# MAtt: A Manifold Attention Network for EEG Decoding

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Recognition of electroencephalographic (EEG) signals highly affect the efficiency of non-invasive brain-computer interfaces (BCIs). While recent advances of deep-learning (DL)-based EEG decoders offer improved performances, the development of geometric learning (GL) has attracted much attention for offering exceptional robustness in decoding noisy EEG data. However, there is a lack of studies on the merged use of deep neural networks (DNNs) and geometric learning for EEG decoding. We herein propose a manifold attention network (mAtt), a novel geometric deep learning (GDL)-based model, featuring a manifold attention mechanism that characterizes spatiotemporal representations of EEG data fully on a Riemannian symmetric positive definite (SPD) manifold. The evaluation of the proposed mAtt on both time-synchronous and -asyncronous EEG datasets suggests its superiority over other leading DL methods for general EEG decoding. Furthermore, analysis of model interpretation reveals the capability of mAtt in capturing informative EEG features and handling the non-stationarity of brain dynamics.

## 1 Introduction and related works

A brain-computer interface (BCI) is a type of human-machine interaction that bridges a pathway from brain to external devices. Electroencephalogram (EEG), a non-invasive neuromonitoring modality with high portability and affordability, has been widely used to explore practical applications of BCI in the real world [1, 2, 3]. For instance, disabled users can type messages through an EEG-based BCI that recognizes the steady-state visual evoked potential (SSVEP) induced by flickering visual targets presented on a screen [4, 5, 6]. Stroke patients who need restoration of motor function undergo motor-imagery (MI) BCI-controlled rehabilitation as an active training [7, 8]. Most EEG-based BCI systems are designed to detect/recognize reproducible time-asynchronous or time-synchronous EEG patterns of interest, depending on the schemes of BCI [9]. For example, the MI EEG pattern is an endogenous oscillatory perturbation without an explicit onset time sources from the motor cortex [10]. On the other hand, a time-synchronous EEG pattern is time-locked to a specific event. For example, the pattern of SSVEP is synchronized to the change of brightness on a flickering visual target. The efficiency of BCI systems largely relies on the accuracy and robustness of the EEG decoder. However, due to the low signal-to-noise ratio (SNR) [11] and non-stationarity [12] of EEG, translating perplexing EEG signals into meaningful information has been a grand technology and scientific challenge in the field.

Recent advances in deep learning (DL) have contributed to the rapid development of DL-based EEG decoding techniques [13]. DL models are capable of extracting features automatically according to

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

given training data. Convolutional neural network (CNN) is one type of the most common DL models and has achieved remarkable performance in tasks such as image recognition and object detection [14, 15, 16]. CNN models newly designed for EEG decoding use convolutional kernels that serve as spatial and temporal filters but with extra flexibility to optimize the transformation of EEG data automatically through model training [17, 18, 19]. In addition to the fast growth of DL-based EEG decoders, geometric learning (GL) approaches, mostly based on Riemannian geometry (RG), have been adopted in the field of BCI [20]. RG is a type of non-Euclidean geometry that has a different interpretation of Euclid's fifth postulate (i.e. parallel postulate) [21]. In GL, geodesic between points on the manifold is a critical feature for classification tasks in BCI. The power and spatial distribution of a segment of multi-channel EEG signals can be coded into a covariance matrix that is symmetric positive definite (SPD) in general. This allows mapping of EEG data directly onto a Riemannian manifold where Riemannian metric is insensitive to extreme outliers and noise [22, 20]. In 2010, Barachant et al. [23] proposed Minimum Distance to Mean (MDM) that maps target EEG data onto the SPD manifold to find the nearest class center. Later on, they developed TSLDA [24] that projects data from the manifold to a specific tangent space where Euclidean classifiers are applicable. RG-based classification for EEG decoding has shown extra robustness as the relationship between data samples can be stably preserved, leading to success in recent data competitions in the BCI field such as 'DecMEG2014'[1] and the 'BCI challenge'[2].

The nascent field of geometric deep learning (GDL) has expanded by emerging techniques to generalize the use of deep neural networks to non-Euclidean spaces. Efforts have been made to transitioning useful operations from Euclidean to Riemannian spaces, including convolution [25, 26, 27], activation function [25, 26], batch normalization [28, 29], that facilitate the ongoing development of GDL tools. SPDNet [25] is a Riemannian network for non-linear SPD-based learning on Riemannian manifolds using bi-linear mapping that mimics Euclidean convolution for visual classification tasks. ManifoldNet [26] offers high performance in medical image classification with manifold autoencoder. [30] characterizes 3D movement via the manifold polar coordinate with a geodesic CNN. [31] performs convolution on the manifold as a generalization of local graph or manifold pseudo-coordinate for vertex classification on graph and shape correspondence task. In contrast of the vast develop of geometric deep learning (GDL) in many other scientific fields, only few studies focus on decoding EEG data with a merge use of GL and DL. [32] proposed a network architecture that integrates fusion of Euclidean-based module and manifold-based module with multiple LSTM and attention structures to extract spatiotemporal information of EEG. [33] proposes a Riemannian-embedding-banks method that separates the entire embeddings into multiple sub-problems for learning spatial patterns of MI EEG signals based on the features extracted from the SPDNet. [34] combines federated learning and transfer learning on Riemannian manifold using the spatial information of EEG. [35] proposes deep optimal transport on the manifold to minimize the cost of domain adaptation from the source domain to the target domain. [36] extracts multi-view representations of EEG. These studies have established cornerstones toward the field of future GDL for EEG decoding, but the increment of performance is yet marginal. Most of the above-mentioned techniques can not map the temporal information of EEG onto the manifold, or still rely on Euclidean tools to handle EEG features. We herein propose a manifold attention network, a novel GDL framework, which maps EEG features on a Riemannian SPD manifold where the spatiotemporal EEG patterns are fully characterized. The main contributions of the present study are the following:

- a manifold attention network proposed for decoding general types of EEG data.
- a lightweight, interpretable, and efficient GDL framework that is capable of capturing spatiotemporal EEG features across Euclidean and Riemannian spaces.
- an empirical validation of our proposed model demonstrating its generalizable superiority over leading DL approaches in EEG decoding.
- neuroscientific insights interpreted by the model that not only echo prior knowledge but also offer a new look into the dynamical brain.

[1] DecMEG2014: https://www.kaggle.com/competitions/decoding-the-human-brain/leaderboard
[2] BCI challenge: https://www.kaggle.com/c/inria-bci-challenge

This article is organized as follows: we first brief the essential background of RG and manifold attention mechanism; next, we describe the proposed mAtt architecture with details of model design and training; we then validate our proposed model experimentally; lastly, we interpret our proposed model with neuroscientific insights.

# 2 Preliminary

A manifold is considered as the expansion of curve and surface in Euclidean space. It is a topological space that can locally regarded as an open set in Hilbert space. Suppose a manifold is endowed with a differential structure (i.e. a collection of charts satisfying transition mapping, which is defined on the overlap of charts), it is then the so-called differential manifold [37]. Riemannian geometry is a differential manifold equipped with Riemannian metric. We consider the Symmetric positive definite (SPD) manifold, which allows us to manipulate manifold-valued data on the manifold directly. The spatial information of EEG signal can be represented as a specific covariance matrix, which records the relationship between channels, and is a critical index for us to understand EEG signal. However, the solution of the Riemannian mean doesn't have a close form, thus we need to calculate the approximate mean in an iteration manner [23, 26] until convergence conditions are satisfied. Riemannian mean is not suitable for being applied in deep learning because of the high computational complexity. Therefore, we have the Log-Euclidean metric below.

## 2.1 Notations

$GL(n, \mathbb{R}) := \{A \in \mathbb{R}^{n\times n} \,|\, determinant(A) \neq 0\}$ is a general linear group, which is the set of all real non-singular sqaure matrices. $(\mathcal{M}, g)$ denotes connected Riemannian manifold. $Sym(n) := \{S \in M_{n\times n}(\mathbb{R}) \mid S^T = S\}$ is the space of all $n \times n$ real symmetric matrices, where $M_{n\times n}(\mathbb{R})$ specifies the space of all real square matrices, $(.)^T$ is the *transpose* operator, and $Sym^+(n) := \{P \in M_{n\times n}(\mathbb{R}) \mid P = P^T, v^T P v > 0, \forall v \in \mathbb{R}^n - \{0\}\}$ is the set of all $n \times n$ symmetric positive definite(SPD) matrices.$< A, B >_F$ means the Frobenius inner product, defined as $Tr(A^T B)$, where $Tr(.)$ is the *trace* operator. $Log(.)$ and $Exp(.)$ are the *principle logrithm* operator for SPD matrix[38] and *exponential* operator for symmetric matrix respectively. Both of them can be computed using the *orthogonal diagonalization*. $Log : Sym^+(n) \mapsto Sym(n)$ is an operator that maps a SPD matrix $P \in Sym^+(n)$ to $Sym(n)$ by:

$$Log(P) = U diag(log(\sigma_1), ..., log(\sigma_n))U^T \tag{1}$$

where $U$ is the matrix of eigenvectors of $P$, since $P \in Sym^+(n)$, $\sigma_i > 0, i = 1, ..., n$

The inverse projection is $Exp$ of symmetric matrix: $Exp : Sym(n) \mapsto Sym^+(n)$, an operator maps a symmetric matrix $S \in Sym(n)$ to $Sym^+(n)$ by:

$$Exp(S) = V diag(exp(\sigma_1), ..., exp(\sigma_n))V^T$$

where $V$ is the matrix of eigenvectors of $S$.

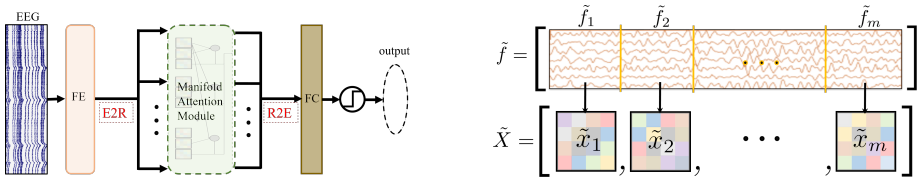
## 2.2 Log-Euclidean metric

Log-Euclidean metric (LEM) offers a more simple, similar, and efficient generalization to calculate the center on the SPD manifold than Affine-invariant metric (AIM) [39, 40]. LEM is a bi-invariant metric on the Lie group on the SPD manifold [40]. The geodesic distance from $P_1$ to $P_2$ on the $Sym^+(n)$ is also given by [40]:

$$\delta_L(P_1, P_2) = \|Log(P_1) - Log(P_2)\|_F \tag{2}$$

Furthermore, we can also define the Log-Euclidean mean($\mathcal{G}$) via the Log-Euclidean distance:

$$\mathcal{G}(P_1, ...P_k) = \underset{P\in Sym^+(n)}{\arg\min} \sum_{l=1}^{k} \delta_L^2(P, P_l) \text{ where } P_1, ..., P_k \in Sym^+(n).$$




(a) (b)

Figure 1: (a) The overview of the proposed model architecture. (b) E2R operation: split latent feature into several epochs, and convert each one to a specific SPD matrix.

Fortunately, the solution to the formula above has a closed form to follow, given by [41]:

$$\mathcal{G} = Exp\left(\frac{1}{k}\sum_{l=1}^{k} Log(P_l)\right)$$

In this work, we are going to use the weighted Log-Euclidean mean that is endowed with different weights in different $P_l$. We denote the weight of each $P_l$ as $w_l$, where $\forall l \in \{1, 2, ..., k\}$. Here, $\{w_l\}_{l=1}^{k}$ satisfies the *convexity constraint* definition (i.e. $\sum_{l=1}^{k} w_l = 1$, and $w_l > 0$). The definition and the corresponding weighted Log-Euclidean mean can be defined and derived as:

$$\mathcal{G}(P_1, ...P_k) = \underset{P \in Sym^+(n)}{\arg\min} \sum_{l=1}^{k} w_l \delta_L^2(P, P_l)$$

and

$$\mathcal{G} = Exp\left(\sum_{l=1}^{k} w_l Log(P_l)\right)$$

respectively.

# 3 Methodology

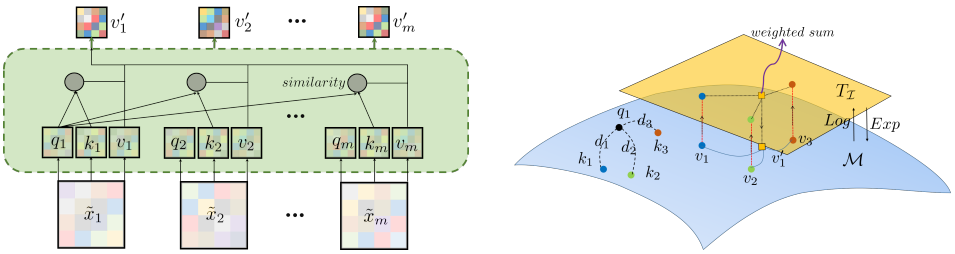
As shown in Figure 1(a), the architecture of mAtt includes components of the feature extraction (FE), the manifold attention module, transitioning from Euclidean to Riemannian space (E2R), and transitioning from Riemannian to Euclidean space (R2E).

## 3.1 Feature extraction of EEG signals

We use two convolutional layers to extract information of the raw EEG signals, where the first convolutional layer performs spatial filtering to the multi-channel EEG signals and the second convolutional layer extracts spatiotemporal features. Our parameter setting follows [19].

## 3.2 From Euclidean space to SPD manifold(E2R operation)

As illustrated in Figure1 (b), we convert the embeddings from the feature extraction stage to the SPD data and map the feature embeddings from Euclidean space to the SPD manifold. Suppose $\tilde{f}$ denotes the embeddings after the feature extraction stage, we divide the whole embeddings into several epochs $\tilde{f}_1, \tilde{f}_2, ..., \tilde{f}_m$, and calculate the sample covariance matrix(SCM) of each $\tilde{f}_i, \forall i \in \{1, 2, ..., m\}$. By doing so, we get a sequence of covariance matrices that present the temporal information of the embeddings $\tilde{f}$ in the form of SPD data, called $SCM_{\tilde{f}_1}, SCM_{\tilde{f}_2}, ..., SCM_{\tilde{f}_m}$. After we get some datapoints, we do trace-normalization and add a small number $\epsilon$ on each main diagonal element for




(a) (b)

Figure 2: (a) The architecture of the proposed manifold attention module. $q_i, k_i, v_i$ refer to the query, key, and value of the $i^{th}$ input matrix $\tilde{x_i}$ respectively; $v_i'$ stands for the $i^{th}$ output of the proposed module. (b) Illustration of the operation of Log-Euclidean mean used in proposed module as $i = 1$ and number of epoch is 3; $q_i$ and $k_j$ refer to $i^{th}$ query and $j^{th}$ key respectively; $d_j$ denotes the distance between $q_1$ and $k_j$ on the SPD manifold $\mathcal{M}$; $T_{\mathcal{I}}$ refers to the tangent space based on identity matrix $\mathcal{I}$.

each $SCM_{\tilde{f}_i}$(i.e. $\frac{SCM_{\tilde{f}_i}}{tr\left(SCM_{\tilde{f}_i}\right)} + \epsilon I$) where $i \in \{1, 2, ..., m\}$, $I$ is the identity matrix, and we set $\epsilon$ as 1e-5 in our source code. The resulting SPD sequence is denoted as $\tilde{X} = [\tilde{x_1}, \tilde{x_2}, ..., \tilde{x_m}]$. We add a small identity matrix on them to promise $\tilde{x_i}$ to be a well-defined SPD matrix.

### 3.3 Manifold attention module

The input of this layer is a sequence of SPD data. The overview of the manifold attention module is illustrated in Figure 2 (a). Motivated by [25] and [42], we capture the spatiotemporal information on the manifold. Suppose the module takes a sequence of SPD matrices $[\tilde{x_1}, \tilde{x_2}, ..., \tilde{x_m}]$, denoted as $\tilde{X}$.

Here we have the query, key, and value in the form of SPD matrices on the manifold [42]. We convert the $\tilde{x_i}$ to the $q_i, k_i$, and $v_i$ via bilinear mapping and exploit non-linear and valid features from each segment. Suppose the shape of $\tilde{x_i}$ is $d_c \times d_c$, and $h_q, h_k$, and $h_v$ is the mapping from $\tilde{x_i}$ to $q_i, k_i$, and $v_i$ respectively:

$$
\begin{aligned}
q_i &= h_q(\tilde{x_i}; W_q) = W_q \tilde{x_i} W_q^T \\
k_i &= h_k(\tilde{x_i}; W_k) = W_k \tilde{x_i} W_k^T \\
v_i &= h_v(\tilde{x_i}; W_v) = W_v \tilde{x_i} W_v^T
\end{aligned}
$$

where $\tilde{x_i} \in Sym^+(d_c)$, $W_q, W_k$, and $W_v \in \mathbb{R}^{d_u \times d_c} (d_u < d_c)$ denotes transformation matrices. To make sure the output $q_i, k_i$, and $v_i$ are also SPD matrices, transition matrices $W_q, W_k$, and $W_v$ are row-full rank matrices.

After we got $q_i, k_i$, and $v_i$ by bilinear mapping, we define the similarity for measuring the $q_i$ and $k_j$ SPD matrices. In Euclidean space, there are several ways to define the similarity. A most common way is to use dot-product[42] to measure the similarity of query and key. However, our query, key, and value are SPD matrices instead of vectors as regular attention. We define the similarity based on the Log-Euclidean distance (equation(2)) between query and key. Suppose we have $q_i$ and $k_j$, for some $i, j \in \{1, ..., m\}$. The similarity $sim(.)$ is a $strictly\ decreasing\ function$ of distance $[0, \infty) \mapsto [0, 1]$ and is defined as: $sim(q_i, k_j) = \frac{1}{1+log(1+\delta_L(q_i, k_j))} := \alpha_{ij}$. Then, the attention matrix is:

$$
\mathcal{A} = [\alpha_{ij}]_{m \times m}
$$

We then use $Softmax$ function to shrink the range along the row direction, making values in row have $convexity\ constraint$ property. The final attention probability matrix $\mathcal{A}'$ is:

$$
\mathcal{A}' = Softmax(\mathcal{A}) = Softmax([\alpha_{ij}]_{m \times m}) = [\alpha'_{ij}]_{m \times m}
$$

where $\alpha'_{ij} = \frac{exp(\alpha_{ij})}{\sum_{k=1}^{m} exp(\alpha_{ik})}$ , $\forall i, j \in 1, \cdots, m$. Finally, we combine the attention probability matrix and $v_1, v_2, ..., v_m$ to get the final output $v'_1, v'_2, ..., v'_m$ and define the output $v'_i (\forall i = 1, 2, ..., m)$ via Log-Euclidean mean as:

$$v'_i = Exp\left(\sum_{l=1}^{m} \alpha'_{il} Log(v_l)\right)$$

The whole mAtt procedure is illustrated in Algorithm 1 in appendix.

As shown in Figure2(b), the output $v'_1$ of our attention module can be comprehended as a projection that translates the weighted sum, on the tangent space, of three different matrices encoded by three different epochs and corresponding attention scores $\alpha'_{11}, \alpha'_{12}, \alpha'_{13}$ (or weights) to a specific representative matrix $v'_1$ on SPD manifold. Herein the weights for the weighted sum on the tangent space is assigned by its query matrix $q_1$ and corresponding keys $k_1, k_2, k_3$ to generate the relevance score between $q_1$ and $k_1, k_2, k_3$ [42, 43].

### 3.4 From Riemannian manifold to Euclidean space(R2E) and loss layers

After passing through attention module, ReEig layer is used to imitate the ReLU function. But it is different from the ReLU function that sets the threshold to the value of input, ReEig sets a threshold to the eigenvalue of the input, can be defined in [25]. R2E operation aims to map the SPD data back to the Euclidean space for executing the final classification, which is composed of a Log layer and regular flatten layer in [25] sequentially. Log layer is the most common skill in geometric deep learning to project the SPD data to the Euclidean space. By doing so, we reduce the manifold to a flat space by $Log(.)$ operation. We take $Log(.)$ operation on the output from the attention module layer $v'_1, v'_2, ..., v'_m \in Sym^+(d_u)$. Denote the whole R2E operation as $h_L \colon Sym^+(d_u) \mapsto \mathbb{R}^{d_u \times (d_u+1)/2}$:

$$h_L(v'_i) = flatten\left(Log(v'_i)\right) = flatten\left(S(diag(log(\sigma_1), \cdots, log(\sigma_{d_u})))S^T\right)$$

where $S$ is the eigenvector-matrix of $v'_i$, and $\sigma_1, \cdots, \sigma_{d_u}$ are the eigenvalues of $v'_i$. The $Log(.)$ operation is the same as equation(1), and the $flatten(A)$ operation flatten the upper triangle of the arbitrary symmetric matrix $A$ .

Finally, we set a fully connected layer and regular softmax operation on embeddings after R2E operation. Suppose the output from the whole model stream is $\hat{y}$, the groundtruth is $y$, we define the loss $\mathcal{L}$ as the cross-entropy loss of $\hat{y}$ and $y$.

## 4 Experiments

Here we evaluate the proposed mAtt using both time-asynchronous and time-synchronous EEG data to give empirical evidence of the advantages. The performance in a general use for EEG decoding is compared against leading DL-based techniques. We incorporate the BCI Competition IV 2a Dataset (BCIC-IV-2a) [44] to assess the performance on time-asynchronous motor-imagery (MI) EEG decoding , the MAMEM EEG SSVEP Dataset II (MAMEM-SSVEP-II) [45] and the BCI challenge error-related negativity (ERN) dataset (BCI-ERN) [46] to assess the performance on time-synchronous SSVEP and ERN EEG decoding. Previous and current state-of-the-art DL-based models listed for comparison with mAtt include MBEEGSE [47], TCNet-Fusion [48], EEG-TCNet [49], FBCNet[50], SCCNet[19], EEGNet[17], and ShallowConvNet[18].

A series of experiments were conducted to evaluate the performances of the mAtt against other EEG decoders with the context of real-world BCI usage taken into account. In the real-world usage of BCI, a user usually needs to go through a training session for collecting a sufficient amount of individual EEG data for training the decoding model before executing the BCI system. To stick with the practical scenario, we performed an individual training scheme where a chunk of trials within a subject are assigned to the training set and the left-over trials within the same subject are used for testing [19, 50]. For the BCIC-IV-2a dataset, we used the first session of a subject to the training set

Table 1: Performance comparison between mAtt and baseline DL methods on MI (BCIC-IV-2a), SSVEP (MAMEM-SSVEP-II), and ERN (BCI-ERN) datasets. Bold fonts mark the highest overall performance (MI, SSVEP: accuracy, ERN: AUC). We adopted Wilcoxon-sign rank test with Bonferroni correction to perform the multiple comparison between all models. The statistical test result is available in the appendix A.9.

| Models | MI | SSVEP | ERN |
|---|---|---|---|
| ShallowConvNet [18] | 61.84±6.39 | 56.93±6.97 | 71.86±2.64 |
| EEGNet [17] | 57.43±6.25 | 53.72±7.23 | 74.28±2.47 |
| SCCNet [19] | 71.95±5.05 | 62.11±7.70 | 70.93±2.31 |
| EEG-TCNet [49] | 67.09±4.66 | 55.45±7.66 | **77.05±2.46** |
| TCNet-Fusion [48] | 56.52±3.07 | 45.00±6.45 | 70.46±2.94 |
| FBCNet [50] | 71.45±4.45 | 53.09±5.67 | 60.47±3.06 |
| MBEEGSE [47] | 64.58±6.07 | 56.45±7.27 | 75.46±2.34 |
| mAtt | **74.71±5.01** | **65.50±8.20** | 76.01±2.28 |

where one out of eight was used for validation for mAtt with $m = 3$. The model with the lowest validation loss within 350 iterations was used for testing on the second session of the same subject. For the MAMEM-SSVEP-II/BCI-ERN dataset, we assigned the first four sessions of a subject to the training set where one out of four was used for validation for mAtt with $m = 7/m = 3$. The model with the lowest validation loss within 180/130 iterations was used for testing on the fifth session of the same subject. The classification performances for BCIC-IV-2a and MAMEM-SSVEP-II datasets were estimated by the mean accuracy across ten repeats for each subject. On the other hand, we use the same criterion as [17], herein the AUC score [51] is adopted to estimate the performance of BCI-ERN dataset due to the imbalanced issue.

### 4.1 Performance comparison

We validate the performance of mAtt against other leading methods. The criteria of selecting the baseline model collection here is based on: 1) code availability and completeness and 2) solid evaluation (e.g. cross-session) without additional auxiliary procedures (e.g. manual feature extraction, data augmentation, pre-trained model, etc). As shown in Table 1, mAtt outperforms all other leading DL methods on both time-synchronous (SSVEP) and -asynchronous (MI) EEG decoding. However, for ERN dataset, mAtt won the second place, only 1% lower than the EEG-TCNet but at least 7% higher than EEG-TCNet on other two kinds of EEG decoding tasks. Nowadays a variety of DL methods employed in EEG classification focus on a specific type of EEG decoding task due to high variability between different types of EEG data [50, 49, 19]. It is difficult to design a DL architecture for decoding various type of EEG data. The table attests the robustness of proposed mAtt, which has strong generalization capacity to adapt general types of EEG data compared to other leading DL models. There are about 3% leap forward the best baseline models on both MI and SSVEP task. The rigorous cross-session training scheme and the preprocessing step that will highly decrease the time resolution of each EEG data we adopted to validate the performance for each model and dataset maybe a reason why some of baseline models may not decode EEG dynamics successfully. Moreover, the rarity of the EEG data may cause arduous overfitting issue for baseline models. We conclude that mAtt has a generalizable superiority in decoding EEG for various types of BCI systems.

### 4.2 Ablation study

We assess the significance of each of the major component in mAtt via a series of ablation analysis. As shown in Table 2, the accuracy reduces if we only use one component in our mAtt to do the classification. However, the combination of our proposed manifold attention module and feature extractor (FE+MA) achieves the best accuracy on all datasets. This implies that there is no redundant component in our proposed mAtt, and each part is needed for its non-negligible functions. The feature extractor aims to denoise and preprocess the EEG signals, and the attention module focus on integrating the preprocessed EEG signals and capturing underlying dynamics in the latent features.

Table 2: Overall accuracy (%) on BCIC-IV-2a and MAMEM-SSVEP-II, and overall auc score on BCI-challenge-ERN (%) with parts within the model appended. FE: feature extractor; MA: manifold attention module; SA: self-attention module.

| Parts appended | BCIC-IV-2a | MAMEM-SSVEP-II | BCI-challenge |
|---|---|---|---|
| FE | 26.08±0.70 | 20.18±1.11 | 73.40±2.27 |
| MA | 60.73±5.80 | 30.51±2.57 | 59.47±3.56 |
| FE+SA | 49.19±2.72 | 22.91±2.00 | 63.77±1.71 |
| FE+MA (proposed) | **74.71**±5.01 | **65.50**±8.20 | **75.71**±2.31 |

We can further compare the result between FE+SA and FE+MA to check the necessity of MA: the performance of FE+MA significantly outperform the regular self-attention based FE+SA on all validation EEG datasets.

### 4.3 Model interpretation

Through analysis for the interpretation of the proposed model, mAtt, we are able to uncover the underlying characteristics learnt from the data. Figure 3 illustrates the gradient response for MI EEG decoding across channel and across time. We can see left/right hand MI responses are strong at C4 and C3 corresponding to right/left motor cortices that control the lateral motor functions of the contralateral side of the body [54]. Both feet and tongue MI, that are not lateral movements, presents strong responses at CPz above the midline of motor cortex. The spatial distribution clearly exhibit

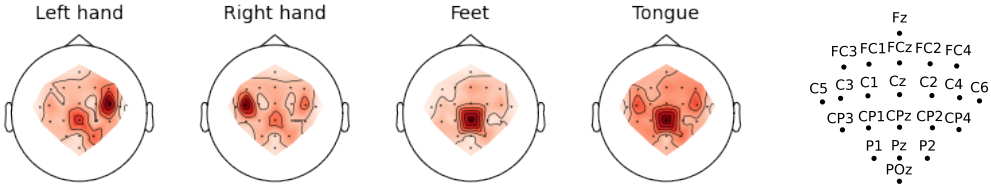



Figure 3: Spatial topomaps for the mean absolute gradient response (computed as in [52, 53]) across time from the visualization of the model S3 in the BCIC-IV-2a dataset for the four motor-imagery classes (left hand, right hand, feet, and tongue). Dark red marks the brains region presenting strong gradient activation at C4 (over right motor cortex) for the left hand, C3 (over left motor cortex) for the right hand, CPz (over motor cortex) for the feet and the tongue motor imagery.

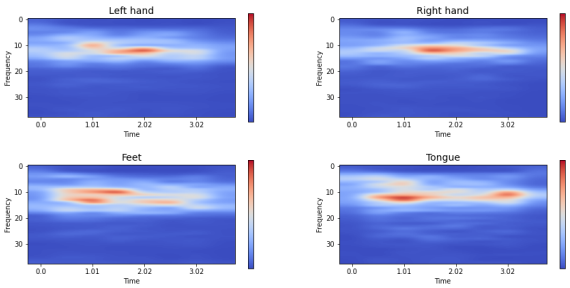
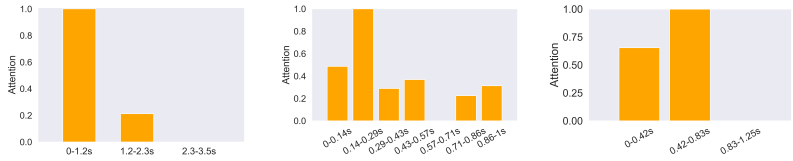



Figure 4: Time-frequency spectrograms from the gradient-based visualization (computed as in [52, 53]) of the model S3 in the BCIC-IV-2a dataset for the four motor-imagery classes (left hand, right hand, feet, and tongue). Strong response of motor imagery is marked by dark red at specific frequency bands and time intervals. Increased response of motor imagery is found at mu band ( 10 Hz) for all classes. The strong response of left/right hand motor imagery occurs at 1-2 seconds, the feet motor imagery is most vivid at 0.5-1 seconds, and the tongue motor imagery induced two peaks at 1 second and 3 seconds.




(a) (b) (c)

Figure 5: (a) The distribution of attention scores across three epochs within a trial gained from the model interpretation of the S3 in the BCIC-IV-2a dataset. (b) The distribution of attention scores across seven epochs within a trial gained from the model interpretation of the S11 in the MAMEM-SSVEP-II dataset. (c) The distribution of attention scores across three epochs within a trial gained from the model interpretation of the S7 in the BCI challenge dataset.

asymmetric pattern for left/right hand MI and symmetric pattern for feet/tongue MI. The temporal information is available in Figure 4, where all four types of MI induce strong response at mu band around 10 Hz. This result is in line with the well-known association between motor function and mu rhythm in EEG recordings [54]. The visualization of our model for SSVEP decoding is exhibit in the appendix. Figure 5 depicts the distribution of attention scores across epochs between the (a) MI, (b) SSVEP, and (c) ERN EEG signals. Here, the attention score refers to the average of the relevance score of an attention network, as described in [42], and has been applied to interpreting an attention-based EEG decoder [43]. For MI EEG signals, attention score is the highest at the first epoch and decreases in the following epochs, which implies that the beginning of the motor imagery may contribute a higher importance determined by the manifold attention module. The profile of attention score for SSVEP EEG signals presents a similar traits that the earlier epochs relate to higher importance. For ERN EEG signals, attention score is the highest on the first two epochs. As we observe a consistency cross EEG datasets that higher attention scores lie in earlier epochs, this may infers that the attention module relies largely on the similarity to the early stage of a trial, which is analogous to baseline correction, a major common procedure in conventional EEG signal processing [55]. This analysis reveals the capability of mAtt in handling the non-stationarity of the dynamical brain activity.

### 4.4 Limitation

In our framework, vacuum permittivity $\epsilon$ is added on all main diagonal elements of covariance $cx_ix_i^T$ to ensure the rigor of SPD matrix. But the operation may cause the repeated singular value $\epsilon$ in $S_i$. Therefore, we proposed possible solutions for this issue: 1) Let $m < n$ when dividing the embeddings into several time segments, reducing the possibility of getting low-rank $S_i$; 2) Let $\epsilon$ be randomly drawn from a specific distribution, such as $Uniform\ (1e-8, 1e-4)$ to solve this issue, which is also a practicable solution; 3) Use the derivative of a low-rank matrix[56] to cope with this issue.

## Conclusion

We propose a manifold attention network as a novel GDL framework for decoding both time-synchronous and -asynchronous EEG decoding. Using back propagation based on the Stiefel manifold, the proposed mAtt is capable of mapping EEG features onto a Riemannian manifold, where spatiotemporal EEG patterns are captured and characterized, within a lightweight architecture. The experimental results suggest the superiority of mAtt over current leading DL methods for both time-synchronous and -asynchronous EEG decoding. With the interpretability of mAtt, we visualize the spatial and temporal EEG patterns, which are in line with prior neuroscientific knowledge and shed light on potential possibility of tracking the brain dynamics. In sum, our privileged method, mAtt, improves the SOTA performance of EEG decoding, and is expected to impact on GDL-based EEG processing with generalizable efficiency and robustness for future development of various BCI systems.

# References


[1] Michal Teplan et al. Fundamentals of eeg measurement. *Measurement science review*, 2(2):1–11, 2002.

[2] Inaki Iturrate, Javier Antelis, and Javier Minguez. Synchronous eeg brain-actuated wheelchair with automated navigation. In *2009 IEEE International Conference on Robotics and Automation*, pages 2318–2325. IEEE, 2009.

[3] D Puthankattil Subha, Paul K Joseph, Rajendra Acharya U, Choo Min Lim, et al. Eeg signal analysis: a survey. *Journal of medical systems*, 34(2):195–212, 2010.

[4] ZT Al-Qaysi, BB Zaidan, AA Zaidan, and MS Suzani. A review of disability eeg based wheelchair control system: Coherent taxonomy, open challenges and recommendations. *Computer methods and programs in biomedicine*, 164:221–237, 2018.

[5] Yu-Te Wang, Masaki Nakanishi, Yijun Wang, Chun-Shu Wei, Chung-Kuan Cheng, and Tzyy-Ping Jung. An online brain-computer interface based on ssveps measured from non-hair-bearing areas. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(1):14–21, 2016.

[6] Xiaogang Chen, Yijun Wang, Masaki Nakanishi, Xiaorong Gao, Tzyy-Ping Jung, and Shangkai Gao. High-speed spelling with a noninvasive brain–computer interface. *Proceedings of the national academy of sciences*, 112(44):E6058–E6067, 2015.

[7] Kai Keng Ang, Karen Sui Geok Chua, Kok Soon Phua, Chuanchu Wang, Zheng Yang Chin, Christopher Wee Keong Kuah, Wilson Low, and Cuntai Guan. A randomized controlled trial of eeg-based motor imagery brain-computer interface robotic rehabilitation for stroke. *Clinical EEG and neuroscience*, 46(4):310–320, 2015.

[8] Luz Maria Alonso-Valerdi, Ricardo Antonio Salido-Ruiz, and Ricardo A Ramirez-Mendoza. Motor imagery based brain–computer interfaces: An emerging technology to rehabilitate motor deficits. *Neuropsychologia*, 79:354–363, 2015.

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

[10] Matteo Feurra, Patrizio Pasqualetti, Giovanni Bianco, Emiliano Santarnecchi, Alessandro Rossi, and Simone Rossi. State-dependent effects of transcranial oscillatory currents on the motor system: what you think matters. *Journal of Neuroscience*, 33(44):17483–17489, 2013.

[11] Don H Johnson. Signal-to-noise ratio. *Scholarpedia*, 1(12):2088, 2006.

[12] Gabriel Emile Hine, Emanuele Maiorana, and Patrizio Campisi. Resting-state eeg: A study on its non-stationarity for biometric applications. In *2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–5. IEEE, 2017.

[13] 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.

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

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

[18] 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.

[19] Chun-Shu Wei, Toshiaki Koike-Akino, and Ye Wang. Spatial component-wise convolutional network (sccnet) for motor-imagery eeg classification. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 328–331. IEEE, 2019.

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

[21] Roberto Torretti. *Philosophy of geometry from Riemann to Poincaré*, volume 7. Springer Science & Business Media, 2012.

[22] 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.

[23] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Riemannian geometry applied to bci classification. In *International conference on latent variable analysis and signal separation*, pages 629–636. Springer, 2010.

[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.

[25] Zhiwu Huang and Luc Van Gool. A riemannian network for spd matrix learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[26] Rudrasis Chakraborty, Jose Bouza, Jonathan Manton, and Baba C Vemuri. Manifoldnet: A deep neural network for manifold-valued data with applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[27] Jose J Bouza, Chun-Hao Yang, David Vaillancourt, and Baba C Vemuri. Mvc-net: A convolutional neural network architecture for manifold-valued images with applications. *arXiv preprint arXiv:2003.01234*, 2020.

[28] Daniel Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, and Matthieu Cord. Riemannian batch normalization for spd neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[29] Minhyung Cho and Jaehyung Lee. Riemannian approach to batch normalization. *Advances in Neural Information Processing Systems*, 30, 2017.

[30] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.

[31] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017.

[32] Guangyi Zhang and Ali Etemad. Rfnet: Riemannian fusion network for eeg-based brain-computer interfaces. *arXiv preprint arXiv:2008.08633*, 2020.

[33] Yoon-Je Suh and Byung Hyung Kim. Riemannian embedding banks for common spatial patterns with eeg-based spd neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 854–862, 2021.

[34] 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.

[35] Ce Ju and Cuntai Guan. Deep optimal transport on spd manifolds for domain adaptation. *arXiv preprint arXiv:2201.05745*, 2022.

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

[37] Wilhelm PA Klingenberg. *Riemannian geometry*, volume 1. Walter de Gruyter, 2011.

[38] 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.

[39] Frédéric Barbaresco. Innovative tools for radar signal processing based on cartan's geometry of spd matrices & information geometry. In *2008 IEEE Radar Conference*, pages 1–6. IEEE, 2008.

[40] 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.

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

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[43] Huy Phan, Kaare B Mikkelsen, Oliver Chen, Philipp Koch, Alfred Mertins, and Maarten De Vos. Sleeptransformer: Automatic sleep staging with interpretability and uncertainty quantification. *IEEE Transactions on Biomedical Engineering*, 2022.

[44] Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, and Gert Pfurtscheller. Bci competition 2008–graz data set a. *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, 16:1–6, 2008.

[45] Spiros Nikolopoulos. MAMEM EEG SSVEP Dataset II (256 channels, 11 subjects, 5 frequencies presented simultaneously). 5 2021.

[46] Perrin Margaux, Maby Emmanuel, Daligault Sébastien, Bertrand Olivier, and Mattout Jérémie. Objective and subjective evaluation of online error correction during p300-based spelling. *Advances in Human-Computer Interaction*, 2012, 2012.

[47] Ghadir Ali Altuwaijri, Ghulam Muhammad, Hamdi Altaheri, and Mansour Alsulaiman. A multi-branch convolutional neural network with squeeze-and-excitation attention blocks for eeg-based motor imagery signals classification. *Diagnostics*, 12(4):995, 2022.

[48] Yazeed K Musallam, Nasser I AlFassam, Ghulam Muhammad, Syed Umar Amin, Mansour Alsulaiman, Wadood Abdul, Hamdi Altaheri, Mohamed A Bencherif, and Mohammed Algabri. Electroencephalography-based motor imagery classification using temporal convolutional network fusion. *Biomedical Signal Processing and Control*, 69:102826, 2021.

[49] Thorir Mar Ingolfsson, Michael Hersche, Xiaying Wang, Nobuaki Kobayashi, Lukas Cavigelli, and Luca Benini. Eeg-tcnet: An accurate temporal convolutional network for embedded motor-imagery brain–machine interfaces. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2958–2965. IEEE, 2020.

[50] 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.

[51] Saharon Rosset. Model selection via the auc. In *Proceedings of the twenty-first international conference on Machine learning*, page 89, 2004.

[52] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[53] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[54] Gert Pfurtscheller, Clemens Brunner, Alois Schlögl, and FH Lopes Da Silva. Mu rhythm (de) synchronization and eeg single-trial classification of different motor imagery tasks. *NeuroImage*, 31(1):153–159, 2006.

[55] Burkhard Maess, Erich Schröger, and Andreas Widmann. High-pass filters and baseline correction in m/eeg analysis. commentary on:"how inappropriate high-pass filters can produce artefacts and incorrect conclusions in erp studies of language and cognition". *Journal of neuroscience methods*, 266:164–165, 2016.

[56] James Townsend. Differentiating the singular value decomposition. Technical report, Technical Report 2016, https://j-towns. github. io/papers/svd-derivative . . . , 2016.

## Checklist

Since we provided a wrong checklist for reviewers, we offer a new checklist here for reviewers to check.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] **We elaborated the contribution and scope of present study in Section1.**
   (b) Did you describe the limitations of your work? [Yes] **We discussed not only limitation, but also the possible solutions for our limitation in Section4.4.**
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] **See Section3 and appendix.**
   (b) Did you include complete proofs of all theoretical results? [Yes] **See Section3 and appendix.**

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] **We provide the instructions and code in the supplemental material for everyone can reproduce our result easily.**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] **We describe the training scheme in Experiments, and the detailed parameters are available in the source codes.**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] **We repeat ten times for any experiment with random initialization and report the standard error for each cell.**
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] **We elaborate on this in the appendix.**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] **We adequately cite any existing assets used in this work.**
   (b) Did you mention the license of the assets? [Yes] **It is provided in the supplemental materials.**
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]