Graph Neural Networks on SPD Manifolds for Motor Imagery Classification: A Perspective From the Time–Frequency Analysis

Ce Ju®, and Cuntai Guan®, Fellow, IEEE

Abstract—The motor imagery (MI) classification has been a prominent research topic in brain–computer interfaces (BCIs) based on electroencephalography (EEG). Over the past few decades, the performance of MI-EEG classifiers has seen gradual enhancement. In this study, we amplify the geometric deep-learning-based MI-EEG classifiers from the perspective of time–frequency analysis, introducing a new architecture called Graph-CSPNet. We refer to this category of classifiers as Geometric Classifiers, highlighting their foundation in differential geometry stemming from EEG spatial covariance matrices. Graph-CSPNet utilizes novel manifold-valued graph convolutional techniques to capture the EEG features in the time–frequency domain, offering heightened flexibility in signal segmentation for capturing localized fluctuations. To evaluate the effectiveness of Graph-CSPNet, we employ five commonly used publicly available MI-EEG datasets, achieving near-optimal classification accuracies in nine out of 11 scenarios. The Python repository can be found at https://github.com/GeometricBCI/Tensor-CSPNet-and-Graph-CSPNet.

Index Terms—Geometric deep learning, graph neural networks, motor imagery classification, spectral clustering, symmetric positive definite manifolds.

I. INTRODUCTION

A BRAIN–COMPUTER interface (BCI) is a technology that measures and analyzes the relevant information of a user’s brain activity and establishes a communication link between the brain and the external environment [1]. Electroencephalography (EEG)-based BCIs are among the most widely used, portable, and cost-effective BCIs and have led to numerous applications, such as poststroke motor rehabilitation, control of a wheelchair system, and video gaming [2]. The control of these EEG-based BCI applications is accompanied by EEG rhythmic changes over the sensorimotor cortices, including the posterior frontal and anterior parietal regions, and a wealth of studies and evidence over the past 30 years has demonstrated that the sensorimotor rhythm changes associated with motor imagery (MI) are effective control signals for BCIs [3], [4], [5].

During the planning and execution of movement, the sensorimotor rhythms exhibit changes in amplitude that are referred to as event-related desynchronization (ERD) and event-related synchronization (ERS), corresponding to decreases and increases in rhythmic activity, respectively. These changes generate patterns that can be reliably and distinctly discerned, enabling the EEG signals to be accurately classified [6]. In an EEG-based MI task, the individual mentally simulates physical movement, which activates the cortical sensorimotor systems [7] and primary sensorimotor areas [8], and an EEG device records EEG signals. Various machine-learning classifiers are then utilized to decode an individual’s intentions [9].

Building on the ERD/ERS effect observed during an MI task, conventional MI-EEG classifiers adopt the time–space–frequency principle to extract EEG signal patterns and rhythms. The essence of this principle lies in the analysis of EEG signals in terms of frequency, time, and space, allowing for the identification of patterns across these dimensions based on the dominant frequencies of rhythmic activity, their periods of occurrence, and their distribution over the sensorimotor cortices. [10], [11] This principle has been generalized to capture patterns across frequency, time, and space in EEGs. However, the nonstationary nature of EEG spectral contents can hinder the effectiveness of conventional methods that rely on statistical stationarity assumptions [12], [13]. In these circumstances, traditional Fourier analysis proves to be ineffective. To address this challenge, the time–frequency analysis is employed to localize rhythmic components in real time, strengthening the MI-EEG classifiers [14], [15], [16], [17]. Examples of such methods include the short-term frequency transform and the wavelet transform [18], [19], [20], which have been integrated into the common spatial pattern (CSP) approach to creating novel wavelet-based classifiers such as the wavelet-CSP classifier [21].

Motivated by the principles of Gabor’s time–frequency theory and Morlet’s wavelet theory [22], [23], [24], this study aims to enhance a geometric deep-learning-based MI-EEG classifier, namely Tensor-CSPNet [25], to facilitate a more effective exploration of local oscillatory components in EEGs. The proposed approach employs graph-based neural networks, Graph-CSPNet, to handle the MI-EEG classification using EEG spatial covariance matrices. A notable characteristic of this neural network incorporates a novel graph BiMap layer, which replaces fixed-length segmentation with a flexible...
time–frequency resolution to capture localized fluctuations, enabling simultaneous analysis in the time–frequency domain. Drawing inspiration from the Heisenberg–Gabor uncertainty principle, the graph BiMap layer strikes a balance by capturing signal information with high spectral resolution and low temporal resolution for low frequencies while adopting the opposite for high frequencies. Specifically, to create a low temporal resolution for low frequencies while adopting signal information with high spectral resolution and principle, the graph BiMap layer strikes a balance by capturing the discriminative power between task and then in the space domain.

Due to the limited number of trials available for training, the subject-specific scenario poses a big challenge for MI-EEG classification. Hence, the proposed geometric classifier will be evaluated on five publicly available MI-EEG datasets with two classifiers. Therefore, the proposed geometric classifier will be evaluated on five publicly available MI-EEG datasets with two classifiers, and existing neural networks on SPD manifolds. The rest of the article is structured as follows. Section II delves into the study history of MI-EEG classifiers and introduces SPD manifolds, graph convolutional networks, and existing neural networks on SPD manifolds. Section III outlines the architecture of Graph-CSPNet. The performance of Graph-CSPNet is then compared in a comprehensive set of experiments in Section IV, and several key issues are discussed in detail in Section V.

### II. PRELIMINARY

In the early stages of MI-EEG classification, feature extraction, and selection were commonly used. This involved hand-crafted features based on time-domain and frequency-domain statistics, such as band power, magnitude-squared coherence, and phase-locking value [26], [27], [28]. Traditional machine-learning approaches, including support vector machines (SVMs) and linear discriminant analysis, were then applied for classification.

The dominant perspective in MI-EEG classification has been the analysis in the time domain. Most methods utilize second-order statistics of EEG signals, specifically the spatial covariance matrix. The CSP methods are widely used spatial filtering techniques. These methods aim to obtain optimal spatial features by maximizing the variance of one class while minimizing the variance of the other [29], [30], [31], [32].

This results in the following time-domain paradigm for EEG signal analysis. Let $X \in \mathbb{R}^{n_c \times n_f}$ be a segment of EEG signals, where $n_c$ is the number of EEG channels, and $n_f$ is the number of sampled points on an epoch duration, and segment $X$ is assumed bandpass filtered, centered, and scaled. The linear classifier that predicts the label for segment $X$ can be written as follows:

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

where $N$ is the number of spatial filters, $\{w_i\}_{i=1}^N \in \mathbb{R}^{n_c}$ are spatial filters, and $\{\beta_i\}_{i=1}^N \in \mathbb{R}$ are biases. Significant attempts have been made to identify more versatile and practical spatial filters for BCI scenarios [9].

Spatial covariance matrices possess a substantial amount of discriminative information. This includes the variance captured in the on-diagonal entries and the coherence between adjacent channels recorded in the off-diagonal entries [33]. This information is closely related to rhythmic oscillations, which can be quantified through frequency analysis techniques like power spectral density and magnitude-squared coherence to track the ERD/ERS effect in MI tasks [5]. By focusing solely on the sensorimotor cortices and utilizing spectral power to assess zero-phase synchrony, classification outcomes are comparable, if not superior, to those achieved using combinations of other spatial features, such as coherence and phase-locking value. This implies that the information embedded within spatial covariance matrices is adequate for MI-EEG classification [27].

Barachant et al. [34] first utilized SPD manifolds for characterizing EEG spatial covariance matrices in a Riemannian-based classification framework. The discipline of SPD manifolds is ingrained with the affine invariant Riemannian metric $g^{\text{AIMR}}$. This approach referred to as the Riemannian-based approach, has since garnered increasing interest in the BCI community, with subsequent studies focusing on optimizing its framework [33], [34], [35] and applications to various BCI tasks [36], [37], [38]. The success of Riemannian-based approaches can be attributed to their ability to capture the discriminative power between task

| Geometric Classifiers | Tensor-CSPNet | Graph-CSPNet (Proposed) |
|-----------------------|--------------|-------------------------|
| Network Input Architecture | Tensorized Spatial Covariance Matrices. BiMaps; CNNs. | Time–Frequency Graph. Graph-BiMaps. |
| Distinctive Structure | CNNs for Temporal Dynamics. | Spectral Clustering for Time–Frequency Distributions. |
| Training Optimizer | Riemannian Optimization. | Riemannian Optimization. |
| Underlying Space Methodology Heritage Design Principle | $(\mathbb{S}_+^{1+}, g^{\text{AIMR}})$. CSP. The Time–Space–Frequency Principle: Exploitation in the frequency, space, and time domains sequentially. | $(\mathbb{S}_+^{1+}, g^{\text{AIMR}})$. CSP; Riemannian-Based Approaches. 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. |

**TABLE I**

**COMPARISON BETWEEN TENSOR-CSPNET AND GRAPH-CSPNET**
classes by utilizing Riemannian distances. The approach theoretically states in [39] that the Riemannian distance between class-related covariance matrices $S^+$ and $S^-$ can be expressed as follows:

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

where each $\lambda_i \in (0, 1)$ is the eigenvalue of (3) in Section II-B. This implies that a more significant Riemannian distance between two EEG spatial covariance matrices indicates a greater discriminative power, as their eigenvalues are more likely to be close to 1.

The exploration of SPD manifolds traces its roots back to the 1970s, springing from the classical task of enhancing geodetic networks [45]. Since then, it has been assimilated into a multitude of disciplines, including but not limited to geometric statistics, signal processing, computer vision, and robotics [46], [47], [48], [49], [50], [51], [52], [53]. The conceptualization of spatial covariance matrices as points on SPD manifolds represents a significant advancement in the field, going beyond its purely formalistic nature. This abstraction acknowledges the positive-definite constraints and infuses new vitality into research by incorporating technical tools from theoretical mathematics and physics. For instance, the abstract formulation of SPD manifolds provides a natural framework for interpolating diffusion tensors and introduces a novel measure of anisotropy in diffusion tensor magnetic resonance imaging [54]. Moreover, applying optimal transport theory on SPD manifolds establishes a solid mathematical foundation for evaluating the variability between calibration-feedback phases in a BCI system [40], [55].

Recently, there has been a gradual shift in the trend of MI-EEG classifiers toward the adoption of deep-learning-based approaches that operate in the time domain, allowing for the automatic extraction of features from the original EEG signals. In response to this trend, the first novel geometric deep-learning-based approach on SPD manifolds, known as Tensor-CSPNet, has been proposed to leverage the time-space-frequency principle in capturing discriminative information from EEG spatial covariance matrices. This approach has demonstrated performance levels close to optimality, comparable to those achieved by convolutional neural networks-based approaches [56], [57], [58], [59]. This approach is referred to as Geometric Classifiers in this article. The success of this approach is attributed to the BiMap layer introduced in Section II-D, which maintains the Riemannian distances consistently during the learning process, as indicated by the affine invariance property of $g_{\text{ARM}}$ as follows:

$$d_{\text{ARM}}(S_1, S_2) = d_{\text{ARM}}(W S_1 W^T, W S_2 W^T)$$

where $S_1, S_2 \in S_{++}$, and weight matrix $W$ of the bimap transformation belongs to the general linear group. This preservation of Riemannian distances ensures the maintenance of discriminative power between task classes. Numerous recent studies have showcased the consistency and potential of this pathway in the realm of MI-EEG classification tasks [41], [42], [43], [44].

In summary, Table II presents the second-order statistics (EEG spatial covariance matrices) approaches for MI-EEG classifiers, with a more detailed discussion of time-domain statistics and frequency-domain statistics found in Section V-H. In the latter half of this section, we will briefly overview the computational techniques related to the proposed method.

### A. SPD Manifolds With Affine Invariant Riemannian Metric

The spatial covariance matrix of an EEG segment $X \in \mathbb{R}^{n_C \times n_C}$ is denoted as $S = XX^T \in S_{++}$, where $S_{++} := \{S \in \mathbb{R}^{n_C \times n_C} : S = S^T \text{ and } x^T S x > 0, \forall x \in \mathbb{R} / \{0\}\}$ is the space of real $n_C \times n_C$ symmetric and positive definite matrices. Equipped with the affine invariant Riemannian metric $g_{\text{ARM}}^P (v, w) := (P^{-(1/2)} v P^{-(1/2)}, P^{-(1/2)} w P^{-(1/2)})$ for each $v$ and $w$ on tangent space $T_P S_{++}$, the space of spatial covariance matrices becomes a Hadamard manifold, which refers to a Riemannian manifold that is complete and simply connected and has an everywhere nonpositive sectional curvature. The Riemannian distance between two matrices $S_1$ and $S_2$ is given by $d_{\text{ARM}}(S_1, S_2) = \| \log(S_1^{(1/2)} S_2^{(1/2)}) \|_F$, where $F$ is the Frobenius norm.

### B. Common Spatial Pattern

Let $S^+ \in \mathbb{R}^{n_C \times n_C}$ be the class-related estimates of spatial covariance matrices of segments in a two-class MI-EEG paradigm using the arithmetic mean, that is, $S^+ = (1/|I^+|) \sum_{i\in I^+} X_i X_i^T$ and $S^- = (1/|I^-|) \sum_{i\in I^-} X_i X_i^T$, where $I^+$ and $I^-$ are sets of segments in two classes, respectively. The CSP method is given by a simultaneous diagonalization of $S^+$ and $S^-$ as follows:

$$WS^+/W^T = \Lambda^+/\Lambda^-$$

| Category of Approach                  | Initiation Time | Principle                                                                 |
|--------------------------------------|-----------------|---------------------------------------------------------------------------|
| Common Spatial Patterns ([29]–[32], etc.) | 1999 [29]       | Search for a linear projection direction that maximizes the discriminative power between classes through solving Eigenvalue Problem 3. |
| Riemannian-Based Approaches ([33]–[35], etc.) | 2011 [34]       | Compare the quantity (i.e., Riemannian distance) consisting of the entire spectrum information of Eigenvalue Problem 3, instead of directly solving it. |
| Geometric Classifiers ([25], [40]–[45], etc.) | 2022 [25]       | Search for a nonlinear projection direction that maximizes the discriminative information between classes using gradient descent. |
where diagonal matrices \( \Lambda^+ \) 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:

\[
(S^+ + S^-)^{-1} S^{+/-} = \lambda^{+/-} w.
\]

A set of discriminative powers consists of \( m \) spatial filters \( \{w_j\}_{j=1}^m \) from both ends of the spectrum, that is, \( z_j = \log(w_j X^T W_j) \), where \( 1 \leq m \leq n_c \).

C. Graph Convolutional Networks

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 nonnegative edge weights \( [a_{ij}] \geq 0 \) for \( 1 \leq i, j \leq N \).

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

\[
H^{(l+1)} \leftarrow \sigma \left( (\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}) H^{(l)} W^{(l)} \right)
\]

where adjacency matrix with added self-connections \( \tilde{A} := A + I_N \), diagonal matrix \( \tilde{D} := \sum_{j=1}^N \tilde{A}_{ij} \), \( \sigma \) 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. Theoretical analysis suggests that graph convolutional networks provide a localized first-order approximation of spectral graph convolution [60].

D. Neural Networks on SPD Manifolds

The SPD matrix-valued network architecture in this section is derived from the following two studies.

1) SPDNet [61]: SPDNet introduces network architectures that operate on SPD matrices, incorporating the following layers.

a) BiMap: This layer performs the bimap transformation \( WSW^T \) on spatial covariance matrix \( S = U \Sigma U^T \). The transformation matrix \( W \) is typically required to have a full-rank rank.

b) ReEig: This layer performs \( U \max(\epsilon I, \Sigma) U^T \), where \( \epsilon \) is a rectification threshold, and \( I \) denotes the identity matrix.

c) LOG: This layer maps spatial covariance matrix \( S \) onto its tangent space at identity matrix \( I \) using \( U \log(\Sigma) U^T \). These layers are designed to ensure that the properties of symmetric and positive definite are maintained throughout the learning process.

2) Riemannian Batch Normalization [62]: This layer is designed as a type of SPD matrix-valued network architecture that utilizes parallel transport for batch centering and biasing around the Riemannian barycenter. Typically, parallel transport serves as a means to connect local information between various points because manifolds are the local concept, where basic operations that are simple in Euclidean spaces become intricate, leading to a distinct calculus on manifolds. This layer has been demonstrated to improve classification performance on EEGs, making it a valuable addition to the existing network architecture on SPD manifolds [25], [42].

III. METHODOLOGY

This section presents a novel geometric deep-learning approach, Graph-CSPNet, for MI-EEG classification. The architecture of Graph-CSPNet is depicted in Fig. 1, and a table containing the parameters for each layer can be found in Section V-F.

A. 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 \( \{\Delta t_i \times \Delta f_j\}_{i \in \mathcal{I}} \), the time–frequency distribution consists of EEG spatial covariance matrices \( \{S(\Delta t_i \times \Delta f_j)\}_{i \in \mathcal{I}} \), where \( S(\Delta t_i \times \Delta f_j) := \tilde{X}_t \tilde{X}_t^\top \) is the covariance matrix of bandpass filtered EEG signal \( \tilde{X}_t \in \mathbb{R}^{n_c \times M} \) within bandwidth \( \Delta f_i \). The determination of a segmentation plan \( \{\Delta t_i \times \Delta f_j\}_{i \in \mathcal{I}} \) 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. The ERD effect is characterized by a localized decrease in amplitude, while the ERS effect is marked by an increase in rhythmic activity amplitude. Both of these effects are highly specific to the frequency band of the event. It is essential to consider the frequency discretization \( \Delta f_j \) when working with traditional frequency bands, including \( \delta \) (<4 Hz), \( \theta \) (4–7 Hz), \( \mu \) (8–13 Hz), \( \beta \) (14–30 Hz), and \( \gamma \) (>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: To provide a concrete example, in the case of hand movement imagination, a time resolution of \( \Delta t = 125 \) ms is applicable to effectively capture the occurrence of the ERD/ERS effects that are specific to the \( \mu \) and \( \beta \) frequency bands, which offer the most effective discrimination. Precise selection of the time resolution is essential to ensure that the neural activity corresponding to these frequency bands is accurately detected and properly analyzed [6], [63].

B. 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 non-parametric statistical approach called the \( \epsilon \)-neighborhoods approach. This approach involves connecting two vertices, \( S_i \) and \( S_j \), with an edge if \( d_{\text{man}}(S_i, S_j) < \epsilon \).

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 \( \epsilon \)-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 \( B_{\epsilon_1 \epsilon_2} \) 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,
task classes. Subsequently, we apply a logarithmic mapping (LOG) layer to transform the SPD matrices onto the tangent space, which are then fed into the Riemannian Batch Normalization (RBN) statistical approach. Once the time–frequency graph is built, we employ an SPD matrix-valued graph convolutional network, consisting of the covariance matrices obtained from these segments serve as the vertices of the time–frequency graph, which is constructed using a novel nonparametric method, we gain a more comprehensive understanding of the intricate connections between EEG spatial covariance matrices within the time–frequency distribution.

Fig. 1. Illustration of the architecture of Graph-CSPNet: The EEG signal is divided into multiple segments in the time–frequency domain. The spatial covariance matrices obtained from these segments serve as the vertices of the time–frequency graph, which is constructed using a novel nonparametric method, we gain a more comprehensive understanding of the intricate connections between EEG spatial covariance matrices within the time–frequency distribution.

S_i and S_j ∈ B_{ε_1,ε_2} must satisfy the following conditions:

\[ 0 \leq \text{mid}(\Delta t_i) - \text{mid}(\Delta t_j) \leq \epsilon_1 \]
\[ |\text{mid}(\Delta f_i) - \text{mid}(\Delta f_j)| \leq \epsilon_2 \]

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 G(V, E) is used to store the similarities between pairs of vertices and is defined as follows:

\[ A = \begin{cases} 
  e^{-\frac{\Delta t}{\epsilon_1}}, & \text{if } S_i \text{ and } S_j \text{ are adjacent} \\
  0, & \text{otherwise}
\end{cases} \]

where preset Gaussian kernel width t > 0. These similarities are computed using the radial basis function kernel, which considers the Riemannian distance between adjacent vertices. If two vertices are not adjacent, the similarity is set to zero. In other words, if \( S_i \) and \( S_j \) are nonadjacent, the similarity value is assigned as zero. The pseudocode for generating adjacency matrix A refers to Section V-B.

This approach is referred to as the Local Graph Topology (LGT) method, as it effectively captures the local relationships within the time–frequency distribution. By utilizing the LGT method, we gain a more comprehensive understanding of the intricate connections between EEG spatial covariance matrices within the time–frequency distribution.

Remark:

1) Edge Weights: 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 Components: 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 IV

\[ A = \begin{pmatrix} A_\theta & A_\mu & A_\beta & A_\gamma \end{pmatrix} \]

C. Graph BiMap Layer

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

\[ H^{(l+1)} \leftarrow \text{RBN} \left( \text{ReEig} \left( W^{(l)} \left( D^{-1} \tilde{A}^{(l)} H^{(l)} W^{(l)\top} \right) \right) \right) \]

where \( \tilde{A}^{(l)} := A^{(l)} + I_N \), \( \tilde{D}_{ii} := \sum_j \tilde{A}_{ij}^{(l)} \), and \( W^{(l)} \) is a trainable transformation matrix with the full-row rank. The RBN (Riemannian batch normalization) and ReEig layers are introduced in Section II-D. \( H^{(l)} \in \mathbb{R}^{[V] \times N} \) is a node function in the lth layer. In particular, \( H^{(0)} := (S^1, \ldots, S^N) \), \( A^{(0)} \) is the adjacency matrix of the time-frequency graph, and \( \tilde{A}^{(l)} := I_N \), for \( l \geq 1 \).

The proposed layer-wise propagation rule is structurally similar to the classical rule in graph convolutional network architectures. The original linear mapping is replaced
with the BiMap mapping, which enables operations on SPD matrices. BiMap ensures that Riemannian distances are consistently preserved during the learning process while maintaining symmetric positive definiteness. This is crucial for MI-EEG classification tasks. Furthermore, instead of using $\bar{D}^{-1/2}A^{(0)}\bar{D}^{-1/2}$, we employ row-normalized $\bar{D}^{-1}\bar{A}^{(0)}$ because we aim to perform normalization only in the time direction. Additionally, we utilize the nonlinear operator ReEig, which drops the smallest eigenvalues, as described in Section V-E, 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 $\bar{D}^{-1}\bar{A}^{(0)}$ is presented as follows:

**Theorem 1:** Given a time–frequency graph $G(V,E)$. For the $l$th graph BiMap layer, let $A^{(l)}$ be its $|V| \times |V|$ adjacency matrix. For $i \in \{1,\ldots,|V|\}$, the perturbed spatial covariance matrix $\bar{S}_i \in \mathbb{R}^{nc \times nc}$ on $(\bar{S}_{++}, g_{\bar{A}^{(l)}})$ is written in to the original spatial covariance matrix $S_i$ and a graph-based perturbation term, that is, $\bar{S}_i = S_i + \Delta S_i$, where $\Delta S_i := A^{(l)}[i,\ldots,|V|]S_i$. Then, the spatial of the spectrum covariance matrix has a perturbation ratio as follows:

$$ (1 - N_iC_i) \leq \frac{\lambda(\bar{S}_i)}{\lambda(S_i)} \leq (1 + N_iC_i) \quad (4) $$

where $C_i = \max_{(i,j)\in E}\{\exp(\lambda_{ij})\}$, $N_i = |\{(j|i, j) \in E\}|$, and $\lambda_{ij}$ is largest eigenvalue of $\log(S_{i}^{(1/2)}S_{j}^{(1/2)})$.

It is imperative that we initially establish the subsequent Lemma to prove Theorem 1.

**Lemma 2:** Let $\{S_i\}_{i=1}^{N}$ be a set of SPD matrices. Then, any linear combination $\sum_{i=1}^{N} \alpha_i S_i$ is still SPD for $\alpha_i > 0$.

**Proof:** The symmetry is obtained by $(\sum_{i=1}^{N} \alpha_i S_i)^T = \sum_{i=1}^{N} \alpha_i S_i^T$, and the positive definite is achieved by $v^T\sum_{i=1}^{N} \alpha_i S_i v = \sum_{i=1}^{N} \alpha_i (v^T S_i v) > 0$, for any $v \neq 0$. This concludes the proof. \( \square \)

**Proof of Theorem 1:** Consider the perturbation $\Delta S_i$ on each node $S_i$ $(1 \leq i \leq |V|)$, which is taken as a linear combination of adjacent nodes as $S_i + \Delta S_i = S_i + \sum_{(i,j)\in E} a_{ij}S_j$, where $a_{ij} := e^{-d_{A^{(l)}}(S_i,S_j)/l} \leq 1$ with a large $l$ 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 2.

The Riemannian distance between node $S_i$ and its adjacent node $S_j$ is $d_{A^{(l)}}(S_i,S_j) := ||\log(S_{i}^{(1/2)}S_{j}^{(1/2)})||_F$. Notice that $|\log(S_{i}^{(1/2)}S_{j}^{(1/2)})|$ is an $nc \times nc$ symmetric matrix and bounded by an inequality according to the Rayleigh–Ritz theorem [64] as $-\lambda_{ij} I \preceq \log(S_{i}^{(1/2)}S_{j}^{(1/2)}) \preceq \lambda_{ij} I$, where $\lambda_{ij}$ is the largest Rayleigh quotient of $\log(S_{i}^{(1/2)}S_{j}^{(1/2)})$. Since $\exp(\lambda_{ij})$ is the eigenvalue of $S_{i}^{(1/2)}S_{j}^{(1/2)}$ according to the definition of exponential matrix and exp is monotone increasing, we have $-\exp(\lambda_{ij}) I \preceq S_{i}^{(1/2)}S_{j}^{(1/2)} \preceq \exp(\lambda_{ij}) I$.

Note that the perturbation $S_i + \Delta S_i = S_i + \sum_{(i,j)\in E} a_{ij}S_j = S_{i}^{(1/2)}(I + \sum_{(i,j)\in E} a_{ij}S_j)S_{i}^{(1/2)}$. Then, we have $(1 - N_iC_i)S_i \preceq S_i + \sum_{(i,j)\in E} a_{ij}S_j \preceq (1 + N_iC_i)S_i$, where $N_i = |\{(j|i,i, j) \in E\}|$ and $C_i = \max_{(i,j)\in E}\{\exp(\lambda_{ij})\}$. Lastly, the spectrum perturbation ratio in (4) can be obtained using Weyl’s monotonicity theorem [65].

Theorem 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.

In order to uphold the lowest possible spectrum change ratio, it is imperative that $\bar{A}^{(0)}$ be designated as the adjacency matrix of the time-frequency graph and $A^{(l)} := I_N$ for $l \geq 1$. The visualization of the impact of spectrum changes, as stated in Theorem 1, is discussed in detail in Section IV-G.

**D. LOG Layer**

After the graph BiMap layers, the resulting SPD matrices are mapped onto the tangent space using the logarithm mapping with the identity matrix as the base point, as explained in the LOG layer in Section II-D. In particular, this logarithm mapping is well-defined and applicable to any SPD matrix with the same dimension.

**E. Loss Function**

For the sake of simplicity, the cross-entropy loss is adopted as the loss function for Graph-CSPNet.

**IV. EXPERIMENTS**

**A. Evaluation Datasets and Scenarios**

As part of the evaluation, we employed five widely used MI-EEG datasets. This section offers a concise overview of these datasets. The data of these datasets cannot be accessed via the MOABB\(^1\) or BNCI Horizon 2020.\(^2\) All digital signals underwent filtering using Chebyshev Type II filters with 4-Hz intervals. The filters were specifically designed to have a maximum passband loss of 3 dB and a minimum stopband attenuation of 30 dB. The selection criteria for the five evaluation datasets from MOABB and BNCI Horizon 2020 requires each training session to have more than 150 trial instances, with at least 70 trials for each category. Table III summarizes the five evaluation datasets.

1) **KU (Also Known as Lee2019_MI, From MOABB):** It comprises EEG signals obtained from 54 subjects participating in a binary class EEG-MI task. The EEG signals were captured at 1000 Hz using 62 electrodes. For evaluation, 20 electrodes in the motor cortex region were selected, that is, FC-5/3/1/2/4/6, C-5/3/1/2/4/5, and CP-5/3/1/2/4/6. The dataset was divided into two sessions, S1 and S2, each composed of two phases: training and testing. Each phase consisted of 100 trials balanced between right and left-hand imagery tasks, resulting in 21 600 trials available for evaluation. The EEG signals were epoched from the first second to 3.5 s

\(^1\)The MOABB package comprises a benchmark dataset for state-of-the-art decoding algorithms encompassing 12 open access datasets and over 250 subjects. This package is accessible at https://github.com/NeuroTechX/moabb.

\(^2\)The datasets within the BNCI Horizon 2020 project are accessible at http://bnci-horizon-2020.eu/database/data-sets.
in reference to the stimulus onset, resulting in a total duration of 2.5 s.

2) Cho2017 (From MOABB): It conducted a BCI experiment with 52 subjects involving motor imagery movement (MI movement) of the left and right hands. The EEG data were collected using 64 Ag/AgCl active electrodes, with a 64-channel montage based on the international 10-10 system to record the EEG signals at a sampling rate of 512 Hz. The recording commenced at second 0, following the prompt, and continued for 3 s until the conclusion of the cross period. For evaluation, 20 electrodes were selected from the motor cortex region, comprising FC-5/3/1/2/4/6, C-5/3/1/2/4/5, and CP-5/3/1/2/4/6. It is worth noting that the EEG data for 13 subjects (Nos. 32, 46, and 49) were omitted, reducing the total number of subjects to 49.

3) BNCI2014001 (Also Known as BCIC-IV-2a, From BNCI Horizon 2020): Its studies involved nine participants who executed an MI task comprising four classes: left hand, right hand, feet, and tongue. The task was recorded using 22 Ag/AgCl electrodes and three EOG channels at a sampling rate of 250 Hz. The recorded data were filtered between 0.5 and 100 Hz, with a 50-Hz notch filter applied. The study was conducted in two sessions on separate days, with each participant completing 288 trials in total (six runs of 12 cue-based trials for each class). The EEG signals were epoched from the cue’s onset at Second 2 to the end of the MI period at Second 6, resulting in a total duration of 4 s.

4) BNCI2014002 (From BNCI Horizon 2020): Thirteen participants were tasked with carrying out sustained 5-s kinesthetic MI of their right hand and feet, as directed by the cue. The session comprised eight runs consisting of 80 trials for each class, amounting to 160 trials per participant. EEG measurement was conducted using a biosignal amplifier and active Ag/AgCl electrodes with a sampling rate of 512 Hz. A total of 15 electrodes were placed. The EEGs were epoched from the cue onset at second 3 to the end of the MI period at second 8, resulting in a total duration of 5 s.

5) BNCI2015001 (From BNCI Horizon 2020): It involved 12 subjects who performed a sustained right-hand versus both feet imagery task. Most of the study was composed of two sessions, conducted on consecutive days, with each participant completing 100 trials for each class, amounting to 200 trials per participant. Four subjects participated in a third session, but the evaluation did not include these data. The data were acquired at a sampling rate of 512 Hz, utilizing a bandpass filter ranging from 0.5 to 100 Hz and a notch filter at 50 Hz. The recording began at Second 3 with the prompt and continued until the end of the cross period at Second 8, with a total duration of 5 s.

We typically evaluate datasets containing two sessions, namely KU, BNCI2014001, and BNCI2015001, using two distinct scenarios. The first is known as tenfold cross-validation (CV), whereby we divide each subject’s data into ten equally sized class-balanced folds, with nine used for training and one used for testing, and this process is repeated ten times. The second scenario is called Holdout, where we employ the first session for training and the second for testing. Given that the EEG data for the two sessions is usually gathered on different days, there may be considerable variability. For the KU dataset, we utilize the first 100 trials of the second session for validation, with the remaining trials used for testing. In contrast, datasets featuring only one session, such as BNCI2014002 and Cho2017, are evaluated solely using the tenfold CV scenario.

### B. Evaluated Segmentation Plans

This study presents an example of nonoverlapping and nonuniform segmentation plans, as shown in Table IV, to evaluate the proposed approach. The term “nonoverlapping” indicates no overlap between time windows, while “nonuniform” 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 IV, there are 60 segments of EEG signals in the KUBNCI2014002/BNCI2015001 datasets (i.e., 60 segments = 5 windows × 6 frequency bands + 10 windows × 3 frequency bands), 48 segments in the BNCI2014001 dataset (i.e., 48 segments = 4 windows × 6 frequency bands + 8 windows × 3 frequency bands), and 33 segments in the Cho2017 dataset (33 segments = 3 windows × 6 frequency bands + 5 windows × 3 frequency bands).

### C. Evaluation Baselines

This study will compare the proposed approach with a range of baseline methods, including FBCSP, FBCNet, and Tensor-CSPNet. The three selected baselines are representative...
algorithms utilizing the ERD/ERS effect for the MI-EEG classification during an MI process through various technological periods.

1) **FBCSP [66]:** This approach is a prominent method within the CSP family, which utilizes the CSP algorithm to extract subband scores from EEG signals and subsequently employs classification algorithms such as SVMs on the selected features.

2) **FBCNet [59]:** This approach is a convolutional neural networks-based classifier designed to classify EEG signals by analyzing EEG patterns within the space–time–frequency domain. Specifically, FBCNet performs on subfrequency bands of EEGs and is highly skilled at capturing complex EEG patterns.

3) **Tensor-CSPNet:** This approach represents the first geometric deep-learning approach to MI-EEG classification, capable of exploiting EEG patterns sequentially across the frequency, space, and time domains by leveraging existing deep neural networks on SPD manifolds.

It is worth noting that comparison with other prevalent MI-EEG classifiers, such as the convolutional neural networks-based approaches and Riemannian-based approaches, is not deemed necessary as previous studies [25] have established that FBCNet and Tensor-CSPNet exhibit superior performance on the subject-specific scenarios of datasets KU and BNCI2014001 under evaluation. Furthermore, while the proposed method is referred to as GNNs, it should be noted that it fundamentally differs from utilizing EEG device channels as graph nodes to construct a GNN in the context of MI-EEG classifiers. The method outlined in the text is the most suitable for comparison with the proposed method. Therefore, other graph-based MI-EEG classifiers will not be considered or adopted for this study.

### D. Configurations of Network Architecture

For the deep-learning methodology, the choice of neural network configuration plays a crucial role in the performance. Hence, we will present neural network configurations for all deep-learning-based approaches used in the evaluation.

For baseline FBCNet, it has relatively few neural network hyperparameters that can be tuned. We used the same nine frequency subbands as FBCSP and set 16 parallel spatial convolution blocks for all datasets. For the geometric classifiers, we have adopted simple yet effective network architectures for different scenarios as follows.

1) **Tensor-CSPNet:** The network architecture consists of a two-layer BiMap block (BiMap layer + ReEig layer + RBN layer), with input–output dimensions varying across different datasets. The frequency segmentation for all datasets is performed in 4-Hz bandwidth increments from 4 to 40 Hz without any overlap. The time segmentation, however, varies for each dataset, as follows. 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: 0-1.5, 0.5-2, and 1-2.5 s. 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 employs three temporal segmentations: 0-1, 1-2, and 2-3 s. 3) For the BNCI2014001 dataset, the BiMap block converts the input dimension from 20 to 30 and then reverts it to an output dimension of 20. The network employs three temporal segmentations: 0-1, 1-2, and 2-3 s.

---

**TABLE IV**

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

**TABLE V**

| 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) |
|         | Holdout (S1 → 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 (14.11) | 67.30 (13.92) | 67.51 (13.89) |
| BNCI2014001 | 10-Fold CV (T) % | 71.29 (16.20) | 75.48 (14.90) | 75.11 (13.68) | 77.55 (15.63) |
| (BCIC-IV-2a) | Holdout (T → 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 → B) % | 73.46 (14.09) | 74.50 (16.01) | 79.04 (14.67) | 79.75 (14.63) |
block transforms the input dimension from 22 to 36 and then back to an output dimension of 22. This network has two temporal segmentations: 0-0.75 and 0.25-1 s. 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: 0-1, 1-2, 2-3, 3-4, and 4-5 s. 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: 0-1, 1-2, 2-3, 3-4, and 4-5 s.

2) **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 IV. In addition, we typically set the forward time flow to half their maximum possible value, as discussed in Section IV-F.

### E. Classification Performance

The evaluation is conducted using subject-specific scenario settings. Two datasets, namely KU and BNCI2014001, have been reinitialized with cross-validation indices, employing different network architectures than those utilized in our previous study [25]. Consequently, the classification accuracies of each baseline may slightly vary from the earlier results within an acceptable range of approximately 1%. The remaining three datasets are being utilized for the first time in our study. For each scenario, we select the best result among multiple runs.

The time–space–frequency principle is fundamental to all three MI-EEG classification algorithms: FBCNet, Tensor-CSPNet, and Graph-CSPNet. Despite differences in network architectures, they all effectively utilize discriminative information from the time, space, and frequency domains. The segmentation technique employed by Tensor-CSPNet and Graph-CSPNet contributes to their superior performance over FBCNet in the three holdout scenarios. This technique breaks down EEG signals into short, quasistationary intervals, thereby reducing distribution shifts. Among the algorithms, Graph-CSPNet demonstrates a slight advantage in nine out of the 10 datasets listed in Table V. This is attributed to its refined time–frequency segmentation technique, which provides a more precise characterization of EEG signals.

### F. Hyperparameters in Graph-CSPNet

We will investigate the influence of hyperparameters in the time–frequency graph, such as $\epsilon$ in the LGT method, on the performance of Graph-CSPNet. This analysis is depicted in Fig. 2. To streamline the evaluation process, the network configuration of Graph-CSPNet will be simplified to include a depth-1 graph BiMap layer with input and output dimensions set to 20.

1) 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 “nongraph” case, that is, Time Direction (0,0,0,0) in Fig. 2. The “nongraph” case refers to the similarity matrix $A^{(0)} := I_N$.

2) Among different time directions, Time Direction (2,2,2,4) yields the best performance, as demonstrated in Fig. 2. The corresponding transformed spectral distributions can be found in Fig. 4(c). Additionally, Time Direction (2,2,2,4) showed statistical significance in the tenfold CV 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 [67] with Bonferroni–Holm correction, with a significance level of $\alpha = 0.05$. Therefore, in all of our experiments, we generally set the forward time flow to the nearest integer values, which is the closest approximation to half-values.

### G. 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 1, both
approaches will introduce changes in the distribution of the spectrum power of spatial covariance matrices.

In this section, we visually examine the extent of these changes by plotting a 2-D projection of the transformed spatial covariance matrices using the LGT method. This allows us to observe the shifts in their spectrum distributions.

In Fig. 3(a), the spatial covariance matrices of the 200th 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 200th 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, Fig. 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 Fig. 3(a).

H. Variant Parameter Configurations of Time–Frequency Graphs

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

Fig. 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 1.

Additionally, Fig. 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 Fig. 5(b) exhibits more connected edges compared to the one in Fig. 5(a).

V. DISCUSSIONS

A. Preset Edge Weights

The edge weights of the graph are determined by calculating the average Riemannian distances between EEG segments across all samples in the training set. In a tenfold CV scenario, the adjacency matrix A is initialized ten times, with slight differences between each fold. As a result, the adjacency matrix A is specific to each individual, and consequently, the architecture of Graph-CSPNet that it generates is also individual-specific.

B. Pseudocodes of Adjacency Matrix Generation

This section provides the pseudocode to generate an adjacency matrix A of the time–frequency graph. A lattice vector is employed to acquire the average EEG spectral covariance matrices within an assigned time and frequency interval in the training set.

Taking the example of the BNCI2014001 dataset, with a 1000-ms signal and nonoverlapping segmentations of...
250 and 125 ms, we obtain 4 and 8 windows, respectively, as shown in Table IV. As a result, there are 48 EEG segments (=4 time windows × 6 frequency bands + 8 time windows × 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 ≤ i ≤ 48.

The Riemannian distances in Algorithm 1 are calculated using the lattice vector. Specifically, the adjacency between two nodes is determined by a time-evolution mechanism. As shown in Table IV, the frequency bands are grouped into 1, 4, 4, 16, and 24 nodes, respectively.

The generation algorithm takes into account 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.

In pseudocode, “(N, w, x, y) ∈ [θ, µ, β, γ]” indicates the selection of one combination from the four frequency bands (θ, µ, β, γ). For example, if θ is chosen, the combination can be represented as (Nθ, wθ, xθ, yθ) with the θ component. To facilitate readability, we have omitted the subscripts. The “Number of Nodes (Nθ, Nµ, Nβ, Nγ)” refers to the number of graph nodes belonging to a specific frequency component. The “Length of Time Horizon (wθ, wµ, wβ, wγ)” corresponds to the number of time windows in the respective frequency component, with each frequency component having different time resolutions, resulting in varying window counts. The inclusion of the maximum forward steps, denoted as T, is necessary because sometimes the distance to the rightmost position (the maximum time step) is less than one forward time step. The expression “i mod w” represents the remainder when i is divided by w. When assigning values to the time and frequency directions, the usual convention is to first fix the frequency and move along the positive time direction. Then, within the same frequency band, higher frequencies are considered. Refer to our GitHub repository mentioned in the abstract for specific code examples.

Algorithm 1 Adjacency Matrix Generation

| Input | Number of Nodes (Nθ, Nµ, Nβ, Nγ); Length of Time Horizon (wθ, wµ, wβ, wγ); Time Direction (xθ, xµ, xβ, xγ); Frequency Direction (yθ, yµ, yβ, yγ). |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Output| Adjacency Matrix A[i*, j*]. |

Initialization of the lattice vector and start node s ← 1;

for (N, w, x, y) ∈ [θ, µ, β, γ] do

for i ← {1, . . . , N} do

/* Initialize the first coordinate for A. */
i* ← s + i;

/* Compute the largest forward time step. */
T ← min{w − i mod w − 1, x};

/* The time direction. */
if j ∈ {i + 1, . . . , i + T + 1} then

j* ← s + j;

A[i*, j*] ← exp(−d2AI RM(Si*, Sj*)/t);

/* The frequency direction. */
if j ∈ {i + w, . . . , i + w + T + 1} then

j* ← s + j;

A[i*, j*] ← exp(−d2AI RM(Si*, Sj*)/t);

A[j*, i*] ← A[i*, j*];

s ← s + N.

In image processing, the Laplacian operator is used to calculate the second-order derivative of an image and is effective in highlighting areas of rapid intensity changes or edges in the image. Similarly, in the context of spectral clustering in the time–frequency graph, we have a similar interpretation. By employing an overlapping segmentation approach instead of a nonoverlapping one, we can capture more detailed local information. As the number of segments increases, spectral clustering places greater emphasis on regions of significant intensity changes on the SPD manifold, which is constructed based on statistical measures in the time–frequency domain.

Specifically, given an infinity segmentation plan that spatial covariance matrices [Si]N i=1 ∈ SD+ + of the time–frequency graph around test trial Š are uniformly Gaussian distributed on SPD manifolds given in [51], the discrete graph Laplacian LM = I − A of the time–frequency graph on networks-based function f will converge to the continuous Laplacian ΔM(f) with the bias term, established by many studies [69], [70], [71], as follows, ∑N i=1 LMF(Si)/ε = (1/2)ΔM f(Š) + O(ε1/2).
D. Geometric Deep Learning

Geometric deep learning constitutes a subfield of machine learning that focuses on developing algorithms and models that can process non-Euclidean structured data, in particular, data represented as graphs and manifolds [72].

GNNs are widely recognized as a primary category in geometric deep learning and have been extensively applied to solve a broad range of problems in various domains, such as traffic networks, graph-based recommender systems, molecule design, and so on [73]. While the network architectures for these applications are typically designed in Euclidean spaces, there has been growing interest in GNNs in non-Euclidean spaces, or alternatively referred to as manifold-valued GNNs [74], [75], [76].

Our scenario precisely fits into the category of GNN problems in a non-Euclidean space, as each node corresponds to an EEG spatial covariance matrix, naturally represented as points on SPD manifolds. The key aspect of our problem lies in the utilization of the BiMap layer in neural networks, which incorporates the BiMap transformation to preserve the discriminative power between task classes, as characterized by the Riemannian distance (see (1), (2), and Table II). Additionally, the graph structure provides a neural network-based framework for propagating information from different time–frequency EEG segments concurrently, guided by principles from neurophysics.

E. ReEig Layer

Let $S$ be a real symmetric $d \times d$ matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ and corresponding orthonormal eigenvectors $\{u_i\}_{i=1}^d$. Then, the spectral decomposition of $S$ is given as $S = \sum_{i=1}^d \lambda_i u_i u_i^\top$. To handle the non-negative eigenvalues in symmetric matrices, we introduce the following lemma, an essential technique in convex optimization [77].

Lemma 3: Projection $S^\dagger := \sum_{i=1}^d \max \{\lambda_i, 0\} u_i u_i^\top$ on the positive semidefinite cone is the extremum of the minimization problem $\|S - S^\dagger\|_2$ with respect to $S \succeq 0$.

Proof: Without loss of generality, let $\lambda_d < 0$ and define the projection $S^\dagger := \sum_{i=1}^d \max \{\lambda_i, 0\} u_i u_i^\top$. It follows that $\|S - S^\dagger\|_2 = -\lambda_d$. To prove this lemma, it suffices to demonstrate that $\|S - S^\dagger\|_2 \geq -\lambda_d$. Since we have $\|S\|_2 = \max \{\lambda_1, -\lambda_d\} = \max \{u_1^\top Su_1, -u_d^\top Su_d\} = \max \{\sup_{|u|_2=1} u^\top Su, \inf_{|u|_2=1} -u^\top Su\}$, then it yields that $\|S - S^\dagger\|_2 \geq \sup_{|u|_2=1} u^\top (S - S^\dagger) u \geq u_d^\top (S - S^\dagger) u_d \geq -u_d^\top Su_d \geq -\lambda_d$. \hfill $\square$

By Lemma 3, we obtain a means to eliminate the insignificant eigenvalues of SPD matrices. In practical applications, we establish a lower bound, $\beta$, with a value greater than 0 to ensure that all eigenvalues are not less than $\beta$, that is, $S^\dagger := \sum_{i=1}^d \max \{\lambda_i, \beta\} u_i u_i^\top$. In the implementation, we consistently set the small eigenvalue as $\beta = 1e - 6$. Roughly speaking, the choice of different magnitudes does not have a significant impact on the performance in the presented scenarios.

F. Architecture of Graph-CSPNet

Table VI summarizes the layers and learnable network parameters in Graph-CSPNet. The total number of learnable network parameters is $N_0(\alpha_1 + \alpha_2) + (cN + 1)\alpha_2^2$. For instance, in the case of BNCI2014001, the network configuration is given by $N = 48$, $c = 4$, $\alpha_1 = 22$, $\alpha_2 = 36$, and $\alpha_3 = 22$, where $\alpha_1$ and $\alpha_2$ are the input and output dimensions for the first graph BiMap layer, respectively, $\alpha_2$ and $\alpha_3$ 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$^{(9,1)}$ (27,104 parameters) and almost two-thirds of the parameters in 5-CSPNet$^{(9,3,1)}$ (232,360 parameters).

G. Optimization on Smooth Manifolds

Due to the parameterization of neural networks in both geometric classifiers involving manifolds, specialized optimizers that operate on manifolds are employed to enhance performance. Specifically, the BiMap and graph BiMap layers have parameters defined on Stiefel manifolds, while the Riemannian Batch Normalization utilizes parameters defined on SPD manifolds. To address the issue of local minima, first-order stochastic optimization methods, such as Adam and AdaGrad, which are commonly used in the Euclidean domain, need to be adapted to the corresponding Riemannian manifolds to accommodate the manifold constraints. This is crucial because the deep-learning approach exhibits high nonconvexity throughout its domain [78].

To meet this requirement, specialized first-order Riemannian adaptive optimization methods have been developed to optimize networks with manifold-valued data [79]. The generalization of adaptive optimization methods involves two essential operators: retraction and parallel transport, which are
established within the framework of Riemannian optimization [80]. Riemannian optimization is a field of mathematical optimization that deals with optimization problems subject to manifold constraints, that is, \(\min_{x \in \mathcal{M}} f(x)\), where \(\mathcal{M}\) is a Riemannian manifold and \(f: \mathcal{M} \to \mathbb{R}\) is a smooth function. Retraction is a first-order approximation of the exponential map in the manifold space, while parallel transport refers to transporting vectors along smooth curves in manifolds equipped with an affine connection. These two operators, within the manifold domain, enable neural networks to perform gradient descent with improved precision in each update iteration.

H. Time-Domain Statistics and Frequency-Domain Statistics

In this section, we focus on exploring the similarities and differences between the statistical characteristics of EEG signals in the time and frequency domains. Typically, the analysis of EEG signals involves extracting statistical features through spatial, temporal, and spectral analyses, either individually or in combination. A survey paper [81] classifies these features into three categories: spatial, frequency (time–frequency), and similarity. These categories encompass various statistical measures obtained from both the time and frequency domains.

1) Spatial Features: These features primarily involve time-domain statistics and can be obtained through supervised methods such as principal component analysis, independent component analysis, and common spatial patterns or through unsupervised methods like the common-average reference and surface Laplacian spatial filters.

2) Frequency Features: These features focus on frequency-domain statistics and include measures such as band power, fast Fourier transform, and wavelet analysis.

3) Similarity Features: This category encompasses features that capture the similarity between EEG signals, with some of these features being frequency-domain statistics. Examples of similarity features include phase-locking value and magnitude-squared.

This section aims to demonstrate the equivalence between time-domain statistics, represented by spatial covariance matrices, and certain frequency-domain statistics, such as band power and magnitude-squared coherence. We argue that classifiers utilizing spatial covariance matrices in MI-EEG classification can capture information comparable to that obtained from other sources, including phase locking value and magnitude-squared coherence.

This relationship has been previously explored in the literature, as discussed in [27]. It has been observed that various frequency-domain features often reflect the same underlying neurophysiological phenomenon during a task. However, this fundamental fact has yet to be explicitly emphasized in previous works, and its recognition is crucial for the successful implementation of geometric classifiers.

Suppose a wide-sense stationary real-valued random process \(X_t\) with zero mean \(\mathbb{E}(X_t) = 0\). The autocorrelation function \(R_X(\tau)\) of \(X_t\) is defined as \(R_X(\tau) := \mathbb{E}(X_t^T X_{t+\tau})\), 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. Hence, the variance of zero-centered \(X_t\) is \(\sigma^2_X := \mathbb{E}|X_t - \mathbb{E}(X_t)|^2 = P_X\). This argument applies to each channel’s time-series \(X_t\) in multichannel EEG data. Thus, the diagonal entries of the spatial covariance matrices correspond to the variance \(\sigma^2_X\), which is equivalent to the geometric classifiers expected (band) power \(P_X\) of that channel \(c\). Using a similar line of reasoning based on the autocovariance function, it can be demonstrated that the off-diagonal entries of the spatial covariance matrices are equivalent to the magnitude-squared coherence without normalization. In practice, implementing a neural network-based approach using nonzero-centered time series yields satisfactory classification performance, and there is no need to preprocess the time series by zero-centering them.

VI. Conclusion

The primary objective of this study is to enhance the geometric classifiers in MI-EEG classification from a time–frequency analysis perspective, aiming to capture the time–frequency information of signals more effectively. Furthermore, this article formally introduces the term geometric classifiers as the name for this category for the first time. To achieve this goal, we present a novel network architecture called Graph-CSPNet based on an SPD matrix-valued time–frequency graph. In Graph-CSPNet, we extend graph convolutional networks to SPD manifolds equipped with \(g^{\text{ARM}}\) and utilize them to locate EEG rhythmic components in the time–frequency domain precisely. In the graph BiMap layer, we utilize preset graph weights calculated using Riemannian distances to effectively preserve discriminative information of different time–frequency spatial covariance matrices, allowing Graph-CSPNet to inherit the characteristics and advantages of several well-known MI-EEG classifiers.

In sum, the study’s contribution is twofold, providing both theoretical and practical advances as follows: Theoretically, Graph-CSPNet signifies an improvement in the feasibility of existing models, approached from the fundamental principle of time–frequency analysis in signal processing. Moreover, the potential value of this study lies in proposing a category of methods called Geometric Classifiers that have a foundation in differential geometry within the era of deep learning. These geometric classifiers are more capable of capturing the characteristics of signal covariance matrices and offer greater possibilities associated with advanced mathematical and physical tools. By incorporating concepts from differential geometry, these methods provide a promising avenue for advancing the understanding and analysis of complex neuro-physiological data, such as EEG spatial covariance matrix and fMRI functional connectivity matrix, within the framework of geometric deep learning. Practically, Graph-CSPNet provides enhanced flexibility in handling variable time–frequency resolution for signal segmentation and capturing localized fluctuations. It performs well on five widely used MI-EEG datasets, utilizing tenfold cross-validation and subject-specific holdout scenarios. In nine out of 11 scenarios, Graph-CSPNet achieves near-optimal results, showcasing improved classifier performance and offering valuable practical advancements.
REFERENCES

[1] J. R. Wolpaw et al., “Brain–computer interface technology: A review of the first international meeting,” IEEE Trans. Rehabil. Eng., vol. 8, no. 2, pp. 164–173, Jun. 2000.

[2] H. Yuan and B. He, “Brain–computer interfaces using sensorimotor rhythms: Current state and future perspectives,” IEEE Trans. Biomed. Eng., vol. 61, no. 5, pp. 1425–1435, May 2014.

[3] G. Pfurtscheller and F. H. L. da Silva, “Event-related EEG/MEG synchronization and desynchronization: Basic principles,” Clin. Neurophysiol., vol. 110, no. 11, pp. 1842–1857, Nov. 1999.

[4] G. Pfurtscheller and C. Neuper, “Motor imagery and direct brain–computer communication,” Proc. IEEE, vol. 89, no. 7, pp. 1123–1134, Jul. 2001.

[5] J. Wolpaw and E. W. Wolpaw, Brain-Computer Interfaces: Principles and Practice, G. Pfurtscheller and D. J. McFarland, Eds.: Oxford Univ. Press, 2012, ch. 3.

[6] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregeren, “EEG-based discrimination between imagination of right and left hand movement,” Electroencephalogr. Clin. Neurophysiology, vol. 103, no. 6, pp. 642–651, Dec. 1997.

[7] M. Jeannerod, “The representing brain: Neural correlates of motor intention and imagery,” Behav. Brain Sci., vol. 17, no. 2, pp. 187–202, Jun. 1994.

[8] R. Beisteiner, P. Höllinger, G. Lindinger, W. Lang, and A. Berthoz, “Mental representations of movements. Brain potentials associated with imagination of hand movements,” Electroencephalogr. Clin. Neurophysiology/Evoked Potentials Sect., vol. 96, no. 2, pp. 183–193, Mar. 1995.

[9] F. Lotte et al., “A review of classification algorithms for EEG-based brain–computer interfaces: A 10 year update,” J. Neural Eng., vol. 15, no. 3, Jun. 2018, Art. no. 031005.

[10] L. Qin, B. Kamoussi, Z. M. Liu, L. Ding, and B. He, “Classification of motor imagery tasks by means of time-frequency-spatial analysis for brain-computer interface applications,” in Proc. 2nd Int. IEEE EMBS Conf. Neural Eng., 2005, pp. 374–376.

[11] T. C. Ferree, M. R. Brier, J. Hart Jr., and M. A. Kraut, “Space–time–frequency analysis of EEG data using within-subject statistical tests followed by sequential PCA,” Neuroimage, vol. 45, no. 1, pp. 109–121, Mar. 2009.

[12] A. P. Allen and C. D. MacKinnon, “Time–frequency analysis of movement-related spectral power in EEG during repetitive movements: A comparison of methods,” J. Neurosci. Methods, vol. 186, no. 1, pp. 107–115, Jan. 2010.

[13] S. Sanei and J. A. Chambers, EEG Signal Processing. Hoboken, NJ, USA: Wiley, 2013.

[14] S. Blanco, R. Q. Quiroga, O. A. Rosso, and S. Kochen, “Time–frequency analysis of electroencephalogram series,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 51, no. 3, pp. 2624–2631, Mar. 1995.

[15] B. J. Roach and D. H. Mathalon, “Event-related EEG time-frequency analysis: An overview of measures and an analysis of early gamma band phase locking in schizophrenia,” Schizophrenia Bull., vol. 34, no. 5, pp. 907–926, Jul. 2008.

[16] A. T. Trillas, M. G. Taipouras, and D. I. Fotiadis, “Epileptic seizure detection in EEGs using time–frequency analysis,” IEEE Trans. Inf. Technol. Biomed., vol. 13, no. 5, pp. 703–710, Sep. 2009.

[17] S. Morales and M. E. Bowers, “Time–frequency analysis methods and their application in developmental EEG data,” Develop. Cognit. Neurosci., vol. 54, Apr. 2022, Art. no. 101067.

[18] H. Adeli, Z. Zhou, and N. Dadmehr, “Analysis of EEG records in an epileptic patient using wavelet transform,” J. Neurosci. Methods, vol. 123, no. 1, pp. 69–87, Feb. 2003.

[19] A. Subasi, “EEG signal classification using wavelet feature extraction and a mixture of expert model,” Exp. Syst. Appl., vol. 32, no. 4, pp. 1084–1093, May 2007.

[20] H. Adeli, S. Ghosh-Dastidar, and N. Dadmehr, “A wavelet-chaos methodology for analysis of EEGs and EEG subbands to detect seizure and epilepsy,” IEEE Trans. Biomed. Eng., vol. 54, no. 2, pp. 205–211, Feb. 2007.

[21] N. Robinson, A. P. Vinod, K. K. Ang, K. P. Lee, and C. T. Guan, “EEG-based classification of fast and slow hand movements using wavelet-CSP algorithm,” IEEE Trans. Biomed. Eng., vol. 60, no. 8, pp. 2123–2132, Aug. 2013.

[22] D. Gabor, “Theory of communication. Part 1: The analysis of information,” J. Inst. Elect. Engineers III, Radio Commun. Eng., vol. 93, no. 26, pp. 429–441, Nov. 1946.
