula for δ as follows,

$$\delta \leq \sqrt{\frac{1-\varepsilon^2}{\varepsilon^2}} \max_{i=\{1,...,N\}} g_e^{AIRM}\left(\log_e(S_i), u\right).$$

Since $\max_{i=\{1,...,N\}} g_e^{AIRM}\left(\log_e(S_i), u\right)$ is upper bounded as it is a finite set, when $\varepsilon \to 1$, we have $\delta \to 0$.

□

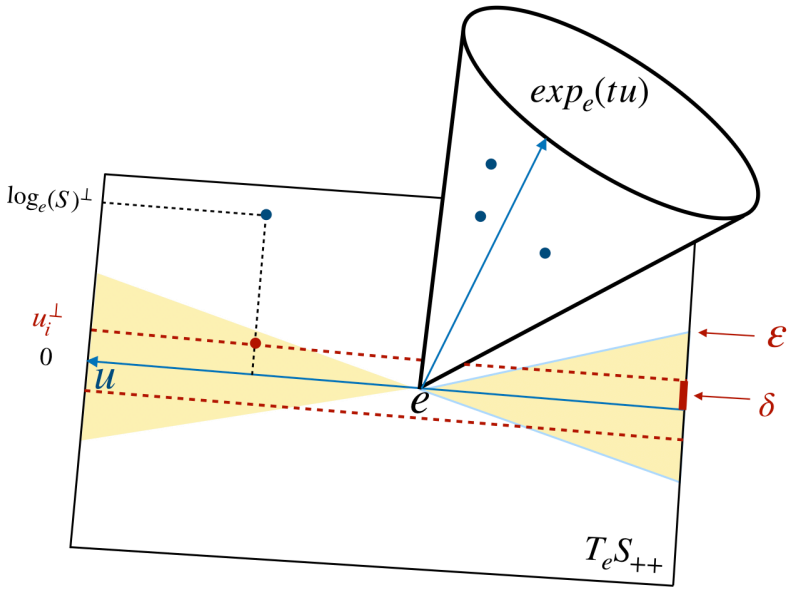


Figure 9.4: Illustration of Relaxation Deviation: The yellow region represents the double cone, where the maximum cosine value of the included angles is ε. The region within the red dashed lines denotes the tubular neighborhood of width δ along the speed of geodesic u on the tangent space.

Here, we briefly explain the rationale for introducing geodesics and the orthogonal projection onto them. The introduction of geodesics stems from a fundamental concept in Geometric Statistics known as the geodesic subspace, which was initially introduced in PGA and related works. In Riemannian geometry, a geodesic is a

curve that locally represents the shortest path between points, generalizing the concept of a straight line. Additionally, the geodesic subspace extends the idea of linear spaces from Euclidean to Riemannian geometry. This concept is crucial in PGA because projection onto the geodesic subspace preserves the Riemannian distance, thereby capturing the variance of manifold-valued data.

In a formal sense, a submanifold H of manifold M is considered geodesic at the point $p \in H$ if all geodesics of N that pass through p remain geodesic of M. A geodesic submanifold H at the Fréchet mean μ is represented as the exponential mapping of the linear span of r tangent vectors $\{w_i\}_{i=1}^r \in \mathcal{T}_\mu M$ as $N = \exp_\mu \left(\text{span}(\{w_i\}_{i=1}^r)\right)$.

In PGA, tangent vectors $\{w_i\}_{i=1}^r$ are computed by sequentially maximizing the Riemannian distance between each data point and the Fréchet mean. This process captures the variance of the data at each step while progressively removing the influence of previously derived directional projections. This completes the reduction of manifold-value data's dimensionality. In this context, w_1 can be analogous to the first principal component in PCA. This work goes beyond finding the geodesic submanifold's tangent vectors $\{w_i\}_{i=1}^r$, obtaining a novel low-dimensional data representation is crucial.

CCA aims to identify the maximum correlation between two variables. Specifically, CCA aims to find linear transformations for paired matrices $X \in \mathbb{R}^{N\times p}$ and $\bar{X} \in \mathbb{R}^{N\times q}$, denoted as $w \in \mathbb{R}^p$ and $\bar{w} \in \mathbb{R}^q$, respectively, to maximize the correlation between these two linear combinations Xw and $\bar{X}\bar{w}$. CCA is typically transformed into an eigenvalue decomposition problem using the Lagrange multipliers method. Consequently, the objective often involves identifying a set of r pairs of canonical variables along with corresponding linear transformations $W \in \mathbb{R}^{p\times r}$ and $\bar{W} \in \mathbb{R}^{q\times r}$, where each column in these matrices signifies a mapping to the respective pair of the canonical covariate. A constraint is imposed to render them uncorrelated to one another to ensure that each pair of canonical variables captures distinct phenomena.

When extending the CCA method to Riemannian manifolds, Kim et al. [193] was the first to replace geodesic submanifolds for linear subspaces. They did not underscore the orthogonal constraints for each pair of canonical covariates and obtained

them through a nonlinear optimization approach. Each resulting pair of canonical covariates corresponds to tangent vectors $\{w_i\}_{i=1}^{r}$ and $\{\bar{w}_i\}_{i=1}^{r}$ for two geodesic submanifolds. Therefore, orthogonal projection on the geodesic can be seen as projecting the data onto a lower-dimensional space by analogy with a linear transformation in classical CCA. Each tangent vector in the tangent space corresponds to a column vector in the linear transformation (canonical covariates).

In this study, our primary goal is to improve the temporal dynamic consistency of simultaneously recorded EEG-fMRI signals. To streamline the method, we utilize a single geodesic path for each modality. This design simplification results in a well-defined convex optimization problem and provides computational advantages within our deep learning architecture. Specifically, employing a single geodesic path for each modality is analogous to using the first pair of canonical covariates in CCA. In other words, the tangent vector w_1, which is a symmetric matrix (the tangent vector for SPD manifolds), directly corresponds to the first canonical covariate. Consequently, although the method for obtaining w_1 differs from the eigenvalue decomposition approach used in CCA or DeepCCA, our proposed component analysis method inherently includes a directional component.

Remark 9.2. This remark will show how to convert geodesic correlation into a constrained nonlinear optimization problem in matrix and Euclidean space. In matrix space, the constrained nonlinear optimization problem is depicted as follows,

$$\begin{aligned} \underset{v,\, w,\, S_i^{\dagger},\, \bar{S}_i^{\dagger}}{\text{minimize}} \quad & -\frac{\sum_{i=1}^{N} g_e^{AIRM}\left(\log_e(S_i^{\dagger}), \log_e(\bar{S}_i^{\dagger})\right)}{\left(\sum_{i=1}^{N} \left\|\log_e(S_i^{\dagger})\right\|_e^2\right)^{\frac{1}{2}} \left(\sum_{i=1}^{N} \left\|\log_e(\bar{S}_i^{\dagger})\right\|_e^2\right)^{\frac{1}{2}}} \\ \text{subject to} \quad & \operatorname{Tr}\Big(\log_e\left(S_i^{-1} S_i^{\dagger}\right) S_i^{\dagger^{-1}} v S_i^{\dagger}\Big) = 0, \\ & \operatorname{Tr}\Big(\log_e\left(\bar{S}_i^{-1} \bar{S}_i^{\dagger}\right) \bar{S}_i^{\dagger^{-1}} w \bar{S}_i^{\dagger}\Big) = 0 \end{aligned} \tag{9.7}$$

with variables $v, w \in Sym_n$ and $S_i^{\dagger}, \bar{S}_i^{\dagger} \in \mathcal{S}_{++}$ for $i = 1, ..., N$.

The consistency targeted by this constrained nonlinear optimization problem using correlation on the tangent space is referred to as geodesic correlation.

The constrained nonlinear optimization problem can also be written in Euclidean space. However, when the results obtained through a constrained nonlinear optimization algorithm in Euclidean space are converted back into matrices, they cannot be guaranteed to be positive definite and symmetric.

Next, we will derive the constrained nonlinear optimization problem in Euclidean space, equivalent to Problem 9.7 in the matrix space.

We substitute the correlation on $(\mathcal{S}_{++}^n, g^{AIRM})$ with conventional correlation denoted as follows,

$$t_v^i := vec\big(\log_e(S_i)\big), \text{and } t_w^i := vec\big(\log_e(\bar{S}_i)\big), \text{ for } i = 1, ..., N.$$

Then, we obtain the constrained nonlinear optimization problem in Euclidean space with a total of $2N$ constraints, depicted as follows,

$$\begin{aligned} \underset{v, w, t_v^i, t_w^i}{\text{minimize}} \quad & -\frac{\sum_{i=1}^N \langle t_v^i, t_w^i \rangle}{(\sum_{i=1}^N \|t_v^i\|_{\ell^2}^2)^{\frac{1}{2}} (\sum_{i=1}^N \|t_w^i\|_{\ell^2}^2)^{\frac{1}{2}}} \\ \text{subject to} \quad & \mathrm{Tr}\Big(\log_e\big(S_i^{-1}\exp_e(t_v^i v)\big)\exp_e(-t_v^i v) v \exp_e(t_v^i v)\Big) = 0, \\ & \mathrm{Tr}\Big(\log_e\big(\bar{S}_i^{-1}\exp_e(t_w^i w)\big)\exp_e(-t_w^i w) w \exp_e(t_w^i w)\Big) = 0 \end{aligned} \tag{9.8}$$

with variables $v, w \in Sym_n$ and $t_v^i, t_w^i \in \mathbb{R}^{n^2}$ for $i = 1, ..., N$.

By solving Problem 9.8 through constrained nonlinear optimization algorithms, even when the resulting t_v and t_w are converted into a matrix, it no longer maintains the positive definite symmetric properties.

In the following, we will derive the Lagrangian for Problem 9.7 in the matrix space. We arrive at the expression for Lagrangian $\mathcal{L}(S^\dagger, \bar{S}^\dagger, v, w, \lambda, \gamma) : \mathcal{S}_{++}^n \times \mathcal{S}_{++}^n \times Sym_n \times Sym_n \times \mathbb{R}^N \times \mathbb{R}^N \longmapsto \mathbb{R}$, as follow,

$$\begin{aligned} \mathcal{L}(S^\dagger, \bar{S}^\dagger, v, w, \lambda, \gamma) := & -\frac{\sum_{i=1}^N g_e^{AIRM}\big(\log_e(S_i^\dagger), \log_e(\bar{S}_i^\dagger)\big)}{(\sum_{i=1}^N \Big\|\log_e(S_i^\dagger)\Big\|_e^2)^{\frac{1}{2}} (\sum_{i=1}^N \Big\|\log_e(\bar{S}_i^\dagger)\Big\|_e^2)^{\frac{1}{2}}} \\ & + \sum_{i=1}^N \lambda_i \mathrm{Tr}\Big(\log_e\big(S_i^{-1}S_i^\dagger\big) S_i^{\dagger^{-1}} v S_i^\dagger\Big) \\ & + \sum_{i=1}^N \gamma_i \mathrm{Tr}\Big(\log_e\big(\bar{S}_i^{-1}\bar{S}_i^\dagger\big) \bar{S}_i^{\dagger^{-1}} w \bar{S}_i^\dagger\Big), \end{aligned}$$

where $\{\lambda_i \in \mathbb{R}\}_{i=1}^N$ and $\{\gamma_i \in \mathbb{R}\}_{i=1}^N$ are Lagrange multipliers.

In the context that follows, we refer to the orthogonal projection $S^\dagger$ or $\bar{S}^\dagger$ as Z or

$\bar{Z}$, respectively. [3] These orthogonal projections are obtained by neural network-based training using gradient descent. In practical implementation, these values precede the input to the LOG layer. The results produced by the LOG layer are denoted as $\log_e(Z_i)$ and $\log_e(\bar{Z}_i)$.

Hence, the Lagrangian duality is naturally a loss function for DeepGeoCCA, depicted as follows,

$$\begin{aligned}\mathcal{L}(Z,\bar{Z}) : = & -\frac{\sum_{i=1}^{N}\operatorname{Tr}\big(\log_e(Z_i)\log_e(\bar{Z}_i)\big)}{\sqrt{\sum_{i=1}^{N}\operatorname{Tr}\big(\log_e^2(Z_i)\big)}\sqrt{\sum_{i=1}^{N}\operatorname{Tr}\big(\log_e^2(\bar{Z}_i)\big)}} \\ & + \sum_{i=1}^{N}\lambda_i \operatorname{Tr}^2\Big(\log_e\big(S_i^{-1}Z_i\big)Z_i^{-1}vZ_i\Big) \\ & + \sum_{i=1}^{N}\gamma_i \operatorname{Tr}^2\Big(\log_e\big(\bar{S}_i^{-1}\bar{Z}_i\big)\bar{Z}_i^{-1}w\bar{Z}_i\Big),\end{aligned}$$

where $\lambda_i, \gamma_i \in \mathbb{R}$ are preset coefficients, for $i = 1, \ldots, N$, and $v, w \in Sym_n$ are preset tangent vectors.

9.2.3 Loss Function

To maximize geodesic correlation among paired views, we utilize the following loss:

$$\mathcal{L}_\rho := -\frac{\sum_{i=1}^{N}\operatorname{Tr}\big(\log_e(Z_i)\log_e(\bar{Z}_i)\big)}{\sqrt{\sum_{i=1}^{N}\operatorname{Tr}\big(\log_e^2(Z_i)\big)}\sqrt{\sum_{i=1}^{N}\operatorname{Tr}\big(\log_e^2(\bar{Z}_i)\big)}},$$

where Z_i and $\bar{Z}_i$ denote the latent SPD representations for two paired views, respectively, and symmetric matrices $\log_e(Z_i)$ and $\log_e(\bar{Z}_i)$ are their projections on the tangent space at e. In addition, we use the relaxed $\varepsilon-$geodesic constraints to penalize projections outside the double-cone regions as follows,

$$\mathcal{L}_\varepsilon := \sum_{i=1}^{N}\min\{\varepsilon, 1-\frac{\big|\operatorname{Tr}\big(\log_e(Z_i)v\big)\big|}{\|\log_e(Z_i)\|_e\,\|v\|_e})\} + \sum_{i=1}^{N}\min\{\varepsilon, 1-\frac{\big|\operatorname{Tr}\big(\log_e(\bar{Z}_i)w\big)\big|}{\big\|\log_e(\bar{Z}_i)\big\|_e\,\|w\|_e}\},$$

where geodesic speeds $v, w \in Sym_n$ are pre-defined or learnable parameters, and ε is a preset relaxation parameter. To mitigate potential dimensional collapse,

[3] Letter Z is commonly recognized as symbols representing latent representations in deep learning.

frequently encountered in the SSL framework [206], we use a variance-preserving loss proposed in [199] to preserve the variance of the original data distribution using a hinge function applied to the standard deviation of the projections along the batch dimension as follows:

$$\mathcal{L}_\sigma := \frac{1}{d}\sum_{i=1}^{d} \max\left\{0, 1 - \sqrt{\mathrm{Var}\big(S(Z)^i\big) + \epsilon}\right\} + \frac{1}{d}\sum_{i=1}^{d} \max\left\{0, 1 - \sqrt{\mathrm{Var}\big(S(\bar{Z})^i\big) + \epsilon}\right\},$$

where d is the dimension of $vec(\log_e(Z))$, $S(Z)^i$ is the vector composed of values at the i^{th} dimension of $\big(vec(\log_e(Z_1)), \cdots, vec(\log_e(Z_N))\big) \in \mathbb{R}^{N\times d}$, and $\epsilon = 10^{-4}$ is a scalar to prevent numerical instabilities.

The overall loss function $\mathcal{L}$ to maximize geodesic correlation combines the correlation loss $\mathcal{L}_\rho$, the relaxed $\varepsilon-$geodesic constraints $\mathcal{L}_\varepsilon$, and the variance-preserving loss $\mathcal{L}_\sigma$, as follows,

$$\mathcal{L} := \alpha_1\mathcal{L}_\rho + \alpha_2\mathcal{L}_\varepsilon + \alpha_3\mathcal{L}_\sigma. \tag{9.9}$$

with coefficients α_1, α_2, and $\alpha_3 \geq 0$.

Remark 9.3. 1). Choice of Geodesic Speeds v and w: The selection of the geodesic speeds v and w for two covariance-based modalities can be influenced by the specific requirements of downstream tasks or solely dictated by the data. For example, a choice can be the most significant canonical correlation obtained from the conventional CCA method as $vec(v), vec(w) := \mathrm{CCA}(\{vec(S_i)\}_{i=1}^N, \{vec(\bar{S}_i)\}_{i=1}^N)$.

2). Choice of Loss Coefficients: Depending on the specific scenario and architecture, the coefficients $(\alpha_1, \alpha_2, \alpha_3)$ can be determined to give more importance to the geodesic correlation or the constraints. By default, we set $\alpha_1 = 1, \alpha_2 = 0.25, \alpha_3 = 0$ to give more weight to $\mathcal{L}_\rho$. If a specific architecture tends to dimensional collapse, we enable the variance-preserving loss $(\alpha_3 > 0)$.

3). Loss function on Sym_n: Note that all loss terms in Loss 9.9 operate with tangent space projections $\log_e(Z_i), \log_e(\bar{Z}_i) \in \mathcal{T}_e\mathcal{S}^n_{++}$ corresponding to the space of symmetric matrices Sym_n. Consequently, our proposed loss can also be computed for any neural network generating latent representations in Sym_n. For example, the output $x \in \mathbb{R}^{n(n+1)/2}$ of any standard neural network layer can be transformed to Sym_n via applying upper^{-1} where the norm-preserving map upper vectorizes a matrix's elements along the upper triangular part.

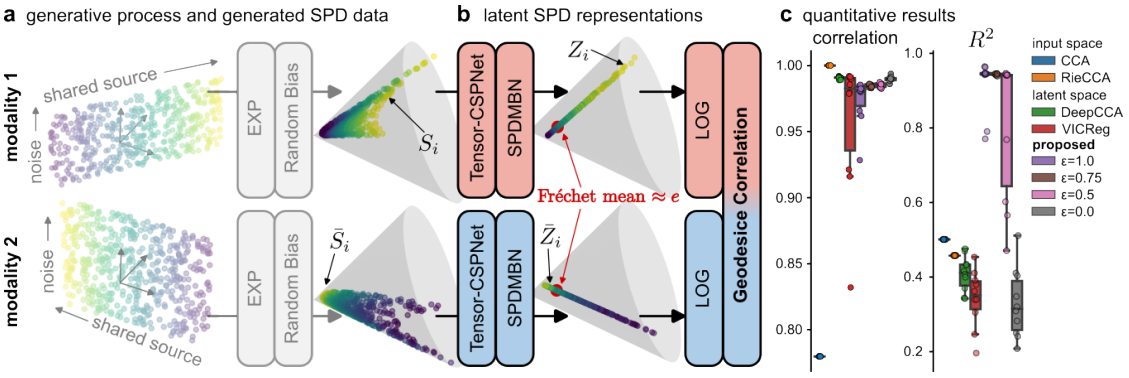


Figure 9.5: Simulations on $\mathcal{S}^2_{++}$. **a**, Generative process and visualization of generated observations ($N = 500$) with a shared latent source. **b**, Visualization of the latent representations learned with DeepGeoCCA ($\varepsilon = 0.75$). **c**, Boxplots summarize the simulation results for test data (10-fold CV). DeepGeoCCA can obtain high geodesic correlation (left panel; higher is better) and at the same time approx. constrain the representations to geodesics (right panel; coefficient of determination R^2; mean across modalities; higher is better).

9.3 Experimental Results

We conducted simulations and two experiments using public datasets to empirically evaluate our proposed framework in challenging neuroimaging tasks. These tasks present several unique challenges, including relatively small datasets, multivariate non-stationary signals with complex dynamics, distribution shifts (particularly severe across subjects, as illustrated in Figure 9.8), low signal-to-noise ratios, and non-linear coupling mechanisms between EEG and fMRI.

9.3.1 Simulations

To study the effect of the $\varepsilon-$geodesic constraint and demonstrate that our framework can recover the activity of a latent, shared source, we generated paired SPD data. Specifically, we generated a shared noise source with fixed, modality-specific projections to Sym_2 and paired it with a modality-specific noise source. To generate matrices on $\mathcal{S}^2_{++}$, we applied the matrix exponential and then biased these matrices to vary around a random, modality-specific Fréchet mean. Figure 9.5 summarizes the generative process and the considered architecture. For both modalities, we

used a Tensor-CSPNet as a dimension adaptation and transformation module (Figure 9.5b). As a tangent space projection module, we combine an SPDMBN layer and a LOG layer.

To quantify performance, we used the fitted models to transform $\{(S_i, \bar{S}_i)\}_{i=1}^N$ into latent representations $\{(Z_i, \bar{Z}_i)\}_{i=1}^N$, converted their tangent space projections to Euclidean vector space features $(X, \bar{X})$, and submitted these features to standard CCA to obtain $(z, \bar{z})$ for the first component. To do so, we computed the matrix logarithm and extracted elements along the upper triangular part while preserving the norm. CCA was fitted to the CV split's entire training data. We report Pearson correlation coefficients between z and $\bar{z}$ to measure geodesic correlation, and the coefficient of determination (R^2) to evaluate the geodesic constraint. We use R^2 to quantify how much variance in X (or $\bar{X}$) can be explained by z (or $\bar{z}$), i.e., $R^2 = 1 \Leftrightarrow$ data distributed along a geodesic.

Figure 9.5c summarizes our framework's cross-validation (CV) results for various ε compared to several baseline methods. The baseline methods were fitted to the generated SPD data (CCA and RieCCA) or used as alternative loss terms (DeepCCA, VICReg) for the considered architecture. For $\varepsilon = 0$ (no geodesic constraint), DeepGeoCCA shared the highest correlation score with DeepCCA among the deep learning methods (left panel) at the cost of small explained variance (i.e., small R^2) in the latent representation (right panel). This result can be expected because both objectives only aim to maximize geometric correlation in latent space. For $\varepsilon = 1$, the trained network traded a noticeable reduction in correlation for minimizing deviations from the geodesic constraint. Moreover, for some CV splits, the network got stuck in poor local minima (noticeable lower correlation and R^2). We found that $\varepsilon = 0.75$ resulted in a good trade-off between both objectives while avoiding poor local minima. Scatterplots in Figure 9.5b visualize the learned representations ($\varepsilon = 0.75$) for a representative CV split. Comparing the correlation scores of CCA and RieCCA demonstrates the importance of utilizing the mathematical structure of SPD matrices. Overall, RieCCA obtained the highest correlation score. This is likely because RieCCA had access to all the training data to find the optimal solution for this toy problem, while the deep learning approaches were fitted to randomly sampled mini-batches.

9.3.2 Simultaneous EEG-fMRI

Implementation Details

The implementation details provided in this section apply to the simulation and EEG-fMRI experiments.

Architecture

Figures 9.6 and 9.7 visualize the architecture used in the simulation and EEG-fMRI experiments.

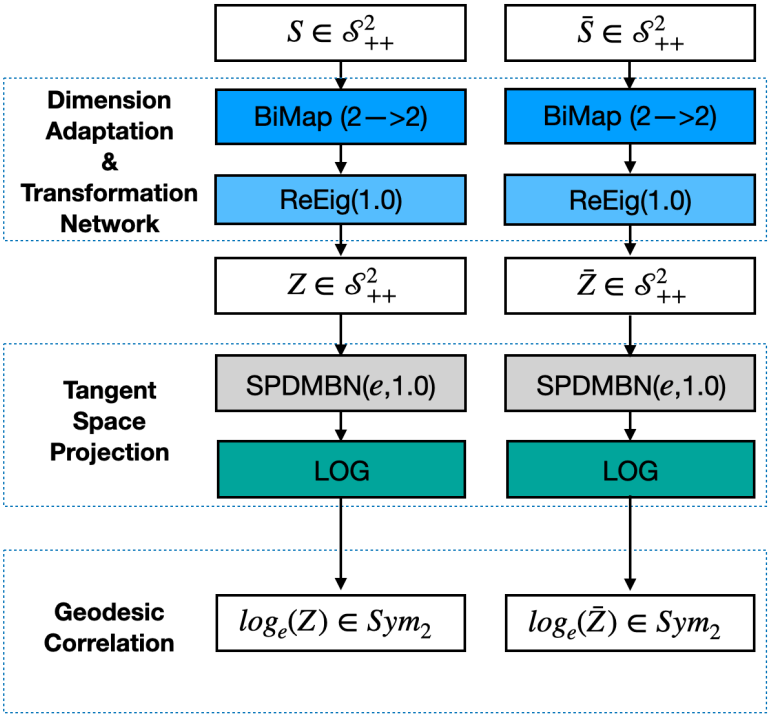


Figure 9.6: Illustration of network architecture for the simulations with paired SPD manifold-valued data. The BiMap layers used unconstrained parameters. For the ReiEig layers, we used a threshold of 1.0. The tangent space projection module used an SPDMBN layer without learnable parameters, i.e., the rebias parameter was fixed at the identity matrix e and the scaling parameter was fixed to 1..

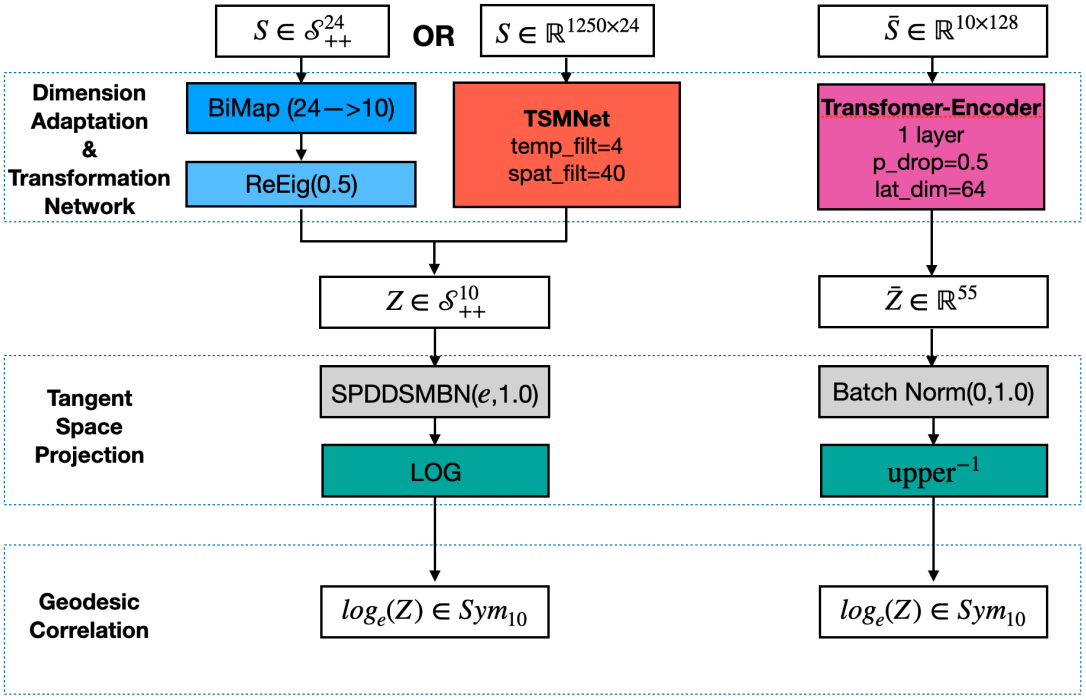


Figure 9.7: Illustration of Network Architecture for the EEG-fMRI experiment. For EEG views $\bar{S}$, we used either Tensor-CSPNet or TSMNet. For the ReiEig layer within Tensor-CSPNet, we used a threshold of 0.5 and unconstrained parameters in BiMap. For TSMNet, we used standard parameters except for 10 latent SPD dimensions instead of 20. The tangent space projection module used a domain-specific SPDMBN layer without learnable parameters. For fMRI views $\bar{S}$, we use a standard single Transformer-encoder layer as implemented in the torch.

Geodesic Speeds

We fitted the geodesic speeds in a data-driven fashion. Specifically, we treated them as learnable parameters on the Stiefel manifold (random initializations). In addition to $\mathcal{L}_\sigma$, we additionally employed them to project the latent tangent space representations $(\log_e(Z_i), \log_e(\bar{Z}_i))$ to scalars and quantify correlation among these in $\mathcal{L}_\rho$.

Parameter Optimization

We used a minibatch-based training scheme to fit the learnable parameters for 250 epochs. Batches contained 128 paired views (stratified for subject and run). As an optimizer, we used the Adam optimizer with a learning rate of 1e-3 (5e-3

for the simulations experiment) and applied a weight decay 1e-3 to all learnable parameters except the manifold-constrained ones. To put more emphasis on maximizing correlation than fulfilling the geodesic projection constraint, we set $\alpha_1 = 1$, $\alpha_2 = 0.25$, and $\alpha_3 = 0$. To discourage the architectures from collapsing dimensions, we fixed the learnable rescaling parameter of SPDMBN to 1 so that the Fréchet variance of the latent representations remains approximately constant. During training, we used a dedicated validation set to monitor the total loss of held-out data after every epoch. The parameters that resulted in the lowest validation set loss were then used to transform the test data.

This experiment serves as a feasibility study. EEG and fMRI offer distinct perspectives on brain dynamics due to their differing spatiotemporal sensitivities and underlying neurophysiological mechanisms. Previous data-driven approaches have relied heavily on domain knowledge to extract low-dimensional representations [190, 191, 207–210]. Our approach diverges fundamentally by focusing on paired observations to learn latent representations where EEG and fMRI dynamics are highly correlated and maintain this correlation in held-out data. To demonstrate feasibility, we utilized a publicly available dataset [60] that includes simultaneous EEG-fMRI recordings from 8 subjects resting under two conditions (eyes closed and eyes open). Both conditions induce changes in whole-brain dynamics, which are detectable by both EEG [211] and fMRI [212], making this dataset an ideal choice for evaluating our proposed framework against baseline methods.

Figure 9.8 summarizes our experimental approach. We used t-SNE to visualize the preprocessed EEG and fMRI data $\{(S_i, \bar{S}_i)\}_{i=1}^N$ as well as the learned representations $\{(Z_i, \bar{Z}_i)\}_{i=1}^N$ for a representative CV split. For EEG data, we tested two architectures, both extracting latent representations in $\mathcal{S}_{++}^{10}$. The first one was similar to the architecture used in the simulations. The second one (TSMNet) is conceptually similar except that it applies convolutional feature extractors to EEG time-series data before covariance pooling [3, 213]. To accommodate large distribution shifts (e.g., in top-left t-SNE plot in Figure 9.8) across subjects and runs often encountered in EEG [72, 214–216], we used subject and run-specific batch normalization layers [3]. For fMRI data, we used a standard transformer-encoder layer [217], which proved helpful in learning latent representations from large-scale, diverse fMRI datasets [1]. This approach considered each fMRI volume inside a sliding window as a token. At the end of each window, we concatenated a

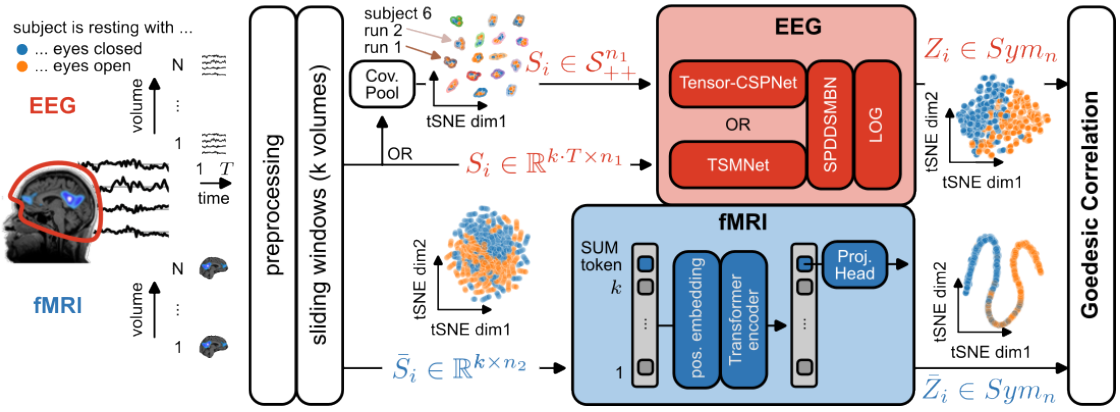


Figure 9.8: DeepGeoCCA for simultaneous EEG-fMRI data. Using simultaneously recorded observations $(S_i, \bar{S}_i)$, we aim to learn a latent space where brain dynamics $(Z_i, \bar{Z}_i)$, shared between EEG and fMRI, covary with high-congruence. After preprocessing and extracting sliding windows, we use established decoder models to convert fMRI time-series activity [1] and oscillatory EEG activity [2, 3] to latent representations. t-SNE visualizations (perplexity=30) summarize input and latent data distributions after model fitting.

learnable summary (SUM) token. Finally, a linear layer projected the transformed SUM token into a symmetric matrix (Sym_{10}) which we used as $\log_e(\bar{Z}_i)$ to minimize Loss 9.9 according to Remark 9.3.

Table 9.2 summarizes the results for a 10-fold cross-validation scenario. We use Pearson correlation to assess the consistency between the learned EEG and fMRI representations and R^2 to quantify the geodesic constraints. The combination of TSMNet and DeepGeoCCA obtained a significantly higher correlation than all other candidates. Learning the considered architectures with alternative loss terms (VICReg, DeepCCA) resulted in clear performance drops for either EEG architecture. Comparing the architectures, our results suggest an advantage of combining learnable convolutional layers with latent covariance pooling over covariance-based inputs. Compared to the performance of RieCCA [4], our results demonstrate the utility of DeepGeoCCA to extract shared, latent brain dynamics whose coupling generalizes to unseen data. Additional results that test generalization across subjects (Table 9.3) show similar effects, confirming the robustness of our framework.

In addition to the 10-fold CV results reported in the main text, this section presents additional results that test generalization across runs (LORO CV scenario) and

[4] To mitigate distribution shifts in the EEG data for the conventional baseline methods (CCA, RieCCA), we debiased the covariance matrices per run [218].

Table 9.2: Simultaneous EEG-fMRI dataset results. As before, test-set model performance (higher is better) is evaluated with correlation and R^2 metrics across 10-fold CV with stratification across subjects and runs. Exhaustive permutation t-tests (df=9, 7 tests with t-max adjustment) were used to identify significant differences between *TSMNet+DeepGeoCCA* and baseline methods.

fMRI model	**EEG** model	**algorithm**	**correlation** ↑ mean (std)	t-val (p-val)	**R^2** ↑ mean (std)	t-val (p-val)
-	-	CCA	0.02 (0.15)	-9.9 (0.002)	0.07 (0.08)	-19.8 (0.002)
		RieCCA	0.38 (0.10)	-5.4 (0.002)	0.01 (0.01)	-88.4 (0.002)
Trans-former	Tensor-CSPNet	DeepCCA	0.42 (0.11)	-3.5 (0.031)	0.30 (0.07)	-12.3 (0.002)
		VICReg	0.29 (0.19)	-4.0 (0.018)	0.28 (0.10)	-9.2 (0.002)
		DeepGeoCCA	0.44 (0.11)	-3.0 (0.049)	**0.58** (0.02)	3.8 (0.023)
	TSMNet	DeepCCA	0.29 (0.10)	-8.8 (0.002)	0.18 (0.09)	-15.2 (0.002)
		VICReg	0.38 (0.10)	-5.5 (0.002)	0.24 (0.03)	-40.5 (0.002)
		DeepGeoCCA	**0.58** (0.08)	-	0.55 (0.02)	-

subjects (LOSO CV scenario). The results are summarized in Table 9.3 and Figure 9.9, along with the 10-fold CV results provided in the main text. Comparing the results, we noticed large variability for the LORO and LOSO evaluation scenarios. All models failed to generalize, particularly for subjects 1 and 5 (highlighted with red dots in Figure 9.9). We consulted the accompanying publication [59] to understand why this is the case. We identified that for these subjects, there was no task effect (eyes open vs. closed) in the EEG data. Since the effect is prominent in the other subjects, naturally, models that utilize the "majority" effect will not generalize to these particular subjects. That is why we report the results in Table 9.3 with (w/) and without (w/o) test-set folds associated with subjects 1 and 5. Apart from the large variability (mostly due to subjects 1 and 5), we found that the overall effects identified in the 10-fold CV scenario are preserved in the other evaluation scenarios. If we exclude the results for subjects 1 and 5, the average correlation score for all methods increases drastically (approx. 0.2). Particularly, the combination of DeepGeoCCA and TSMNet yields correlations close to 0.7 on held-out data. These additional results clearly demonstrate that our proposed framework can extract representations that generalize to new subjects and pave the way for potential clinical applications with sufficiently large and diverse datasets.

Table 9.3: Simultaneous EEG-fMRI dataset results. Additional test-set results for the EEG-fMRI dataset that extend Table 9.2 with regard to the considered evaluation scenario. In addition to the 10-fold CV scenario results reported in Table 9.2, this table also summarizes results for leave-one-run-out (LORO) and leave-one-subject-out (LOSO) CV scenarios. For LORO and LOSO, we report summary statistics (mean and std) with (w/) and without (w/o) outlier subjects (i.e., subjects 1 and 5).

	evaluation:	10-fold	LORO		LOSO	
EEG model	**outlier:** **algorithm**	w/	w/	w/o	w/	w/o
		correlation ↑				
none	CCA	0.02 (0.15)	0.18 (0.35)	0.34 (0.21)	0.13 (0.30)	0.27 (0.17)
	RieCCA	0.38 (0.10)	0.37 (0.44)	0.59 (0.13)	0.24 (0.39)	0.45 (0.10)
Tensor-	DeepCCA	0.42 (0.11)	0.36 (0.43)	0.56 (0.23)	0.30 (0.32)	0.45 (0.16)
CSPNet	VICReg	0.29 (0.19)	0.24 (0.43)	0.46 (0.21)	0.31 (0.28)	0.43 (0.19)
	DeepGeoCCA	0.44 (0.11)	0.39 (0.45)	0.58 (0.27)	0.24 (0.47)	0.44 (0.35)
TSMNet	DeepCCA	0.29 (0.10)	0.17 (0.34)	0.25 (0.32)	0.12 (0.46)	0.30 (0.35)
	VICReg	0.38 (0.10)	0.42 (0.36)	0.58 (0.22)	0.25 (0.37)	0.42 (0.19)
	DeepGeoCCA	0.58 (0.08)	0.46 (0.45)	0.68 (0.18)	0.45 (0.43)	0.68 (0.14)
		R2 ↑				
none	CCA	0.07 (0.08)	-2.32 (1.29)	-2.26 (1.31)	-1.05 (0.70)	-1.19 (0.76)
	RieCCA	0.01 (0.01)	0.00 (0.02)	-0.00 (0.02)	0.01 (0.01)	0.01 (0.01)
Tensor-	DeepCCA	0.30 (0.07)	0.27 (0.13)	0.28 (0.13)	0.23 (0.10)	0.24 (0.12)
CSPNet	VICReg	0.28 (0.10)	0.18 (0.12)	0.20 (0.14)	0.23 (0.07)	0.24 (0.07)
	DeepGeoCCA	0.58 (0.02)	0.56 (0.05)	0.57 (0.05)	0.54 (0.07)	0.54 (0.08)
TSMNet	DeepCCA	0.18 (0.09)	0.10 (0.06)	0.11 (0.05)	0.13 (0.04)	0.13 (0.05)
	VICReg	0.24 (0.03)	0.22 (0.11)	0.23 (0.10)	0.22 (0.09)	0.22 (0.09)
	DeepGeoCCA	0.55 (0.02)	0.53 (0.06)	0.54 (0.06)	0.54 (0.02)	0.54 (0.02)

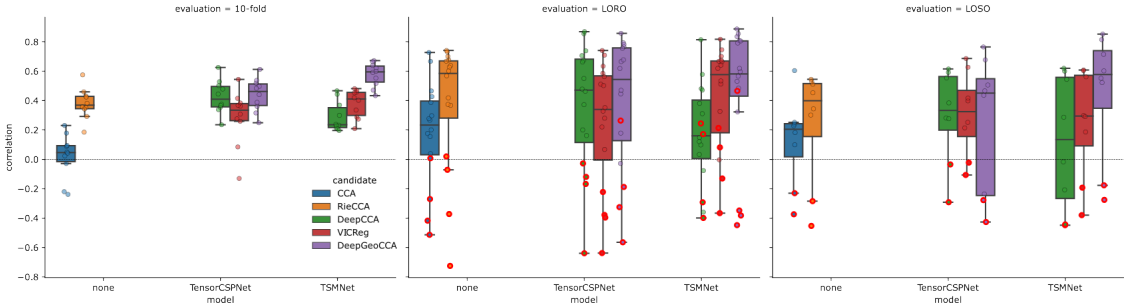


Figure 9.9: Graphical representation of the results summarized in Table 9.3 for the correlation metric on test set data. Each dot summarizes a CV split result. (**left**) 10-fold CV scenario. (**middle**) LORO scenario (16 runs=folds). Red dots highlight fold for which the test set was sampled from outlier subjects (subjects 1 and 5). (**right**) LOSO scenario (8 subjects = folds).

Specific challenges associated with neuroimaging data

For the neuroimaging tasks that we study, we have to deal with relatively small datasets (recording human neuroimaging data is expensive and time-consuming),

non-stationary signals, and distribution shifts that limit generalization across days and subjects (also indicated in Figure 9.8), inherently low signal-to-noise ratios, transient high-variance artifacts (i.e., outliers), and non-linear, poorly understood coupling mechanisms that link views of different modalities [219]. For more details, let us refer to recent reviews elaborating not only on challenges but also opportunities of EEG BCI [220] and simultaneous EEG-fMRI [186].

9.3.3 Multi-View EEG

Korean University Dataset

The following two-view DeepGeoSSL model is employed:

- 1^{st}-View: Two electrodes (C3 and C4);
- 2^{nd}-View: Twenty electrodes (FC-5/3/1/2/4/6, C-5/3/1/z/2/4/5, and CP-5/3/1/z/2/4/6).

All 21,600 trials are utilized in training the self-supervised learning framework.

For the downstream classification task, we use a tensor-based approach for segmentation in time and frequency domains [2]. In the time domain, there are three segments as follows:

$$[0 \sim 1.5],\ [0.5 \sim 2],\ \text{ and } [1 \sim 2.5].$$

The frequency band has been partitioned into 9 segments, each spanning a 4Hz bandwidth. These segments cover the range from 4Hz to 40Hz and are evenly distributed within this range. Chebyshev Type II filters with 4 Hz intervals were employed to process the digital signals. These filters were meticulously designed to ensure a maximum passband loss of 3 dB and a minimum stopband attenuation of 30 dB.

It results in 27 channels (3 temporal segments $\times$ 9 frequency bands). Additionally, the BiMap layer in the architecture takes the input dimension of 20, transforms it to 30, and then returns it to an output dimension of 20. The architecture details can be found in Figure 9.10.

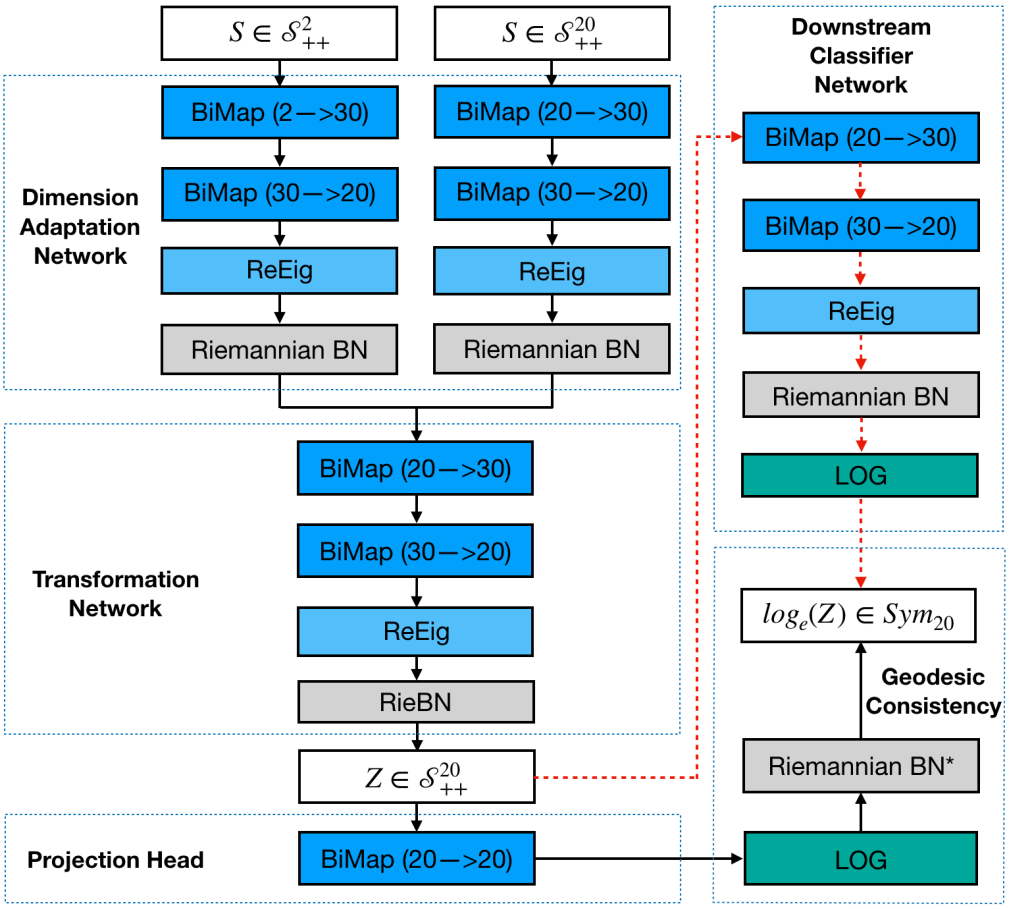


Figure 9.10: Illustration of network architecture for the KU dataset: The spatial covariance matrices from both the 2-channel EEG setup (i.e., C3 and C4) and the 20-channel EEG setup (i.e., FC-5/3/1/2/4/6, C-5/3/1/z/2/4/5, and CP-5/3/1/z/2/4/6) are transformed into a common dimension using the Dimension Adaptation Network. During the training of the self-supervised learning (indicated by the solid black lines), the Transformation Network and Projection Head retain the parameters in the neural networks, and geodesic correlation is calculated. The dashed red lines represent the downstream task classifier. Specifically, in the upper right corner of Riemannian BN*, the asterisk $*$ denotes that we use a specially designed Riemannian BN to perform center-parallel translation of outputs to point e.

BNCI2015001 Dataset

The following two-view DeepGeoSSL model is employed:

- 1^{st}-View: Two electrodes (FCz and C3);
- 2^{nd}-View: Twenty electrodes (FC-3/z/4, C-5/3/1/z/2/4/6, and CP-3/z/4).

All 5,600 trials are utilized in training the self-supervised learning framework.

In the time domain, there are three segments as follows:

$$[0 \sim 1],\ [1 \sim 2],\ [2 \sim 3],\ [3 \sim 4],\ \text{ and } [4 \sim 5].$$

The frequency division was performed at the same intervals and filter settings as in the KU dataset, resulting in 45 channels (5 temporal segments $\times$ 9 frequency bands). Additionally, the BiMap layer in the architecture takes the input dimension of 13, transforms it to 30, and then returns it to an output dimension of 13. The architecture details can be found in Figure 9.11.

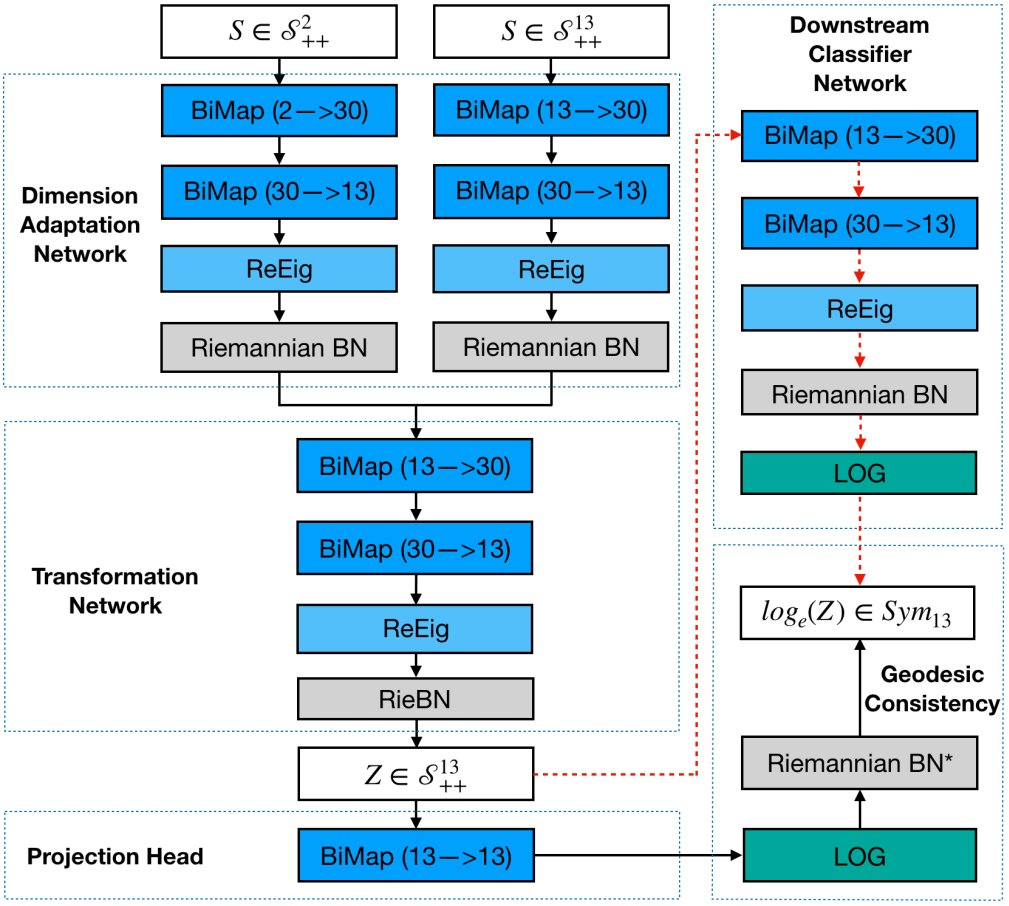


Figure 9.11: Illustration of network architecture for the BNCI2015001 dataset: The spatial covariance matrices for the 2-channel EEG setup are derived from FCz and C3, while for the 20-channel EEG setup, they are derived from FC-3/z/4, C-5/3/1/z/2/4/6, and CP-3/z/4. The description of this architecture is consistent with it provided in Figure 9.10.

Geodesic Speed

Since the downstream motor imagery tasks emphasize spectral information for classification, we utilize the largest principal component from Tangent Principle Component Analysis (Tangent PCA) [5] as the initial speed for both v and w as follows,

$$v = w := \text{TPCA}(\{vec(\log_e(\bar{S}_i))\}_{i=1}^{N}),$$

where $\{\bar{S}_i\}_{i=1}^{N}$ represents a set of 20-channel EEG signals. In particular, we compute $v = w$ using Tangent PCA for each channel of 27 channels for the KU dataset and 45 channels for the BNCI2015001 dataset.

The choice of this speed can be understood as both modalities projecting signals onto high-information-content spectral principal component directions and keeping the two signals geodesically consistent. This is akin to neural networks augmenting low-information-content signals with sufficient information.

Estimation of Covariance Matrices

It is well-known that the estimation of the sample covariance matrix can be inaccurate, particularly in high-dimensional settings with limited observations. To address this challenge, we employ a regularization technique known as shrinkage, which involves adding a scaled identity matrix (typically multiplied by a small value, such as 1e-2 in this experiment) to the sample covariance matrix.

Average Accuracy Changes

In this experiment, two figures present the average accuracy changes for each participant across six scenarios in two datasets. The baseline is the result obtained by running the 2-channel EEGs using Tensor-CSPNet. The changes are calculated by subtracting the baseline results from DeepGeoCCA ($\varepsilon = 0.05$, $\alpha_1 = 1, \alpha_2 = 1$, $\alpha_3 = 0.1$), as shown in Figures 9.12 and 9.13.

[5] We implemented the initialization of v and w using the Tangent PCA approach according to the Geomstats Package in the following address: `https://geomstats.github.io/notebooks/06_practical_methods__riemannian_frechet_mean_and_tangent_pca.html`.

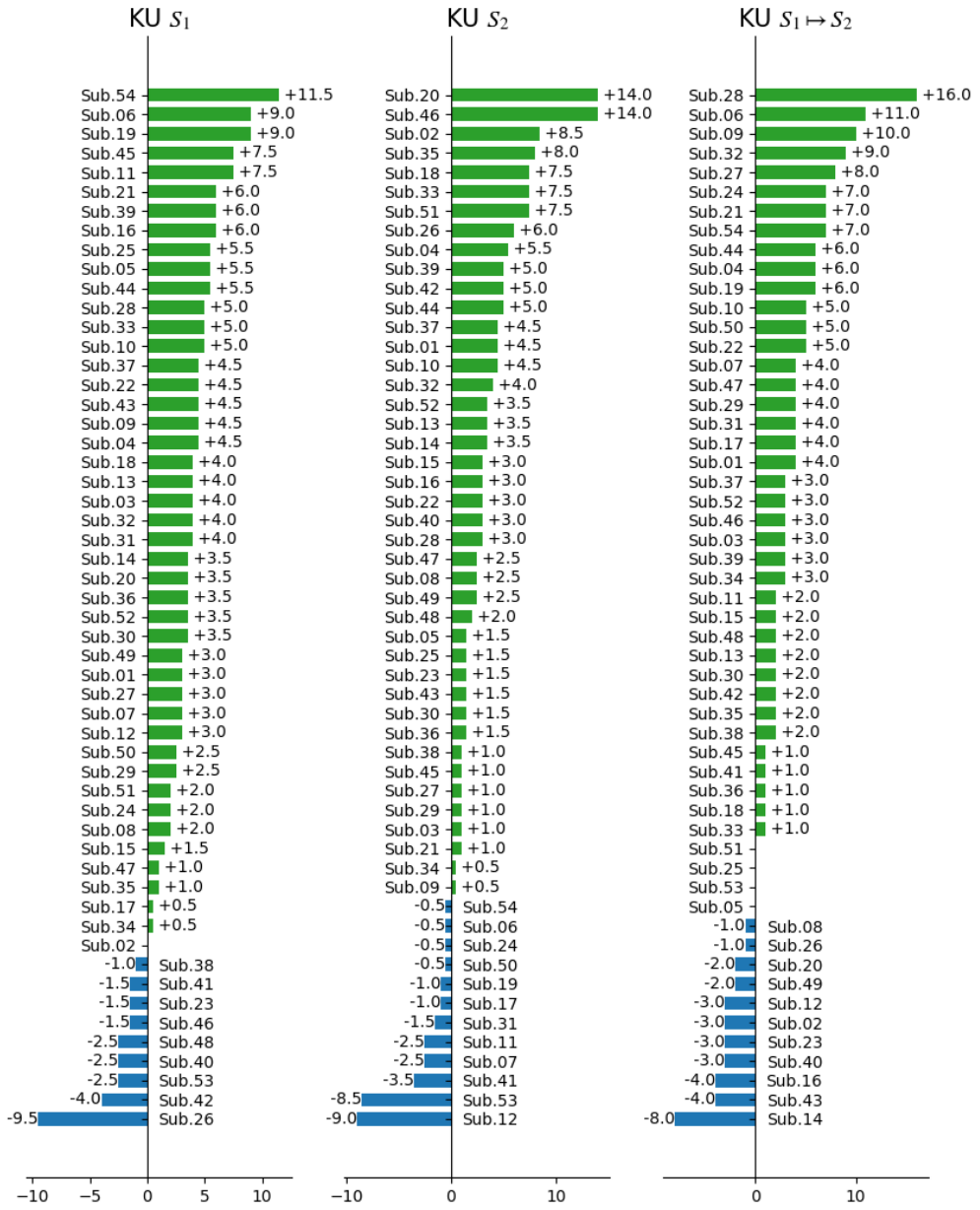


Figure 9.12: This figure displays the average accuracy changes (%) for each subject in the KU dataset across two 10-fold cross-validation scenarios S_1 and S_2, and a holdout scenario $S_1 \mapsto S_2$. Subjects with larger improvements are positioned at the top, arranged in descending order from top to bottom. Green represents an increase, while blue represents a decrease. Corresponding increases are labeled with the respective subject number.

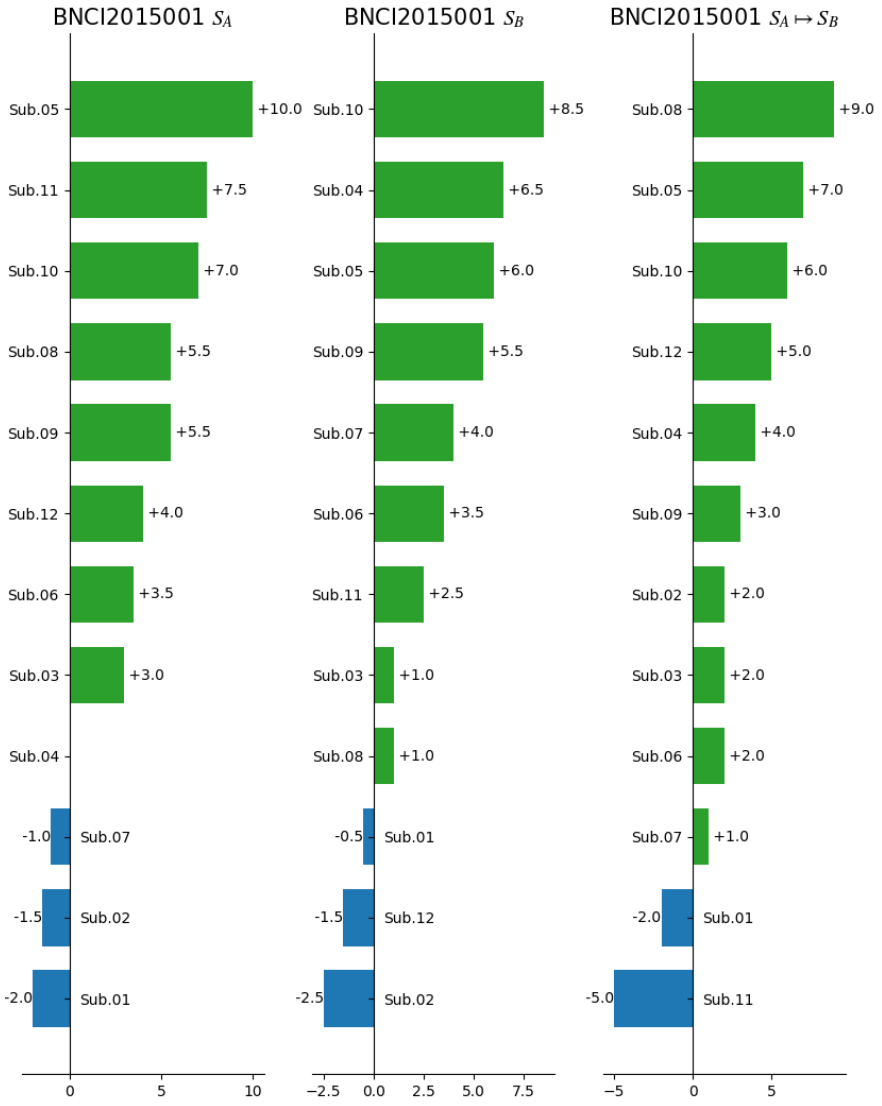


Figure 9.13: This figure displays the average accuracy changes (%) for each subject in the BNCI2015001 dataset across two 10-fold cross-validation scenarios S_A and S_B, and a holdout scenario $S_A \mapsto S_B$. Subjects with larger improvements are positioned at the top, arranged in descending order from top to bottom. Green represents an increase, while blue represents a decrease. Corresponding increases are labeled with the respective subject number.

Table 9.4: Multi-View EEG downstream results. Average (std across subjects in brackets) classification accuracy for the motor imagery task (higher is better). We considered two public datasets and scenarios: within-session 10-fold CV (sessions S_1 and S_2) and a hold-out scenario (train S_1; test S_2). For the KU and BNCI2015001 datasets, the full-channel EEG views comprise 20 and 13 channels covering the sensorimotor cortex. The network architecture was Tensor-CSPNet.

Channel	SSL-Loss	KU (54 subjects, 2 classes)			BNCI2015001 (12 subjects, 2 classes)		
		S_1	S_2	$S_1 \rightarrow S_2$	S_A	S_B	$S_A \rightarrow S_B$
2	Lower Bound	62.97 (15.72)	65.85 (16.03)	62.89 (12.18)	74.12 (18.20)	75.62 (14.22)	75.00 (14.10)
	VICReg	66.12 (15.80)	67.35 (16.80)	65.20 (13.51)	77.21 (15.96)	77.25 (13.46)	77.00 (13.97)
	DeepCCA	66.24 (15.73)	68.11 (15.67)	65.20 (13.24)	76.46 (16.83)	78.33 (11.33)	76.58 (13.23)
	DeepGeoCCA	**66.26** (17.05)	**68.27** (17.23)	**65.93** (13.68)	**77.58** (16.26)	**78.46** (12.82)	**77.83** (13.11)
Full	Upper Bound	70.56 (16.26)	72.86 (15.86)	68.85 (14.18)	84.33 (10.10)	86.46 (14.19)	85.17 (11.31)

Table 9.4 summarizes the multi-view EEG downstream results. Compared to the lower bound (using 2 EEG channels with no SSL pre-training), pre-training with DeepGeoCCA led to statistically significant group-level improvements in both datasets presented in Table 9.5. Although our proposed loss function consistently achieved significant improvements, the effect was relatively modest across scenarios when compared to other SSL pre-training losses. Nonetheless, these results demonstrate that our framework, which pre-trains SPD neural networks such as Tensor-CSPNet with paired views, can enhance performance in downstream tasks.

Table 9.5 additionally summarizes the significance of these differences. In these figures, green areas indicate the subject's improvements, while blue areas represent those with declines. Our method is effective for a large proportion of subjects in all scenarios.

This experiment aims to evaluate the performance of DeepGeoCCA in a downstream task, namely EEG-based motor imagery classification. During planning and executing but also imagining limb movements, EEG captures spatio-spectrally localized fluctuations in sensorimotor rhythms. These fluctuations give rise to identifiable patterns that consistently enable the classification of EEG data.

Motivated by the emergence of mobile, few-channel EEG devices, we generate pairs of 2-channel EEG and multi-channel EEG views, i.e., full sensorimotor coverage. Like standard SSL, we use DeepGeoCCA for model pre-training to improve the downstream classification performance of the 2-channel EEG views. Here, we fixed geodesic speeds $v = w$ and initialized them with tangent space principal component vectors of the EEG spatial covariance matrices of the full-channel EEG views.

Table 9.5: Simultaneous EEG-fMRI dataset results. Statistical test results for downstream test set accuracy differences in the multi-view EEG experiment, extending Table 9.4. We used permutation, paired t-tests to identify significant differences between the lower bound (i.e., no pre-training; 2 channel EEG) with DeepGeoCCA for three scenarios (S_1, S_2, $S_1 \rightarrow S_2$; t-max adjustment for multiple comparisons) and the KU dataset (df=53, 1e4 permutations) and the BNCI dataset (df=11, exhaustive permutations). Significant differences are highlighted in bold (p-val ≤ 0.05) and trends in italic (p-val ≤ 0.1).

		KU dataset		**BNCI dataset**	
scenario	**SSL pretraining**	mean (std)	t-val (p-val)	mean (std)	t-val (p-val)
S_1	no	62.97 (12.46)	-6.1 (0.0001)	74.12 (16.50)	-3.1 (0.0391)
	DeepGeoCCA	**65.90** (12.95)	-	**77.58** (14.56)	-
S_2	no	65.85 (12.95)	-4.0 (0.0006)	75.62 (14.58)	-2.9 (0.0469)
	DeepGeoCCA	**68.11** (13.64)	-	**78.46** (13.95)	-
$S_1 \rightarrow S_2$	no	62.89 (12.29)	-4.4 (0.0003)	75.00 (14.73)	-2.6 (0.0825)
	DeepGeoCCA	**65.43** (13.42)	-	*77.83* (13.70)	-

Ablation Study

The loss function of DeepGeoCCA has a total of four hyperparameters: ε for the ε-geodesic constraints, and three coefficients α_1, α_2, and α_3 for the three loss

functions. In the following two ablation studies, we analyze these parameters:

In the first experiment, the model's overall performance does not vary significantly with different values of ε. Smaller values of ε lead to a slight improvement in classification performance. The results reported in the main text are based on the setting where $\varepsilon = 0.05$. In the second experiment, we observed that using the variance loss slightly improved the performance compared to not using it. Therefore, in the results reported in the main text, we used the variance loss with a coefficient of $\alpha = 0.1$.

$\mathcal{L}_\varepsilon$	S_1	S_2
Lower Bound	62.97 (15.72)	65.85 (16.03)
$\varepsilon = 0.9$	65.94 (16.10)	68.19 (16.02)
$\varepsilon = 0.7$	65.82 (15.35)	67.77 (16.04)
$\varepsilon = 0.5$	66.23 (16.10)	68.06 (16.66)
$\varepsilon = 0.3$	66.14 (15.93)	68.09 (16.07)
$\varepsilon = 0.1$	66.03 (15.56)	68.13 (15.87)
$\varepsilon = 0.05$	**66.26** (17.05)	**68.27** (17.23)
$\varepsilon = 0$	66.32 (16.40)	68.13 (16.55)

(a) $\mathcal{L}_\varepsilon$ in $\mathcal{L}$: $\alpha_1 = \alpha_2 = 1$, $\alpha_3 = 0.1$.

$\mathcal{L}_\sigma$	S_1	S_2
Lower Bound	62.97 (15.72)	65.85 (16.03)
$\alpha_3 = 0$	65.74 (16.59)	67.33 (16.34)
$\alpha_3 = 0.1$	**66.26** (17.05)	**68.27** (17.23)

(b) $\mathcal{L}_\sigma$ in $\mathcal{L}$: $\alpha_1 = \alpha_2 = 1$, $\varepsilon = 0.05$.

Table 9.6: Ablation Studies: Average (std across subjects in brackets) classification accuracy for the motor imagery task (higher is better). We considered the KU dataset with scenario 10-fold CV (sessions S_1 and S_2).

9.4 Discussions

Motivated by widespread applications of covariance-based data in uni- and multimodal neuroimaging, we capture the correlation consistency of paired views on SPD manifolds using a novel measure called geodesic correlation. geodesic correlation is a generalized concept of correlation for SPD manifold-valued data. It is well suited for geometric deep learning with SPD manifold-valued data compared to conventional and Riemannian correlations.

To maximize geodesic correlation in a latent space between paired views of SPD manifold-valued data, we propose a novel loss function using the relaxed geodesic constraints and present a novel geometric deep learning framework, referred to as DeepGeoCCA. DeepGeoCCA is more generally applicable than previous classical non-deep learning approaches because, in a general sense, deep learning is capable of handling larger datasets with intricate high-dimensional data distributions or

aligning latent representations of neural data collected from a diverse population. Unlike conventional deep learning, our framework respects the geometric structure of the SPD manifold-valued data. In simulations and experiments with EEG and fMRI data, we found that DeepGeoCCA can learn representations whose dynamics generalize favorably to held-out data while preserving task-relevant information.

In Table 9.7, we analyze the similarities and differences between several related variants of the CCA method and our proposed method from three perspectives, including:

- Nonlinear Projection: Projections on the nonlinear subspace or not.
- Eigendecomposition: Utilizing the eigendecomposition method or not, i.e., eigendecomposition for Frobenius Norm function $\left\|\sum_{11}^{-1/2}\sum_{12}\sum_{22}^{-1/2}\right\|_{\mathcal{F}}$.
- Neural Networks-Based Approach: the neural networks-based approach or not.
- Geometry: SPD matrix-valued outputs or not.

Table 9.7: Comparisons between different CCAs.

Methodology	Nonlinear Projection	Eigendecomposition	Neural Networks	Geometry
CCA	×	✓	×	×
DeepCCA	✓	✓	✓	×
RieCCA	✓	✓	×	✓
DeepGeoCCA	✓	×	✓	✓

We view this work as a proof-of-concept demonstrating that geometric deep learning can effectively extract meaningful representations from simultaneous EEG-fMRI data obtained from paired observations. Although our experiments with EEG-fMRI data were limited to a small dataset featuring well-known task-related effects in either modality, this work represents the first fully data-driven SSL approach in this context. For applications beyond the tasks considered, it will be necessary to identify relevant factors of variation to generate paired views. Naturally, views associated with the same subject could be treated as paired. For instance, our framework could facilitate the relationship between functional connectomes, derived from fMRI, and anatomical connectomes, derived from diffusion tensor imaging, to identify latent, potentially clinically relevant modes of variation across subjects.

Part III

Conclusions

Chapter 10

Conclusions

10.1 Summary and Significance

In this thesis, we present and advance innovative EEG-based motor imagery classifiers, termed geometric classifiers. These classifiers utilize neural networks on SPD manifolds to derive discriminative features from EEG spatial covariance matrices. To address the challenges of cross-session variability, the classifiers integrate an optimal transport-domain adaptation framework. Furthermore, we explore cutting-edge generative models for synthesizing EEG spatial covariance matrices, which enhances the accuracy of the geometric classifiers. We also propose a novel method for integrating multimodal neuroimaging data, such as simultaneous EEG-fMRI, via a geodesic correlation approach. This multimodal fusion, implemented within a self-supervised framework, significantly boosts the performance of the geometric classifiers in downstream tasks.

The significance of this study lies in its shift from traditional CNN-based methods for EEG motor imagery classification to a geometric deep learning approach. This novel approach builds on concepts related to conventional CSP-based classifiers. The proposed classifiers, such as Tensor-CSPNet and Graph-CSPNet, achieve performance on par with leading classifiers across various public datasets, demonstrating their practical effectiveness. More importantly, the integration of Riemannian geometry with these methods links them to modern mathematics, statistics, and physics, providing new insights and tools for brain-computer interfacing beyond

the deep learning paradigm. Consequently, this work bridges the gap between artificial intelligence and applied mathematics.

10.2 Future Directions

Several practical improvements are needed along this pathway, including, but not limited to, the following:

- Computational Limitations: The foundational architecture of this study is currently optimized for small-scale data and a limited number of device channels. To improve computational efficiency, it is necessary to optimize these network structures from a computational perspective.
- Scenario-Specific Adaptation: The methods proposed in this research are generally applicable but require further optimization for specific scenarios. These scenarios are technically more complex and involve additional engineering considerations, necessitating structural modifications to the models to better accommodate these specific environments.
- Scalability: The neural network architectures proposed in this study differ from state-of-the-art deep learning solutions, such as large language models. Investigating the scalability of these architectures in comparison to large language models could be crucial for enhancing their adaptability to broader applications.

Part IV

Appendices

Appendix A

Diffusion Model

This appendix aims to provide a brief introduction to the algorithm for score-based generative modeling. For a formal convention, we suppose we have samples of spatial covariance matrices $\{S_i\}_{i=1}^{N} \in \mathbb{R}^{n_C \times n_C}$ from an unknown distribution $p_{data}(S)$.

A.1 Score-Based Generative Modeling

In the score-based generative modeling [175, 176], the score network $s_\theta : \mathbb{R}^{n_C \times n_C} \longmapsto \mathbb{R}^{n_C \times n_C}$ is a deep neural network parametrized by θ and used to learn the score of a probability density $\nabla_S \log p(S)$.

To train score network s_θ, a technique called *denoising score matching* [221, 222] is proposed to firstly replace $p_{data}(S)$ using a Gaussian noise σ-pertubed version $p^{\sigma}_{data}(\tilde{S})$, where $p^{\sigma}_{data}(\tilde{S}) = \int p^{\sigma}_{\mathcal{N}}(\tilde{S}|S) p_{data}(S)\, dS$, and the denoising objective $\mathcal{J}_D(\theta)$ with noise level σ is then given as follows:

$$\mathcal{J}^{\sigma}_{D}(\theta) := \mathbb{E}_{p^{\sigma}_{\mathcal{N}}(\tilde{S}|S) p_{data}(S)} ||s_\theta(\tilde{S}) - \nabla_{\tilde{S}} \log p^{\sigma}_{\mathcal{N}}(\tilde{S}|S)||.$$

Keep in mind that the noise model term $\nabla_{\tilde{S}} \log p^{\sigma}_{\mathcal{N}}(\tilde{S}|S)$ has a simple analytic form, written $\nabla_{\tilde{S}} \log p^{\sigma}_{\mathcal{N}}(\tilde{S}|S) = (S - \tilde{S})/\sigma^2$.

In the sampling phase, Langevin dynamics are applied to recursively generate samples using the score function s_θ, as follows,

$$\tilde{S}_t = \tilde{S}_{t-1} + \frac{\epsilon}{2} s_\theta(\tilde{S}_{t-1}) + \sqrt{\epsilon} Z_t,$$

where initial $\tilde{S}_0 \sim \pi(S)$ (prior distribution) and fixed step size $\epsilon > 0$ are given, and $Z_t \in \mathcal{N}(0, I)$.

A.2 Diffusing Samples through Stochastic Diffusion Equations

The integration of score-based generative modeling with diffusion probabilistic modeling has provided a unified framework for capturing the evolution of a continuum of distributions over time t through Stochastic Diffusion Equations (SDEs) [177].

Formally, an SDE is of the form by:

$$dS_t = f(S_t, t)\ dt + g(S_t, t)\ dW_t,$$

where $f(s, t)$ and $g(s, t)$ are drift coefficient and diffusion coefficient, respectively. W_t is a standard Wiener process or Brownian motion. If the noise coefficient vanishes, the above SDE becomes an ordinary differential equation $dS_t = f(S_t, t)dt$.

The objective of score-based generative modeling is to construct a diffusion process $\{S_t \in \mathbb{R}^{n_C \times n_C}\}_{t \in [0,T]}$, with t being a continuous time variable in the interval $[0, T]$, such that $S_0 \sim p_0(S)$ and $S_T \sim p_T(S)$, where p_0 (i.e., p_{data} in Section A.1) represents the data distribution and p_T is the prior distribution. Under the assumption that the diffusion coefficient is independent of the state variable S_t, the score-based generative modeling involves the following Itô SDE:

$$dS_t = f(X_t, t)\ dt + g(t)\ dW_t,$$

where $f(\ \cdot\ , t) : \mathbb{R}^{n_C \times n_C} \longmapsto \mathbb{R}^{n_C \times n_C}$ and $g(t) \in \mathbb{R}$, and $W_t \in \mathbb{R}^{n_C \times n_C}$ is a Wiener process.

To generate samples, the score-based generative modeling follows a time-reversal diffusion process by starting from samples of $S_T \sim p_T(S)$ and reversing the process to obtain a sample $S_0 \sim p_0(S)$. This time-reversal diffusion process is implemented using the following steps:

$$dS_t = \big(f(S_t, t) - g(t)^2 \nabla_S \log p_t(S_t)\big)\ dt + g(t)\ d\bar{W}_t,$$

where p_t is the marginal distribution of S_t, and $\bar{W}_t$ is a standard Wiener process in the reverse-time direction. Marginal distribution $\nabla_S \log p_t(S)$ for all time t is usually unknown.

The approach to achieve $\nabla_S \log p_t(S)$ is the above *denoising score matching* in Section A.1 utilizing a time-dependent neural network $s_\theta(S, t)$ to estimate $\nabla_S \log p_t(S)$ by squeezing the loss $\mathcal{J}_D\big(\theta; \lambda\big)$, as follows,

$$\int_0^T \mathbb{E}_{p_{0t}(S_t|S_0)\times p_0(S_0)} \lambda(t) || s_\theta(S_t, t) - \nabla_{\tilde{S}} \log p_{0t}(S_t|S_0) ||\ dt,$$

where $p_{0t}(S_t|S_0)$ is the transition distribution from S_0 to S_t, and $\lambda : [0, T] \to \mathbb{R}_{>0}$ is a positive weighting function. Therefore, by deriving the reverse diffusion process and sampling from p_0 according to the reverse-time SDE, we can obtain samples of $S_0 \sim p_0(S)$ from the samples of $S_T \sim p_T(S)$.

Appendix B

Canonical Correlation Analysis

We will briefly introduce canonical correlation analysis [187], a deep learning-based canonical correlation analysis extension [201], and a generalized canonical correlation analysis method on general Riemannian manifolds [193].

Canonical correlation analysis (CCA) is a statistical method that identifies links between variable sets from different modalities [187]. Applications of CCA to neuroimaging data have been widely utilized in contemporary cognitive neuroscience; please refer to [223] for further details.

B.1 Canonical Correlation Analysis

Let two paired matrices of data distributions be $X \in \mathbb{R}^{N \times p}$ and $\bar{X} \in \mathbb{R}^{N \times q}$, where N represents the number of observations, while p and q denote the dimensions of two variables, respectively. *Paired* means that each i^{th} row in both X and $\bar{X}$ are coupled (e.g., recorded at the same time). We assume the columns in X and $\bar{X}$ are zero-mean.

CCA aims to find two linear transformations, denoted as $w \in \mathbb{R}^p$ and $\bar{w} \in \mathbb{R}^q$, in such a way that the cosine of the angle Θ between the projected vectors $z^\dagger$ and $\bar{z}^\dagger$ onto the unit ball is maximized, as described below,

$$\cos\Theta := \max_{z^\dagger, \bar{z}^\dagger \in \mathbb{R}^N} \langle\, z^\dagger, \bar{z}^\dagger \,\rangle,$$

where $z^\dagger := (Xw)/\left\|Xw\right\| \in \mathbb{R}^N$, and $\bar{z}^\dagger := (\bar{X}\bar{w})/\left\|\bar{X}\bar{w}\right\| \in \mathbb{R}^N$.

B.2 Deep Canonical Correlation Analysis

Deep canonical correlation analysis (DeepCCA) employs neural networks to learn latent representations with the goal maximize the CCA objective between latent representations of two modalities, as follows,

$$\max_{\theta,\bar{\theta}} \operatorname{corr}\big(\varphi(x;\theta), \bar{\varphi}(\bar{x};\bar{\theta})\big),$$

where $\varphi(\,\cdot\,;\theta)$ and $\bar{\varphi}(\,\cdot\,;\bar{\theta})$ are neural networks with parameters θ and $\bar{\theta}$, respectively.

Inspired by CCA, DeepCCA is proposed to compute and maximize the following matrix multiplication term in the loss function:

$$\mathcal{L} := \left\| \sum_{\varphi\varphi}^{-1/2} \sum_{\varphi\bar{\varphi}} \sum_{\bar{\varphi}\bar{\varphi}}^{-1/2} \right\|_{\mathcal{F}},$$

where $\sum_{\varphi\varphi}$ and $\sum_{\bar{\varphi}\bar{\varphi}}$ represent within-modality covariances of latent representation, and $\sum_{\varphi\bar{\varphi}}$ represents cross-modality covariance of latent representations. Their preprocessing procedure includes centering and regularization.

B.3 Riemannian Canonical Correlation Analysis

Given a set of paired manifold-valued data $\{x_i\}_{i=1}^N$ and $\{\bar{x}_i\}_{i=1}^N \in (\mathcal{M}, g)$, Riemannian canonical correlation analysis (RieCCA) seeks to maximize the Riemannian correlation defined as follows,

$$\cos\Theta := \langle z/\left\|z\right\|, \bar{z}/\left\|\bar{z}\right\| \rangle_{\ell^2},$$

where z and $\bar{x}$ are the logrithm of orthogonal projections of $\{x_i\}_{i=1}^N$ and $\{\bar{x}_i\}_{i=1}^N$ onto two geodesic submanifolds $\exp_{\mu_x}(U_x), \exp_{\mu_{\bar{x}}}(\bar{U}_{\bar{x}}) \subset \mathcal{M}$, respectively.

Each coordinate of z and $\bar{z}$ are $z_i := \left\|\log_{\mu_x}(x_i^\dagger)\right\|_{\mu_x}$ and $\bar{z}_i := \left\|\log_{\mu_{\bar{x}}}(\bar{x}_i^\dagger)\right\|_{\mu_{\bar{x}}}$, respectively, and open neighborhoods $U_x \subset \mathcal{T}_{\mu_x}\mathcal{M}$ and $\bar{U}_{\bar{x}} \subset \mathcal{T}_{\mu_{\bar{x}}}\mathcal{M}$ are linearly combined by tangent vectors at Fréchet means μ_x and $\mu_{\bar{x}}$, respectively.

In particular, orthogonal projections on the geodesic submanifolds are defined as follows

$$x_i^\dagger := \min_{x\in\mathcal{M}} d_g^2(x_i, \exp_{\mu_x}(U_x)), \text{ and } \bar{x}_i^\dagger := \min_{\bar{x}\in\mathcal{M}} d_g^2(\bar{x}_i, \exp_{\mu_{\bar{x}}}(\bar{U}_{\bar{x}})),$$

for all $i \in \{1, ..., N\}$.

List of Author's Publications

Journal Articles

Ce Ju and Cuntai Guan, Tensor-Cspnet: A Novel Geometric Deep Learning Framework for Motor Imagery Classification, IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 12, pp. 10955-10969, Dec. 2023.

Ce Ju and Cuntai Guan, Graph Neural Networks on SPD Manifolds for Motor Imagery Classification: A Perspective from the Time-Frequency Analysis, IEEE Transactions on Neural Networks and Learning Systems, 2023 [1].

Conference Proceedings

Ce Ju, Reinmar Josef Kobler, and Cuntai Guan, Score-Based Data Generation for EEG Spatial Covariance Matrices: Towards Boosting BCI Performance, the 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society, 2023. (IEEE EMBC 2023, Oral)

Ce Ju†, Reinmar Josef Kobler†, Liyao Tang, Cuntai Guan, and Motoaki Kawanabe, Deep Geodesic Canonical Correlation Analysis for Covariance-Based Neuroimaging Data, the twelfth International Conference on Learning Representations, 2024. (ICLR 2024, Spotlight)

[1] This is an early access article in IEEE Xplore, available in an electronic archive before its appearance in a regular issue of the journal.

Newsletter

Ce Ju and Cuntai Guan, Geometric Deep Learning-Based Classifiers for Motor Imagery Classification using Electroencephalograms, IEEE Brain Newsletter, 2024.

Preprints

Ce Ju and Cuntai Guan, Deep Optimal Transport on SPD Manifolds for Domain Adaptation, Under Review.

Bibliography

[1] Armin Thomas, Christopher Ré, and Russell Poldrack. Self-Supervised Learning of Brain Dynamics from Broad Neuroimaging Data. In *Advances in Neural Information Processing Systems*, volume 35, pages 21255–21269. Curran Associates, Inc., 2022. xxv, 23, 143, 144

[2] Ce Ju and Cuntai Guan. Tensor-cspnet: A novel geometric deep learning framework for motor imagery classification. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10955–10969, 2023. xxv, 8, 54, 110, 119, 144, 147

[3] Reinmar Kobler, Jun-ichiro Hirayama, Qibin Zhao, and Motoaki Kawanabe. SPD domain-specific batch normalization to crack interpretable unsupervised domain adaptation in EEG. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 6219–6235. Curran Associates, Inc., 2022. xxv, 41, 143, 144

[4] Jonathan R Wolpaw, Niels Birbaumer, William J Heetderks, Dennis J McFarland, P Hunter Peckham, Gerwin Schalk, Emanuel Donchin, Louis A Quatrano, Charles J Robinson, Theresa M Vaughan, et al. Brain-computer interface technology: a review of the first international meeting. *IEEE transactions on rehabilitation engineering*, 8(2):164–173, 2000. 3

[5] Han Yuan and Bin He. Brain–computer interfaces using sensorimotor rhythms: current state and future perspectives. *IEEE Transactions on Biomedical Engineering*, 61(5):1425–1435, 2014.

[6] Kai Keng Ang, Cuntai Guan, Kok Soon Phua, Chuanchu Wang, Longjiang Zhou, Ka Yin Tang, Gopal J Ephraim Joseph, Christopher Wee Keong Kuah, and Karen Sui Geok Chua. Brain-computer interface-based robotic end effector system for wrist and hand rehabilitation: results of a three-armed randomized controlled trial for chronic stroke. *Frontiers in neuroengineering*, 7: 30, 2014.

[7] Ander Ramos-Murguialday, Marco R Curado, Doris Broetz, Özge Yilmaz, Fabricio L Brasil, Giulia Liberati, Eliana Garcia-Cossio, Woosang Cho, Andrea Caria, Leonardo G Cohen, et al. Brain-machine interface in chronic stroke: randomized trial long-term follow-up. *Neurorehabilitation and neural repair*, 33(3):188–198, 2019.

[8] Nicholas Cheng, Kok Soon Phua, Hwa Sen Lai, Pui Kit Tam, Ka Yin Tang, Kai Kei Cheng, Raye Chen-Hua Yeow, Kai Keng Ang, Cuntai Guan, and Jeong Hoon Lim. Brain-computer interface-based soft robotic glove rehabilitation for stroke. *IEEE Transactions on Biomedical Engineering*, 67(12): 3339–3351, 2020.

[9] Ravikiran Mane, Tushar Chouhan, and Cuntai Guan. Bci for stroke rehabilitation: motor and beyond. *Journal of neural engineering*, 17(4):041001, 2020. 3

[10] Gert Pfurtscheller and FH Lopes Da Silva. Event-related eeg/meg synchronization and desynchronization: basic principles. *Clinical neurophysiology*, 110(11):1842–1857, 1999. 3, 14

[11] Gert Pfurtscheller and Christa Neuper. Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, 89(7):1123–1134, 2001. 13, 14

[12] Gert Pfurtscheller and Dennis J McFarland. Bcis that use sensorimotor rhythms. *Brain-computer interfaces: principles and practice*, page 227, 2012. 3, 14

[13] 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. 3, 65

[14] Marc Jeannerod. The representing brain: Neural correlates of motor intention and imagery. *Behavioral and Brain sciences*, 17(2):187–202, 1994. 4, 13

[15] R Beisteiner, P Höllinger, G Lindinger, W Lang, and A Berthoz. Mental representations of movements. brain potentials associated with imagination of hand movements. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 96(2):183–193, 1995. 4

[16] 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. 5, 8, 18

[17] Siavash Sakhavi, Cuntai Guan, and Shuicheng Yan. Learning temporal information for brain-computer interface using convolutional neural networks. *IEEE transactions on neural networks and learning systems*, 29(11):5619–5629, 2018.

[18] 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. 5, 8, 18

[19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 5

[20] Zhiwu Huang, Ruiping Wang, Shiguang Shan, Xianqiu Li, and Xilin Chen. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *International conference on machine learning*, pages 720–729. PMLR, 2015. 5

[21] Zhiwu Huang and Luc Van Gool. A riemannian network for spd matrix learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[22] Daniel Brooks, Olivier Schwander, Frederic Barbaresco, Jean-Yves Schneider, and Matthieu Cord. Riemannian batch normalization for spd neural networks. *Advances in Neural Information Processing Systems*, 32, 2019. 5, 40

[23] Johannes Müller-Gerking, Gert Pfurtscheller, and Henrik Flyvbjerg. Designing optimal spatial filters for single-trial eeg classification in a movement task. *Clinical neurophysiology*, 110(5):787–798, 1999. 8, 15, 86

[24] 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. 8, 34

[25] Ce Ju and Cuntai Guan. Graph neural networks on spd manifolds for motor imagery classification: A perspective from the time-frequency analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 8, 54, 110, 119

[26] Ce Ju, Dashan Gao, Ravikiran Mane, Ben Tan, Yang Liu, and Cuntai Guan. Federated transfer learning for eeg signal classification. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 3040–3045. IEEE, 2020. 8, 109

[27] Jean Decety. The neurophysiological basis of motor imagery. *Behavioural brain research*, 77(1-2):45–52, 1996. 13

[28] Martin Lotze and Ulrike Halsband. Motor imagery. *Journal of Physiology-paris*, 99(4-6):386–395, 2006. 13

[29] Jennifer A Stevens and Mary Ellen Phillips Stoykov. Using motor imagery in the rehabilitation of hemiparesis. *Archives of physical medicine and rehabilitation*, 84(7):1090–1092, 2003. 13

[30] Sjoerd De Vries and Theo Mulder. Motor imagery and stroke rehabilitation: a critical. *J Rehabil Med*, 39:5–13, 2007. 13

[31] Ruth Dickstein and Judith E Deutsch. Motor imagery in physical therapist practice. *Physical therapy*, 87(7):942–953, 2007. 13

[32] Aymeric Guillot and Christian Collet. Construction of the motor imagery integrative model in sport: a review and theoretical investigation of motor imagery use. *International Review of Sport and Exercise Psychology*, 1(1): 31–44, 2008. 13

[33] Jonathan R Wolpaw. Brain-computer interfaces (bcis) for communication and control. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 1–2, 2007. 13

[34] Jd R Millan, Frederic Renkens, Josep Mourino, and Wulfram Gerstner. Noninvasive brain-actuated control of a mobile robot by human eeg. *IEEE Transactions on biomedical Engineering*, 51(6):1026–1033, 2004. 13

[35] Karl LaFleur, Kaitlin Cassady, Alexander Doud, Kaleb Shades, Eitan Rogin, and Bin He. Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface. *Journal of neural engineering*, 10(4):046003, 2013.

[36] Jianjun Meng, Shuying Zhang, Angeliki Bekyo, Jaron Olsoe, Bryan Baxter, and Bin He. Noninvasive electroencephalogram based control of a robotic arm for reach and grasp tasks. *Scientific Reports*, 6(1):38565, 2016. 13

[37] Lars Ersland, Gunnar Rosén, Arvid Lundervold, Alf Inge Smievoll, Terje Tillung, Håkan Sundberg, et al. Phantom limb imaginary fingertapping causes primary motor cortex activation: an fmri study. *Neuroreport*, 8(1): 207–210, 1996. 13

[38] R Beisteiner, G Gomiscek, M Erdler, C Teichtmeister, E Moser, and L Deecke. Comparing localization of conventional functional magnetic resonance imaging and magnetoencephalography. *European Journal of Neuroscience*, 7(5):1121–1124, 1995. 13

[39] Gert Pfurtscheller. Functional brain imaging based on erd/ers. *Vision research*, 41(10-11):1257–1260, 2001. 14, 65

[40] Christa Neuper and Gert Pfurtscheller. Event-related dynamics of cortical rhythms: frequency-specific features and functional correlates. *International journal of psychophysiology*, 43(1):41–58, 2001. 14

[41] Nicolas Brodu, Fabien Lotte, and Anatole Lécuyer. Comparative study of band-power extraction techniques for motor imagery classification. In *2011 IEEE symposium on computational intelligence, cognitive algorithms, mind, and brain (CCMB)*, pages 1–6. IEEE, 2011. 15

[42] Dean J Krusienski, Dennis J McFarland, and Jonathan R Wolpaw. Value of amplitude, phase, and coherence features for a sensorimotor rhythm-based brain–computer interface. *Brain research bulletin*, 87(1):130–134, 2012. 27

[43] Jean-Philippe Lachaux, Eugenio Rodriguez, Jacques Martinerie, and Francisco J Varela. Measuring phase synchrony in brain signals. *Human brain mapping*, 8(4):194–208, 1999. 15

[44] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995. 15, 16

[45] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936. 15

[46] 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. 15

[47] Florian Yger. A review of kernels on covariance matrices for bci applications. In *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2013. 15, 33

[48] Benjamin Blankertz, K-R Muller, Dean J Krusienski, Gerwin Schalk, Jonathan R Wolpaw, Alois Schlogl, Gert Pfurtscheller, Jd R Millan, Michael Schroder, and Niels Birbaumer. The bci competition iii: Validating alternative approaches to actual bci problems. *IEEE transactions on neural systems and rehabilitation engineering*, 14(2):153–159, 2006. 15

[49] Michael Tangermann, Klaus-Robert Müller, Ad Aertsen, Niels Birbaumer, Christoph Braun, Clemens Brunner, Robert Leeb, Carsten Mehring, Kai J Miller, Gernot Mueller-Putz, et al. Review of the bci competition iv. *Frontiers in neuroscience*, page 55, 2012. 15

[50] 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. 17

[51] Guido Dornhege, Benjamin Blankertz, Matthias Krauledat, Florian Losch, Gabriel Curio, and Klaus-Robert Müller. Optimizing spatio-temporal filters for improving brain-computer interfacing. *Advances in Neural Information Processing Systems*, 18, 2005. 17

[52] 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. 17

[53] Fabien Lotte and Cuntai Guan. Regularizing common spatial patterns to improve bci designs: unified theory and new algorithms. *IEEE Transactions on biomedical Engineering*, 58(2):355–362, 2010. 17

[54] 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. 17

[55] Ebrahim A Mousavi, Jerome J Maller, Paul B Fitzgerald, and Brian J Lithgow. Wavelet common spatial pattern in asynchronous offline brain computer interfaces. *Biomedical Signal Processing and Control*, 6(2):121–128, 2011. 17

[56] Wojciech Samek, Motoaki Kawanabe, and Klaus-Robert Müller. Divergence-based framework for common spatial patterns algorithms. *IEEE Reviews in Biomedical Engineering*, 7:50–72, 2013. 17

[57] Ravikiran Mane, Effie Chew, Karen Chua, Kai Keng Ang, Neethu Robinson, A Prasad Vinod, Seong-Whan Lee, and Cuntai Guan. Fbcnet: A multi-view convolutional neural network for brain-computer interface. *arXiv preprint arXiv:2104.01233*, 2021. 18

[58] Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. Deep learning-based electroencephalography analysis: a systematic review. *Journal of neural engineering*, 16(5): 051001, 2019. 18

[59] Johan N. van der Meer, André Pampel, Eus J.W. Van Someren, Jennifer R. Ramautar, Ysbrand D. van der Werf, German Gomez-Herrero, Jöran Lepsien, Lydia Hellrung, Hermann Hinrichs, Harald E. Möller, and Martin Walter. Carbon-wire loop based artifact correction outperforms post-processing EEG/fMRI corrections—A validation of a real-time simultaneous EEG/fMRI correction method. *NeuroImage*, 125:880–894, 2016. 22, 145

[60] Johan van der Meer, André Pampel, Eus van Someren, Jennifer Ramautar, Ysbrand van der Werf, German Gomez-Herrero, Jöran Lepsien, Lydia Hellrung, Hermann Hinrichs, Harald Möller, and Martin Walter. "Eyes Open - Eyes Closed" EEG/fMRI data set including dedicated "Carbon Wire Loop" motion detection channels. *Data in Brief*, 7:990–994, 2016. 22, 143

[61] Madeleine Bullock, Graeme D. Jackson, and David F. Abbott. Artifact Reduction in Simultaneous EEG-fMRI: A Systematic Review of Methods and Contemporary Usage. *Frontiers in Neurology*, 12:622719, 2021. 22

[62] Philip J. Allen, Oliver Josephs, and Robert Turner. A Method for Removing Imaging Artifact from Continuous EEG Recorded during Functional MRI. *NeuroImage*, 12(2):230–239, 2000. 22

[63] R.K. Niazy, C.F. Beckmann, G.D. Iannetti, J.M. Brady, and S.M. Smith. Removal of FMRI environment artifacts from EEG data using optimal basis sets. *NeuroImage*, 28(3):720–737, 2005. 22

[64] Mainak Jas, Denis A. Engemann, Yousra Bekhti, Federico Raimondo, and Alexandre Gramfort. Autoreject: Automated artifact rejection for MEG and EEG data. *NeuroImage*, 159:417–429, 2017. 22

[65] Alain de Cheveigné. Sparse time artifact removal. *Journal of Neuroscience Methods*, 262:14–20, 2016. 22

[66] Alain De Cheveigné. ZapLine: A simple and effective method to remove power line artifacts. *NeuroImage*, 207:116356, 2020. 23

[67] Alain de Cheveigné and Lucas C. Parra. Joint decorrelation, a versatile tool for multichannel data analysis. *NeuroImage*, 98:487–505, 2014. 23

[68] Pierre Ablin, Jean-Francois Cardoso, and Alexandre Gramfort. Faster Independent Component Analysis by Preconditioning With Hessian Approximations. *IEEE Transactions on Signal Processing*, 66(15):4040–4049, 2018. 23

[69] Luca Pion-Tonachini, Ken Kreutz-Delgado, and Scott Makeig. ICLabel: An automated electroencephalographic independent component classifier, dataset, and website. *NeuroImage*, 198:181–197, 2019. 23

[70] Oscar Esteban, Christopher J. Markiewicz, Ross W. Blair, Craig A. Moodie, A. Ilkay Isik, Asier Erramuzpe, James D. Kent, Mathias Goncalves, Elizabeth DuPre, Madeleine Snyder, Hiroyuki Oya, Satrajit S. Ghosh, Jessey Wright, Joke Durnez, Russell A. Poldrack, and Krzysztof J. Gorgolewski. fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nature Methods*, 16(1): 111–116, 2019. 23

[71] Kamalaker Dadi, Gaël Varoquaux, Antonia Machlouzarides-Shalit, Krzysztof J. Gorgolewski, Demian Wassermann, Bertrand Thirion, and Arthur Mensch. Fine-grain atlases of functional modes for fMRI analysis. *NeuroImage*, 221:117126, 2020. 23

[72] Alexandre Défossez, Charlotte Caucheteux, Jérémy Rapin, Ori Kabeli, and Jean-Rémi King. Decoding speech perception from non-invasive brain recordings, 2023. 23, 143

[73] Hae-Jeong Park and Karl Friston. Structural and functional brain networks: from connections to cognition. *Science*, 342(6158):1238411, 2013. 26

[74] Ayumu Yamashita, Yuki Sakai, Takashi Yamada, Noriaki Yahata, Akira Kunimatsu, Naohiro Okada, Takashi Itahashi, Ryuichiro Hashimoto, Hiroto Mizuta, Naho Ichikawa, et al. Generalizable brain network markers of major depressive disorder across multiple imaging sites. *PLoS biology*, 18(12): e3000966, 2020. 26

[75] Dean J Krusienski, Dennis J McFarland, JOSÉ C Principe, and E Wolpaw. Bci signal processing: feature extraction. *Brain-Computer Interfaces: Principles and Practice, eds JR Wolpaw and EW Wolpaw (New York, NY: Oxford University Press)*, pages 123–146, 2012. 26

[76] Peter Petersen, S Axler, and KA Ribet. *Riemannian geometry*, volume 171. Springer, 2006. 27, 89, 106

[77] Jürgen Jost and Jeurgen Jost. *Riemannian geometry and geometric analysis*, volume 42005. Springer, 2008. 27

[78] P Thomas Fletcher, Conglin Lu, Stephen M Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging*, 23(8):995–1005, 2004. 31, 35, 36, 37

[79] P Thomas Fletcher and Sarang Joshi. Riemannian geometry for the statistical analysis of diffusion tensor data. *Signal Processing*, 87(2):250–262, 2007. 31, 35

[80] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A riemannian framework for tensor computing. *International Journal of computer vision*, 66:41–66, 2006. 31

[81] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009. 31

[82] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. *Fast and Simple Computations on Tensors with Log-Euclidean Metrics.* PhD thesis, INRIA, 2005. 32

[83] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 56(2):411–421, 2006. 32

[84] Xavier Pennec. Manifold-valued image processing with spd matrices. In *Riemannian geometric statistics in medical image analysis*, pages 75–134. Elsevier, 2020. 32

[85] 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. 33

[86] Alexandre Barachant, Stphane Bonnet, Marco Congedo, and Christian Jutten. Common spatial pattern revisited by riemannian geometry. In *2010 IEEE International Workshop on Multimedia Signal Processing*, pages 472–476. IEEE, 2010. 34

[87] 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. 34

[88] Or Yair, Mirela Ben-Chen, and Ronen Talmon. Parallel transport on the cone manifold of spd matrices for domain adaptation. *IEEE Transactions on Signal Processing*, 67(7):1797–1811, 2019. 34

[89] Thomas Fletcher. Geodesic regression on riemannian manifolds. In *Proceedings of the Third International Workshop on Mathematical Foundations of Computational Anatomy-Geometrical and Statistical Methods for Modelling Biological Shape Variability*, pages 75–86, 2011. 35

[90] Maurice Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. In *Annales de l'institut Henri Poincaré*, volume 10, pages 215–310, 1948. 35

[91] Karsten Grove and Hermann Karcher. How to conjugate c 1-close group actions. *Mathematische Zeitschrift*, 132(1):11–20, 1973. 35

[92] Tom Fletcher. Statistics on manifolds. In *Riemannian geometric statistics in medical image analysis*, pages 39–74. Elsevier, 2020. 35

[93] P Thomas Fletcher. Geodesic regression and the theory of least squares on riemannian manifolds. *International journal of computer vision*, 105:171–185, 2013. 37

[94] Wilfrid Gangbo and Robert J. McCann. The geometry of optimal transportation. *Acta Mathematica*, 177(2):113 – 161, 1996. 38

[95] John Lott and Cédric Villani. Ricci curvature for metric-measure spaces via optimal transport. *Annals of Mathematics*, pages 903–991, 2009. 38

[96] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 38

[97] Kenneth Atz, Francesca Grisoni, and Gisbert Schneider. Geometric deep learning on molecular representations. *Nature Machine Intelligence*, 3(12): 1023–1032, 2021. 38

[98] Raphael JL Townshend, Stephan Eismann, Andrew M Watkins, Ramya Rangan, Maria Karelina, Rhiju Das, and Ron O Dror. Geometric deep learning of rna structure. *Science*, 373(6558):1047–1051, 2021.

[99] Huazhu Fu, Yitian Zhao, Pew-Thian Yap, Carola-Bibiane Schönlieb, and Alejandro F Frangi. Guest editorial special issue on geometric deep learning in medical imaging. *IEEE Transactions on Medical Imaging*, 42(2):332–335, 2023. 38

[100] Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012. 39

[101] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

[102] Laurent Sifre and Stéphane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1233–1240, 2013. 39

[103] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. 39

[104] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019. 39

[105] Zhiwu Huang and Luc Van Gool. A Riemannian Network for SPD Matrix Learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 2036–2042. AAAI Press, 2017. event-place: San Francisco, California, USA. 40

[106] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008. 41

[107] Max Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian Optimization in PyTorch, 2020. 41

[108] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000. 42

[109] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000. 42

[110] Guy Rosman, Michael M Bronstein, Alexander M Bronstein, and Ron Kimmel. Nonlinear dimensionality reduction by topologically constrained isometric embedding. *International Journal of Computer Vision*, 89(1):56–68, 2010. 42

[111] Antonio Criminisi, Jamie Shotton, Ender Konukoglu, et al. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and trends® in computer graphics and vision*, 7(2–3):81–227, 2012. 42

[112] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010. 42

[113] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003. 42

[114] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006. 42

[115] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 42, 112

[116] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2000. 42, 43

[117] Frank Nielsen. An elementary introduction to information geometry. *Entropy*, 22(10):1100, 2020. 43

[118] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951. 43

[119] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009. 43

[120] Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, and Gustavo K Rohde. Optimal mass transport: Signal processing and machine-learning applications. *IEEE signal processing magazine*, 34(4):43–59, 2017. 43

[121] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015. 43

[122] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Center for Research in Economics and Statistics Working Papers*, (2017-86), 2017.

[123] Alfred Galichon. *Optimal transport methods in economics*. Princeton University Press, 2018.

[124] Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Neural architecture search with bayesian optimisation and optimal transport. *Advances in neural information processing systems*, 31, 2018. 43

[125] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pages 666–704, 1781. 43

[126] Leonid V Kantorovich. On the translocation of masses. *Journal of mathematical sciences*, 133(4):1381–1382, 2006. 43

[127] Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4): 375–417, 1991. 45

[128] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in neural information processing systems*, 30, 2017. 45, 88

[129] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 447–463, 2018. 46, 88

[130] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[131] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *The Journal of Machine Learning Research*, 22(1):3571–3578, 2021.

[132] Viet Huynh, Dinh Q Phung, and He Zhao. Optimal transport for deep generative models: State of the art and research challenges. In *IJCAI*, pages 4450–4457, 2021. 45

[133] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016. 45, 86, 88, 91, 104

[134] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3733–3742, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964. 45, 86, 88

[135] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017. 56

[136] John S Barlow. Methods of analysis of nonstationary eegs, with emphasis on segmentation techniques: a comparative review. *Journal of clinical neurophysiology: official publication of the American Electroencephalographic Society*, 2(3):267–304, 1985. 59

[137] David P Allen and Colum D MacKinnon. Time–frequency analysis of movement-related spectral power in eeg during repetitive movements: A comparison of methods. *Journal of neuroscience methods*, 186(1):107–115, 2010. 63

[138] S Blanco, R Quian Quiroga, OA Rosso, and S Kochen. Time-frequency analysis of electroencephalogram series. *Physical review E*, 51(3):2624, 1995. 63

[139] Brian J Roach and Daniel H Mathalon. Event-related eeg time-frequency analysis: an overview of measures and an analysis of early gamma band phase locking in schizophrenia. *Schizophrenia bulletin*, 34(5):907–926, 2008.

[140] Alexandros T Tzallas, Markos G Tsipouras, and Dimitrios I Fotiadis. Epileptic seizure detection in eegs using time–frequency analysis. *IEEE transactions on information technology in biomedicine*, 13(5):703–710, 2009.

[141] Santiago Morales and Maureen E Bowers. Time-frequency analysis methods and their application in developmental eeg data. *Developmental Cognitive Neuroscience*, 54:101067, 2022. 63

[142] Hojjat Adeli, Ziqin Zhou, and Nahid Dadmehr. Analysis of eeg records in an epileptic patient using wavelet transform. *Journal of neuroscience methods*, 123(1):69–87, 2003. 63

[143] Abdulhamit Subasi. Eeg signal classification using wavelet feature extraction and a mixture of expert model. *Expert Systems with Applications*, 32(4): 1084–1093, 2007.

[144] Hojjat Adeli, Samanwoy Ghosh-Dastidar, and Nahid Dadmehr. A wavelet-chaos methodology for analysis of eegs and eeg subbands to detect seizure and epilepsy. *IEEE Transactions on Biomedical Engineering*, 54(2):205–211, 2007. 63

[145] Neethu Robinson, A Prasad Vinod, Kai Keng Ang, Keng Peng Tee, and Cuntai T Guan. Eeg-based classification of fast and slow hand movements using wavelet-csp algorithm. *IEEE Transactions on Biomedical Engineering*, 60(8):2123–2132, 2013. 63

[146] Dennis Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-part III: radio and communication engineering*, 93(26):429–441, 1946. 63

[147] Jean Morlet, G Arens, E Fourgeau, and D Glard. Wave propagation and sampling theory—part i: Complex signal and scattering in multilayered media. *Geophysics*, 47(2):203–221, 1982.

[148] Jetal Morlet, G Arens, Eliane Fourgeau, and D Giard. Wave propagation and sampling theory—part ii: Sampling theory and complex waves. *Geophysics*, 47(2):222–236, 1982. 63

[149] James W Demmel. *Applied numerical linear algebra*. SIAM, 1997. 68

[150] Roger A Horn, Roger A Horn, and Charles R Johnson. *Topics in matrix analysis*. Cambridge university press, 1994. 69

[151] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30, 2006. 73

[152] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 82

[153] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980. 82

[154] Salem Said, Lionel Bombrun, Yannick Berthoumieu, and Jonathan H Manton. Riemannian gaussian distributions on the space of symmetric positive definite matrices. *IEEE Transactions on Information Theory*, 63(4):2153–2170, 2017. 82, 100

[155] Matthias Hein, Jean-Yves Audibert, and Ulrike von Luxburg. From graphs to manifolds–weak and strong pointwise consistency of graph laplacians. In *International Conference on Computational Learning Theory*, pages 470–485. Springer, 2005. 83

[156] Amit Singer. From graph to manifold laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128–134, 2006.

[157] Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, 2008. 83

[158] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007. 85

[159] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[160] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 85

[161] Yuan-Pin Lin and Tzyy-Ping Jung. Improving eeg-based emotion classification using conditional transfer learning. *Frontiers in human neuroscience*, 11:334, 2017. 86

[162] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007. 86

[163] Zoltan J Koles, Michael S Lazar, and Steven Z Zhou. Spatial patterns underlying population differences in the background eeg. *Brain topography*, 2 (4):275–284, 1990. 86

[164] Or Yair, Felix Dietrich, Ronen Talmon, and Ioannis G Kevrekidis. Domain adaptation with optimal transport on the manifold of spd matrices. *arXiv preprint arXiv:1906.00616*, 2019. 86, 88, 92, 104, 106

[165] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013. 87, 93

[166] Manfredo Perdigao Do Carmo and J Flaherty Francis. *Riemannian geometry*, volume 6. Springer, 1992. 90

[167] Robert J McCann. Polar factorization of maps on riemannian manifolds. *Geometric & Functional Analysis GAFA*, 11(3):589–608, 2001. 90, 93

[168] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016. 94

[169] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*, pages 153–171. Springer, 2017. 94

[170] Young-Heon Kim and Brendan Pass. Multi-marginal optimal transport on riemannian manifolds. *American Journal of Mathematics*, 137(4):1045–1060, 2015. 94

[171] Rosanna Turrisi, Rémi Flamary, Alain Rakotomamonjy, and Massimiliano Pontil. Multi-source domain adaptation via weighted joint distributions optimal transport. In *Uncertainty in Artificial Intelligence*, pages 1970–1980. PMLR, 2022. 94

[172] Clément Bonet, Benoıt Malézieux, Alain Rakotomamonjy, Lucas Drumetz, Thomas Moreau, Matthieu Kowalski, and Nicolas Courty. Sliced-wasserstein on symmetric positive definite matrices for m/eeg signals. pages 2777–2805, 2023. 99

[173] Suvrit Sra and Reshad Hosseini. Conic geometric optimization on the manifold of positive definite matrices. *SIAM Journal on Optimization*, 25(1): 713–739, 2015. 106

[174] Kay Gregor Hartmann, Robin Tibor Schirrmeister, and Tonio Ball. Eeggan: Generative adversarial networks for electroencephalograhic (eeg) brain signals. *arXiv preprint arXiv:1806.01875*, 2018. 109

[175] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 109, 163

[176] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33: 12438–12448, 2020. 163

[177] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 109, 164

[178] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 121

[179] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 121

[180] Stefan Debener, Markus Ullsperger, Markus Siegel, and Andreas K Engel. Single-trial eeg-fmri reveals the dynamics of cognitive function. *Trends in cognitive sciences*, 10(12):558–563, 2006. 123

[181] Wanmei Ou, Aapo Nummenmaa, Jyrki Ahveninen, John W Belliveau, Matti S Hämäläinen, and Polina Golland. Multimodal functional imaging using fmri-informed regional eeg/meg source estimation. *Neuroimage*, 52(1): 97–108, 2010.

[182] René J Huster, Stefan Debener, Tom Eichele, and Christoph S Herrmann. Methods for simultaneous eeg-fmri: an introductory review. *Journal of Neuroscience*, 32(18):6053–6060, 2012.

[183] Sepideh Sadaghiani and Jonathan Wirsich. Intrinsic connectome organization across temporal scales: New insights from cross-modal approaches. *Network Neuroscience*, 4(1):1–29, 2020. 123

[184] F. Moeller, L. Tyvaert, D. K. Nguyen, P. LeVan, A. Bouthillier, E. Kobayashi, D. Tampieri, F. Dubeau, and J. Gotman. EEG-fMRI: Adding to standard evaluations of patients with nonlesional frontal lobe epilepsy. *Neurology*, 73 (23):2023–2030, 2009. 123

[185] Jean Daunizeau, Helmut Laufs, and Karl J. Friston. EEG-fMRI Information Fusion: Biophysics and Data Analysis. In Christoph Mulert and Louis Lemieux, editors, *EEG - fMRI*, pages 511–526. Springer Berlin Heidelberg, 2009. ISBN 978-3-540-87918-3 978-3-540-87919-0. 123

[186] Tracy Warbrick. Simultaneous EEG-fMRI: What Have We Learned and What Does the Future Hold? *Sensors*, 22(6):2262, 2022. 123, 147

[187] Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics: methodology and distribution*, pages 162–190. Springer, 1992. 123, 167

[188] Viivi Uurtio, João M Monteiro, Jaz Kandola, John Shawe-Taylor, Delmiro Fernandez-Reyes, and Juho Rousu. A tutorial on canonical correlation methods. *ACM Computing Surveys (CSUR)*, 50(6):1–33, 2017. 123

[189] Sven Dähne, Vadim V. Nikulin, David Ramírez, Peter J. Schreier, Klaus-Robert Müller, and Stefan Haufe. Finding brain oscillations with power dependencies in neuroimaging data. *NeuroImage*, 96:334–348, 2014. 123

[190] Sven Dähne, Felix Bießman, Wojciech Samek, Stefan Haufe, Dominique Goltz, Christopher Gundlach, Arno Villringer, Siamac Fazli, and Klaus-Robert Müller. Multivariate Machine Learning Methods for Fusing Functional Multimodal Neuroimaging Data. *Proceedings of the IEEE*, 103(9): 1–22, 2015. 123, 143

[191] Fani Deligianni, Maria Centeno, David W Carmichael, and Jonathan D Clayden. Relating resting-state fmri and eeg whole-brain connectomes across frequency bands. *Frontiers in neuroscience*, 8:258, 2014. 123, 143

[192] Fani Deligianni, David W Carmichael, Gary H Zhang, Chris A Clark, and Jonathan D Clayden. Noddi and tensor-based microstructural indices as predictors of functional connectivity. *Plos one*, 11(4), 2016. 123

[193] Hyunwoo J. Kim, Nagesh Adluru, Barbara B. Bendlin, Sterling C. Johnson, Baba C. Vemuri, and Vikas Singh. Canonical correlation analysis on riemannian manifolds and its applications. *Comput Vis ECCV.*, (8690):251–267, 2014. 124, 129, 134, 167

[194] Faezeh Fallah and Bin Yang. Supervised canonical correlation analysis of data on symmetric positive definite manifolds by riemannian dimensionality reduction. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8369–8373. IEEE, 2020. 124

[195] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International conference on machine learning*, pages 1083–1092. PMLR, 2015. 124

[196] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33: 9912–9924, 2020. 124

[197] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[198] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024. PMLR, 2021.

[199] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. 124, 138

[200] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023. 124

[201] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR, 2013. 124, 167

[202] 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. 125

[203] Maher Moakher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM journal on matrix analysis and applications*, 26(3):735–747, 2005. 126

[204] Frederick Emory Croxton. Applied general statistics. In *Applied general statistics*, pages 754–754. 1967. 129

[205] Serge Lang. *Fundamentals of differential geometry*, volume 191. Springer Science & Business Media, 2012. 132

[206] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022. 138

[207] Jackob N. Keynan, Avihay Cohen, Gilan Jackont, Nili Green, Noam Goldway, Alexander Davidov, Yehudit Meir-Hasson, Gal Raz, Nathan Intrator, Eyal Fruchter, Keren Ginat, Eugene Laska, Marc Cavazza, and Talma Hendler. Electrical fingerprint of the amygdala guides neurofeedback training for stress resilience. *Nature Human Behaviour*, 3(1):63–73, 2019. 143

[208] Jonathan Wirsich, Anne-Lise Giraud, and Sepideh Sadaghiani. Concurrent EEG- and fMRI-derived functional connectomes exhibit linked dynamics. *NeuroImage*, 219:116998, 2020.

[209] Rodolfo Abreu, João Jorge, Alberto Leal, Thomas Koenig, and Patrícia Figueiredo. EEG Microstates Predict Concurrent fMRI Dynamic Functional Connectivity States. *Brain Topography*, 34(1):41–55, 2021.

[210] Takeshi Ogawa, Hiroki Moriya, Nobuo Hiroe, Motoaki Kawanabe, and Jun ichiro Hirayama. Eeg-based neurofeedback with network components extraction: a data-driven approach by multilayer ica extension and simultaneous eeg-fmri measurements. *bioRxiv*, 2021. 143

[211] Robert J. Barry, Adam R. Clarke, Stuart J. Johnstone, Christopher A. Magee, and Jacqueline A. Rushby. EEG differences between eyes-closed and eyes-open resting conditions. *Clinical Neurophysiology*, 118(12):2765–2773, 2007. 143

[212] Víctor Costumero, Elisenda Bueichekú, Jesús Adrián-Ventura, and César Ávila. Opening or closing eyes at rest modulates the functional connectivity of V1 with default and salience networks. *Scientific Reports*, 10(1):9137, 2020. 143

[213] Daniel Wilson, Robin Tibor Schirrmeister, Lukas Alexander Wilhelm Gemein, and Tonio Ball. Deep Riemannian Networks for EEG Decoding, 2022. 143

[214] 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. 143

[215] 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, 2019.

[216] Dongrui Wu, Yifan Xu, and Bao-Liang Lu. Transfer Learning for EEG-Based Brain-Computer Interfaces: A Review of Progress Made Since 2016. *IEEE Transactions on Cognitive and Developmental Systems*, pages 1–1, 2020. 143

[217] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 143

[218] 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, 2018. 144

[219] Marios G. Philiastides, Tao Tu, and Paul Sajda. Inferring Macroscale Brain Dynamics via Fusion of Simultaneous EEG-fMRI. *Annual Review of Neuroscience*, 44(1):315–334, 2021. 147

[220] Stephen H. Fairclough and Fabien Lotte. Grand Challenges in Neurotechnology and System Neuroergonomics. *Frontiers in Neuroergonomics*, 1:602504, 2020. 147

[221] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005. 163

[222] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011. 163

[223] Hao-Ting Wang, Jonathan Smallwood, Janaina Mourao-Miranda, Cedric Huchuan Xia, Theodore D Satterthwaite, Danielle S Bassett, and Danilo Bzdok. Finding the needle in a high-dimensional haystack: Canonical correlation analysis for neuroscientists. *NeuroImage*, 216:116745, 2020. 167