Fast Explanation of RBF-Kernel SVM Models Using Activation Patterns

Mengqi Zhang
School of Computer Science and Infomatics
Cardiff University
Cardiff, UK
Zhangm55@cardiff.ac.uk

Matthias Treder
School of Computer Science and Infomatics
Cardiff University
Cardiff, UK
matthias.treder@gmail.com

David Marshall
School of Computer Science and Infomatics
Cardiff University
Cardiff, UK
marshallad@cardiff.ac.uk

Yuhua Li
School of Computer Science and Infomatics
Cardiff University
Cardiff, UK
liy180@cardiff.ac.uk

Abstract—Machine learning models have significantly enriched the toolbox in the field of neuroimaging analysis. Among them, Support Vector Machines (SVM) have been one of the most popular models for supervised learning, but their use primarily relies on linear SVM models due to their explainability. Kernel SVM models are capable classifiers but more opaque. Recent advances in eXplainable AI (XAI) have developed several feature importance methods to address the explainability problem. However, noise variables can affect these explanations, making irrelevant variables regarded as important variables. This problem also appears in explaining linear models, which the linear pattern can address. This paper proposes a fast method to explain RBF kernel SVM globally by adopting the notion of linear pattern in kernel space. Our method can generate global explanations with low computational cost and is less affected by noise variables. We successfully evaluate our method on simulated and real MEG/EEG datasets.

Index Terms—support vector machine, RBF kernel, activation pattern, neuroimaging, EEG

I. INTRODUCTION

In neuroimaging data analysis, the goal extends beyond merely achieving high performance in machine learning models. The primary objective of neuroscience lies in unravelling the mechanisms of information transmission and processing within the brain. To this end, it is essential not only to analyse but also to interpret and visualize the data utilised by models. Facilitating the explainability of a model's predictions is vital in meeting this requirement.

The Support Vector Machine (SVM) is a widely used model in neuroimaging data analysis due to its simplicity, ability to handle outliers, and exceptional performance. Although linear SVM models are most commonly deployed, kernel-based SVM models can be utilised in nonlinear scenarios and extract more extensive information from the data. Compared with linear SVM models, with sufficient data, these models perform better in some neuroimaging analysis tasks [1]. However, the use of these models is limited, one reason being their explainability issues. Due to the recent development of the

XAI technique, a typical way to understand these models is using feature importance score [2]. It should be noted that both linear and nonlinear model explanations may be influenced by class-irrelevant variables, such as suppressor variables [3], which can inflate the importance scores of features that are, in actuality, irrelevant to the classification task at hand [4].

In order to address this issue, a method that relies on covariance patterns [5] is proposed. By assuming the observations can be generated from a generative model, the classifier weights can be transformed into an activation pattern that reflects the class-related features. However, this method is only applicable to linear models.

This notion can also be adopted when using RBF-SVM models. These models operate on the assumption that kernel functions are capable of transforming data into a higher-dimensional space, where it becomes possible to separate them linearly. Our research demonstrates that it's feasible to construct activation patterns within this kernel space. Moreover, we can assess the significance of features by projecting these high-dimensional patterns back into the input space. Notably, this method of measuring feature importance is less susceptible to the influence of noise variables, such as suppressor variables.

This paper proposes a novel explanation method for RBF-kernel SVM focusing on neuroimaging data analysis. The contributions are summarised as follows:

1) We propose a novel activation pattern based method for explaining RBF-kernel SVM in neuroimaging analysis tasks. We called this method the Estimated Activation Pattern (EAP).
2) This method generates a global feature importance explanation and is more robust to noise variables than comparable approaches.
3) We show the empirical times of different methods. This method requires less computational cost.

This paper proceeds as follows. Section 2 introduces background information and related works. Section 3 presents our proposed method. Section 4 details the experiment evaluation and results. Section 5 concludes the paper with a discussion of further work.

II. BACKGROUND AND RELATED WORKS

Neuroimaging analysis usually serves two purposes: to successfully decode the signal, like brain-computer interface, or to infer knowledge of brain function by explaining the trained models. Mass univariate analysis [6] is a common method that can help gain knowledge from neuroimaging data. These methods concentrate on single variables. Multivariate pattern analysis (MVPA) [7], based on machine learning techniques, has become a popular alternative because of its ability to obtain multivariate patterns that consider multiple variables simultaneously.

To gain insight from trained models, one way is directly using the model performance as information, such as searchlight [8], quantifying how much information a group of channels contains through model performance changes. Another way involves extracting knowledge from model structures, as demonstrated by the study on linear patterns [5]. The evolution of eXplainable Artificial Intelligence (XAI) has significantly enhanced the toolkit for comprehending learned information, particularly in dealing with complex models [9]–[12]. As one of the most popular methods in neuroimaging analysis, efforts have been made to explain the non-linear SVM model. In [13] have shown using rules as explanations. The problem with this method is that explanations may be hard to understand as the number of rules increases. Recently, several feature importance methods have been introduced in this field [14], [15]. However, some studies [16] report noise variables would also affect these methods even in linear models [5]. Some model-agnostic techniques require sampling or permutation processes that may be affected by feature interactions and correlations. It is crucial to consider these factors to ensure the reliability of such methods, as ignoring them may lead to inaccurate results [17].

III. METHODS

A. Linear activation pattern

Explanation using model weights is of limited value because class-uncorrelated variables can have large weights. For example, large weights may be assigned to class-uncorrelated variables for better accuracy, like to fulfil the model structure need. In [5], they addressed this problem and proposed a method to solve it in neuroimaging analysis tasks. In their work, the n d-dimension observations $\mathbf{X} \in \mathcal{R}^{n \times d}$ are assumed to be generated by k latent factors $\mathbf{L} \in \mathcal{R}^{n \times k}$ using specific patterns $\mathbf{W}_{generative} \in \mathcal{R}^{d \times k}$, which is: $\mathbf{X} = \mathbf{L}\,\mathbf{W}_{generative}^T + \epsilon$, where the ϵ represents the noise variables. The latent variables could be a certain brain process or the different classes interested in the analysis task. It can be seen as an encoding process using pattern $\mathbf{W}_{generative}$. While the classier is a decoding process which is $\mathbf{L} = \mathbf{X}\,\mathbf{W}_{linear}$, where the $\mathbf{W}_{linear} \in \mathcal{R}^{d \times k}$ is the weights of the linear model.

By assuming the latent variables are independent, they conclude that with any classifier $\mathbf{L} = \mathbf{X}\,\mathbf{W}_{linear}$, the corresponding pattern can be constructed as $\mathbf{W}_{generative} = \Sigma_{\mathbf{X}}\mathbf{W}_{linear}\Sigma_L^{-1}$, where $\Sigma_{\mathbf{X}}$ represent the covariance matrix of observations $\mathbf{X}$ and Σ_L is the covariance matrix of latent factors $\mathbf{L}$. Specifically for binary classification cases, i.e. $\mathbf{k} = 1$, since there is only one non-zero constant element in Σ_L, the above pattern can be simplified as $\mathbf{W}_{generative} \propto \Sigma_{\mathbf{X}}\mathbf{W}_{linear}$.

B. Construct activation pattern in kernel space

We can extend this idea and construct the pattern in kernel space. The SVM model, both linear and non-linear, aims to obtain a hyperplane between two classes. For the kernel-based SVM model, this hyperplane is represented by part of the data after mapped (assuming the unknown dimension is $\mathbf{m}$), the support vectors $\mathbf{S} \in \mathcal{R}^{s \times m}$ with associated coefficients $\alpha \in \mathcal{R}^s$, where the number of support vectors is $\mathbf{s}$. This model assumes that the mapping function $\phi(.)$ can map non-linearly separable data from input space into high-dimensional space where data can be separated with a hyperplane. This high-dimensional hyperplane can be represented as: $\mathbf{W}_{kernel} = \Sigma_i^s \alpha_i \phi(\mathbf{x}_i)$, where $\phi(\mathbf{x_i}) \in \mathcal{R}^m$ represent the vector in kernel space. To make it clear for the calculation shown below, we rewrite this into matrix format:

$$\mathbf{W}_{kernel} = \mathbf{S}^T \alpha \qquad (1)$$

The mapped data covariance can also be represented in kernel space. There is no guarantee that the mapped data have zero means. For calculation convenience, we introduce the centering matrix $\mathbf{H} = \mathbf{I_n} - \frac{1}{\mathbf{n}}\mathbf{1_n}$, where $\mathbf{H} \in \mathcal{R}^{n \times n}$ and the $\mathbf{I_n} \in \mathcal{R}^{n \times n}$ is $\mathbf{n}$ dimensional identity matrix and $\mathbf{1_n}$ is the n-by-n matrix of all 1. $\mathbf{FH}$ can be seen as the sample minus mean step while calculating variance/covariance. The centring matrix has the useful property that $\mathbf{HH}^T = \mathbf{H}$, which can reduce calculation steps. The mapped covariance matrix is as follows:

$$\begin{aligned} \Sigma_{\phi(\mathbf{x})} &= \frac{1}{n}\mathbf{F}^T\mathbf{H}(\mathbf{F}^T\mathbf{H})^T \\ &= \frac{1}{n}\mathbf{F}^T\mathbf{H}\mathbf{F} \end{aligned} \qquad (2)$$

where $\mathbf{F} \in \mathcal{R}^{n \times m}$ and $\mathbf{F} = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), ..., \phi(\mathbf{x}_n)]$ represent the mapped data matrix.

Combining (1) and (2), the pattern $\mathbf{W}_{\phi-generative} \in \mathcal{R}^m$ can be constructed in kernel space as:

$$\begin{aligned} \mathbf{W}_{\phi-generative} &= \frac{1}{n}\Sigma_{\phi(\mathbf{x})}\mathbf{W}_{kernel} \\ &= \frac{1}{n}\mathbf{F}^T\mathbf{H}\mathbf{F}\mathbf{S}^T\alpha \end{aligned} \qquad (3)$$

We should note that the current pattern $\mathbf{W}_{\phi-generative}$ is in high-dimensional space, which we do not know specifically.

However, this is still a combination of mapped data $\mathbf{F}$. We can easily calculate the combination coefficient $\mathbf{P} \in \mathcal{R}^m$ as follows:

$$\text{Kernel Pattern Coefficient}: \quad \mathbf{P} = \frac{1}{n}\mathbf{HFS}^T\alpha \quad (4)$$

C. Mapping method: fixed point iteration

After constructing the pattern in kernel space, the next step is to map this result back to the input space. To address this problem, we can apply pre-image techniques [18], [19]. Those methods have previously been used for kernel PCA denoising [20], [21].

The idea is to search the associated result $\mathbf{x}^* \in \mathcal{R}^d$ in input space by minimizing the mean squared error (MSE) between $\mathbf{x}^*$ and the target $\mathbf{W}_{\phi-generative}$ in kernel space. The mean squared distance is as follows:

$$
\begin{aligned}
\text{MSE}(\mathbf{x}^*) &= ||\mathbf{W}_{\phi-generative} - \phi(\mathbf{x}^*)||^2 \\
&= \mathbf{P}^T\mathbf{F}^T\mathbf{FP} - \phi(\mathbf{x}^*)^T\mathbf{FP} \\
&\quad - \mathbf{P}^T\mathbf{F}^T\phi(\mathbf{x}^*) + \phi(\mathbf{x}^*)^T\phi(\mathbf{x}^*)
\end{aligned}
\quad (5)
$$

where $\phi(\mathbf{x}^*) \in \mathcal{R}^m$ represents the mapped result $\mathbf{x}^*$ in kernel space. It is obvious that $\mathbf{P}^T\mathbf{F}^T\mathbf{FP}$ is a fixed constant. And $\phi(\mathbf{x}^*)^T\mathbf{FP} = \mathbf{P}^T\mathbf{F}^T\phi(\mathbf{x}^*) = \Sigma_i^n \mathbf{p}_i k(\mathbf{x}^*, \mathbf{x}_i)$, where $k(\mathbf{x}^*, \mathbf{x}_i)$ is the kernel function. After introducing the RBF kernel $k(\mathbf{x}_1, \mathbf{x}_2) = \exp\frac{||\mathbf{x}_1 - \mathbf{x}_2||^2}{\gamma}$, the minimizing function can be simplified as: $\arg\min_{\mathbf{x}^*} \text{MSE}(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - 2\Sigma_i^n \mathbf{p}_i k(\mathbf{x}^*, \mathbf{x}_i) = 1 - 2\Sigma_i^n \mathbf{p}_i \exp\frac{||\mathbf{x}_i - \mathbf{x}^*||^2}{\gamma}$

Here, we use the fixed-point iteration method to search the results. The fixed-point iteration method limited the search space, which decreased the computation cost. Furthermore, the result of this method will have the same scale as the input vectors have [18], which makes the results more straightforward when using the pattern to explain the model. By setting the derivative for $\mathbf{x}^*$ to zero, the fixed-point iteration format is shown below:

$$\mathbf{x}_{t+1}^* = \frac{\Sigma_i^n \mathbf{p}_i k(\mathbf{x}_i, \mathbf{x}_t^*)\mathbf{x}_i}{\Sigma_i^n \mathbf{p}_i k(\mathbf{x}_i, \mathbf{x}_t^*)} \quad (6)$$

The fixed-point iteration method needs an initial vector for iteration. In our setting, all data should be scaled before training classifiers. So, we initialize the initial vector by sampling from a normal distribution with 0 mean and standard deviation of 10, i.e., using the same mean and larger variance than the scaled data. Some studies report that the fixed-point iteration method may suffer from instability and local minima problems [22]. Instead of directly using the final results, we log three solutions with the most minor loss score during the iteration to improve the stability. Then, calculate the normalization of the absolute values of these three solutions. The final solution will take the mean of these normalized results. Running multiple times with different initial vectors is recommended.
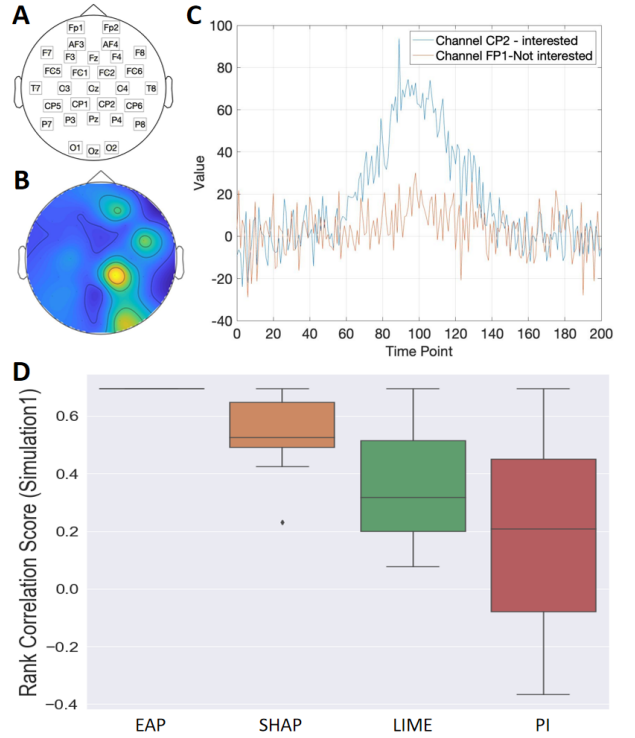

Fig. 1. Example of EEG simulation dataset. Panel A shows the channel position of the simulation. Panel B shows the signal pattern of an example simulation dataset in experiment 1, and Panel C shows 2 example channels in this dataset. The light-coloured area represents the class-related channel. These class-related channels are randomly selected. Panel D shows the rank correlation score between absolute explanation results and absolute generation weights.

IV. EXPERIMENTS AND RESULTS

This section conducts a series of experiments to evaluate our proposed method. The first two experiments use simulated electroencephalography (EEG) datasets. Experiment 3 uses a real visual task EEG and MEG dataset to evaluate how our proposed method performs in real-setting practice. The classifier implemented here is the RBF kernel SVM model. Three existing model explanation methods are implemented as comparisons.

A. Datasets

1) Simulation dataset: Experiments 1 and 2 use ten simulated datasets each, all generated using the MVPA-Light MAT-LAB toolbox [23]. These simulated EEG signals are structured as epoched Event-Related Potential (ERP) settings, uniformly comprising 1000 samples categorized into two classes. Each sample is characterized by 30 channels and 200 time points, with a singular simulated peak signal.

The generation process involves creating a base signal through a normal probability density function, subsequently normalized with fixed mean and standard deviation across all samples. To introduce variability akin to real-world scenarios, each base signal is modulated by a random number drawn from the standard normal distribution. The signal matrix

of each sample is generated by multiplying this modulated base signal with a pattern vector, representing signal strength across channels/features. Moreover, smoothed Gaussian noise is added to each channel. **signal**, **distractor** and **noise** are generated using the same strategy shown above while the pattern vectors differ. The pattern vectors for **signal** and **distractor** are pre-defined and for **noise** are all zeros, i.e., pure noise. Fig. 1-A shows the channel topography and data samples.

a) Experiment 1: These datasets consist of two parts: **signal** and **noise**. For **signal**, the pattern vector is zero except for 6 randomly selected channels which assigned positive weights. The signal pattern can be seen as the ground truth. To introduce nonlinearity, this signal pattern undergoes multiplication by varying coefficients, producing two classes: positive ($\{1, -1\}$) and negative ($\{0.5, -0.5\}$). The composition of the simulation data is $\mathbf{X} = 0.25 \times \textbf{signal} + 0.75 \times \textbf{noise}$.

b) Experiment 2: These datasets consist of three parts: **signal**, **distractor** and **noise**. The pattern vectors for **signal**, **distractor** are pre-defined which are shown in Fig. 2-A. Similar to Experiment 1, the signal pattern undergoes multiplication by distinct coefficients for positive and negative classes, while the distractor pattern is modulated by random numbers drawn from the standard normal distribution to mimic signal changes. The composition of the simulation data is $\mathbf{X} = 0.25 \times \textbf{signal} + 0.25 \times \textbf{distractor} + 0.5 \times \textbf{noise}$.

2) Experiment 3: Real dataset: We used a visual task neuroimaging dataset [24]. The EEG and MEG signals are measured using Elekta Neuromag Vectorview 306 system. All sixteen participants were asked to see pictures of faces and scrambled faces. For each participant, around 290 samples are logged for each class.

a) Preprossessing: Those irrelevant channels, such as ECG and EOG, are first removed. Then apply bandpass filter between 1Hz to 40 Hz with windowed sinc Finite Impulse Response (FIR) filters. For EEG data, signals are re-referenced using the average reference method. After referencing, the data is downsampled to 220HZ to save computation costs. Then trials/samples are segmented based on the event file provided by the dataset. This step can ensure the event-related signal will appear at a similar time point for all trials. Each trial contains 0.5s before seeing the picture, and 1s after. Baseline correlation is applied based on a time window from 0.5s to 0s before the stimulus, which can highlight the signal differences. Finally, 70 EEG and 102 MEG channels (magnetometer) are selected for classification. All preprocessing tasks are carried out using Fieldtrip and MVPA-light toolbox on MATLAB.

Unlike the simulations in which the number and time of peak are fixed, the interested signal of real MEG/EEG data should be picked out first. Two interest time intervals are detected based on local minima in the Global Field Power [25]. The two time intervals are 70 − 145 ms and 145 − 220 ms.

Algorithm 1: Algorithm for EAP

Input:

N:	Number of iterations
α, γ:	Model coefficients of the trained SVM
$[\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x_s}]$:	Support vectors of the trained SVM
$[\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x_n}]$:	Training data
cth:	The minimum changes

Output:

$\mathbf{x}^*$:	Explanation vector

```
// Estimate pattern in kernel space
```
1 Calculate centre matrix $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n$;
2 **for** *i = 1:n* **do**
3   **for** *j = 1:s* **do**
4   Calculate the element $\mathbf{FS}^T = \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$ at row **i** and column **j** using the kernel function with γ used by the model.;
5   **j++** ;
6   **end**
7   **i++** ;
8 **end**
9 Calculate the coefficient of estimated pattern $\mathbf{P} = \frac{1}{n}\mathbf{HFS}^T\alpha$. The estimated pattern in kernel space is $\sum_{i=1}^{n} \mathbf{p}_i\phi(x_i)$.

```
// Mapping the estimated pattern into
   kernel space
```
10 $\mathbf{t} \leftarrow 0$;
11 $\mathbf{x}_0^* \leftarrow$ initialised from standard normal distribution;
12 $\textbf{diff} \leftarrow 1$;
13 **while** *(t ≤ N) AND (|diff| < cth)* **do**
14   *numerator* $\leftarrow 0$;
15   *denominator* $\leftarrow 0$;
16   **for** *i = 1:n* **do**
17   numerator = numerator + $\mathbf{p}_i\mathbf{k}(\mathbf{x}_i, \mathbf{x}_{t-1}^*)\mathbf{x}_i$;
18   denominator = denominator + $\mathbf{p}_i\mathbf{k}(\mathbf{x}_i, \mathbf{x}_{t-1}^*)$;
19       $i++$;
20   **end**
21    $\mathbf{x}_t^* = \frac{numerator}{denominator}$;
22    $\mathbf{L}_t = \text{MSE}(\mathbf{x}_t^*)$;
23   **if** *t > 1* **then**
24       $\textbf{diff} = \frac{\text{MSE}(\mathbf{x}_{t-1}^*) - \text{MSE}(\mathbf{x}_t^*)}{\text{MSE}(\mathbf{x}_{t-1}^*)}$
25   **end**
26    $t++$;
```
     // Log 3 different solutions with
        the smallest loss
```
27    $\mathbf{x}_{l1}^*, \mathbf{x}_{l2}^*, \mathbf{x}_{l3}^* \leftarrow$ log three solutions with the most minor loss score during the iteration
28 **end**
29 $\mathbf{x}_{l1}^*, \mathbf{x}_{l2}^*, \mathbf{x}_{l3}^* \leftarrow$ calculate the absolute value and normalised ;
30 $\mathbf{x}^* = (\mathbf{x}_{l1}^* + \mathbf{x}_{l2}^* + \mathbf{x}_{l3}^*)/3$

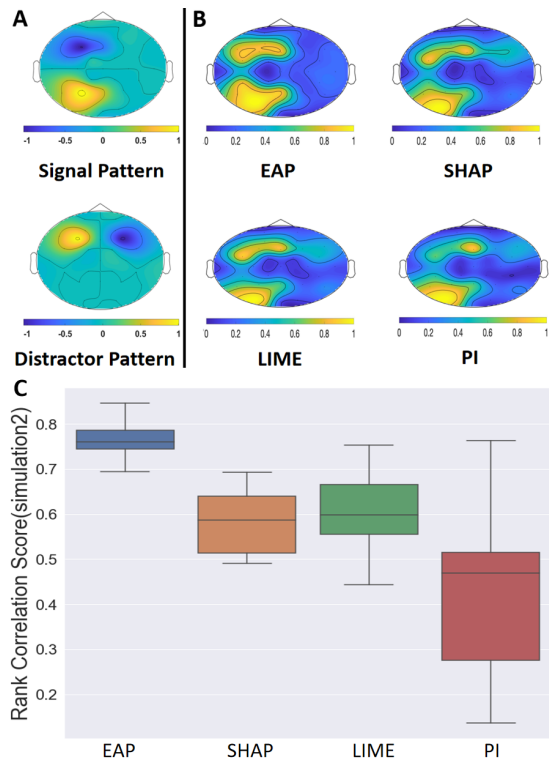|	PI	LIME	SHAP	EAP
Experiment 1	4.03	554.44	18414.00	2.24
Experiment 2	4.43	573.18	18730.36	2.45
EEG 1st interval	10.93	752.26	11763.12	2.30
EEG 2nd interval	9.18	8984.92	9060.15	1.88
MEG 1st interval	9.55	717.92	10401.64	1.83
MEG 2nd interval	13.15	12531.63	9828.30	1.53
Iris-versicolor-vs-virginica	0.02	3.33	42.95	0.07

Fig. 2. The patterns for the simulation dataset in experiment 2 and results. The upper plot of panel A shows the simulation generation weights of the signal pattern. The lower plot of panel A shows the simulation generation weights of the class-irrelevant distractor pattern. These distracting signals are irrelevant to the different classes but may affect the explanation results. Panel B shows the explanation results of different methods. Panel C shows the rank correlation score between the absolute weights of the signal pattern and absolute explanation results.

B. Classifier

In all three experiments, we use the statistical feature of the signal [26], i.e., the mean channel value among selected time intervals, as the classification feature. All classifiers are built in Python using Sci-kit learn toolbox. Hyperparameters are tunning based on 5-fold cross-validation. Before building models, data are first scaled using the standard scaler as default.

C. Comparison methods

Several state-of-the-art explanation methods are selected for comparison, including permutation importance (PI) [27], [28], local interpretable model-agnostic explanation (LIME) [29], and Shapley additive explanations (SHAP) [30].

These methods will generate results in different scales. The explanations may not have the same scale. For the convenience of comparison, all results use their absolute values and then normalized between 0 to 1.

D. Results

1) Experment 1: Fig. 1 shows the results of experiment 1. As mentioned in the previous section, the randomly assigned signal pattern can be seen as the ground truth of experiment 1. The rank correlation score is calculated to show how the

explanation results performed. As shown in Fig. 1, all methods can obtain good results in experiment 1. The correlation score of EAP outperformed other methods and showed a low variance compared to other methods. The permutation importance shows the worst results.

2) Experment 2: The results shown in Fig. 2-B are the averaged results of simulation datasets. All simulation datasets have the same ground truth. Under the view of averaged results, all methods have roughly picked up the true area shown in the signal pattern. While the EAP result shows less affected by the distractor signal, the other three methods are, to some extent, affected by the distractor factors. Fig. 2 Panel C shows the rank correlation score between the absolute signal pattern and the explanation results. Like the results shown in experiment 1, the EAP methods outperformed the other three methods with lower variance. The results have indicated that the EAP method can generate robust explanation results under the effect of distractor factors.

3) Experment3: Fig. 3 shows the results of experiment 3. All explanation results are taken in absolute value, rescaled to 0 to 1, and averaged over the 16 participants. Unlike simulation datasets, we do not have the exact ground truth of real-world MEG/EEG data. But we can compare our results with previous studies.

The selected time intervals are consistent with the two components of the visual tasks brain cognitive process, which are P100 and N170. These two components have been reported in many previous studies [31], [32].

As shown in the figure, all methods in EEG experiments in the first time interval (70-145ms) highlight the bottom back area, which reflects the occipital area. PI, LIME, and SHAP also highlight the two side areas of both hemispheres, which reflect the temporal area of the brain. In the second time interval (145-220ms), the EAP method mainly focuses on channels located at the occipital and occipital-temporal areas on the right hemisphere. While the other 3 methods focus mainly on channels located in the middle occipital area. The first time interval reflects the P100 component in visual-related studies. Channels located in the middle occipital area are reported in previous studies [33], [34]. These channels show signal differences related to face vs. non-face stimulus around 100ms after the stimulus. The second time interval reflects the N170 component. Signal differences are reported at channels located at the occipital-temporal area on both hemispheres [33], [34]. In contrast, channels located in the right occipital-

	Sepal length	Sepal width	Petal length	Petal width
PI	0.0376	0	0.9037	1
LIME	0.0785	0	0.6554	1
SHAP	0.0666	0	0.6189	1
EAP	**0.5028**	0	1	0.9775
P-value(T-test)	1.72E-07	0.0018	3.18E-22	2.23E-26

A 70 - 145ms   145 - 220ms
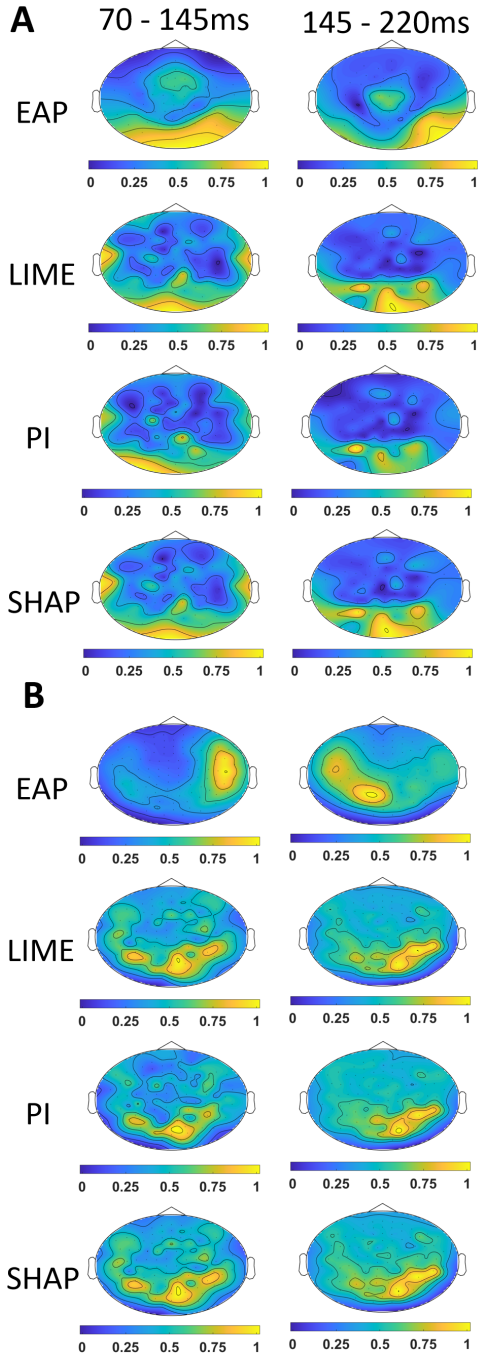
EAP

LIME

PI

SHAP

B

EAP

LIME

PI

SHAP

Fig. 3. The results of experiment 3. Figure A shows the explanation results of EEG data. Figure B shows the explanation results of MEG data.

temporal area detect significantly stronger signals than the left counterpart [35], [36]. In summary, compared with all methods, the EAP method better reflects the previous study.

For results of MEG data, in the first time interval, the EAP method highlights the channels located in the anterior area of the right hemisphere, which is the temporal area. While other methods highlight channels located in occipital and occipital-temporal areas. In the second time interval, the EAP result highlights the left occipital-temporal area. The other 3 methods highlight the middle occipital area and right occipital-temporal areas. In previous studies, in the early stage (around 100 ms after stimulus), some report almost no channel level difference between face vs. non-face stimulus [37]. In contrast, channel level differences are found at channels located at the right temporal area, left frontal-temporal area, and right occipital-temporal area at around 150 ms [38]. For the second time interval, channel level differences are reported at the occipital-temporal area of both hemispheres [37], [39]. However, channel level differences are reported at the left occipital-temporal area and the right temporal area at around 200 ms. In summary, all methods partly reflect the previous studies.

4) Empirical computation cost: This subsection provides an empirical comparison of computational cost between the proposed method and other benchmark methods. The experiment is carried out on a desktop with an i7 9700k CPU with 32 GB RAM. The operating system is Ubuntu 20.04. Table I summarizes the run time, which shows our EAP method is multiple order faster than LIME and SHAP and a few times faster than PI.

5) Additional experiment on non-neuroimaging data: The proposed method primarily focuses on analyzing neuroimaging data. However, we are also interested in exploring the applicability of our methodology to handle general tabular data. To this end, we evaluate the proposed method using the iris dataset with the Versicolor class and the Virginia class. Fig. 4 shows the feature distribution between iris-versicolor and iris-virginica. We also perform t-test between the two classes for each feature as listed in table II. Based on visual observation of Fig. 4 and the t-test results in table II, petal length and petal width are the most useful features and sepal width is the least useful, while sepal length is medially useful. The experimental results in table II indicate that all methods correctly highly rate the two most important features of petal length and petal width and dismiss the least important feature of sepal width. However, the EAP method can better identify

the usefulness of sepal length while other explanation methods underestimated the importance score of this feature as shown in table II.
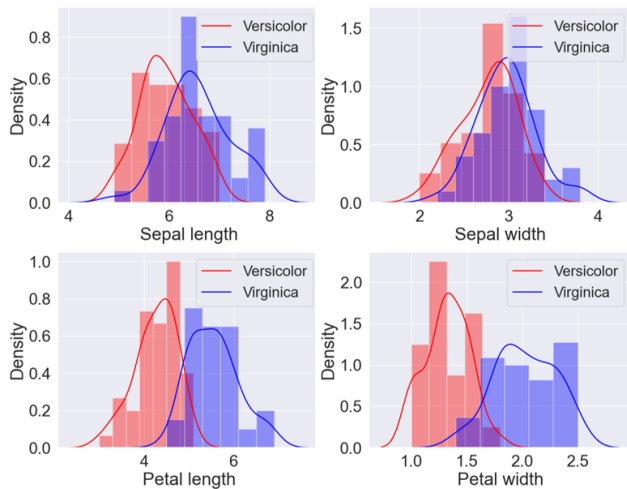

Fig. 4. The feature distribution of the features in the iris dataset between iris-versicolor and iris-virginica. From the plot, the Sepal length can provide some information to distinguish the two classes, while the sepal width cannot.

V. CONCLUSION AND FURTHER WORK

In summary, this paper proposes a novel explanation method EAP for the RBF kernel SVM model. We implement our method and three other state-of-art explanation methods to the EEG simulation dataset and real MEG/EEG dataset of visual tasks. The experimental results show that the EAP can obtain the global explanation results with relatively low computation costs. Furthermore, the EAP method can generate robust explanation results and is less affected by distractor factors, both in MEG/EEG datasets and non-neuroimaging data.

This method may also have potential for other kinds of kernel types. One of our future directions is applying our method to other kinds of kernel methods with different kernel types. Besides, one limitation of our method is that the EAP method currently can only generate global explanations. Most feature importance based methods operate under the assumption of feature independence. This assumption can skew results due to the influence of noise variables that are dependent on other informative features. A fruitful direction would be to extend our method to generate local explanations.

REFERENCES

[1] R. N. Sekkal, F. Bereksi-Reguig, D. Ruiz-Fernandez, N. Dib, and S. Sekkal, "Automatic sleep stage classification: From classical machine learning methods to deep learning," *Biomedical Signal Processing and Control*, vol. 77, p. 103751, 2022.

[2] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019.

[3] D. J. Krus and S. M. Wilkinson, "Demonstration of properties of a suppressor variable," *Behavior Research Methods, Instruments, & Computers*, vol. 18, pp. 21–24, 1986.

[4] R. Wilming, C. Budding, K.-R. Müller, and S. Haufe, "Scrutinizing xai using linear ground-truth data with suppressor variables," *Machine learning*, vol. 111, no. 5, pp. 1903–1923, 2022.

[5] S. Haufe, F. Meinecke, K. Görgen, S. Dähne, J.-D. Haynes, B. Blankertz, and F. Bießmann, "On the interpretation of weight vectors of linear models in multivariate neuroimaging," *NeuroImage*, vol. 87, pp. 96–110, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811913010914

[6] D. M. Groppe, T. P. Urbach, and M. Kutas, "Mass univariate analysis of event-related brain potentials/fields i: A critical tutorial review," *Psychophysiology*, vol. 48, no. 12, pp. 1711–1725, 2011.

[7] J. V. Haxby, "Multivariate pattern analysis of fmri: the early beginnings," *Neuroimage*, vol. 62, no. 2, pp. 852–855, 2012.

[8] N. Kriegeskorte, R. Goebel, and P. Bandettini, "Information-based functional brain mapping," *Proceedings of the National Academy of Sciences*, vol. 103, no. 10, pp. 3863–3868, 2006.

[9] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, and D. Shen, "Review of artificial intelligence techniques in imaging data acquisition, segmentation, and diagnosis for covid-19," *IEEE reviews in biomedical engineering*, vol. 14, pp. 4–15, 2020.

[10] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[11] B. Kim, R. Khanna, and O. O. Koyejo, "Examples are not enough, learn to criticize! criticism for interpretability," *Advances in neural information processing systems*, vol. 29, 2016.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[13] N. Barakat and A. P. Bradley, "Rule extraction from support vector machines: a review," *Neurocomputing*, vol. 74, no. 1-3, pp. 178–190, 2010.

[14] S. Valentin, M. Harkotte, and T. Popov, "Interpreting neural decoding models using grouped model reliance," *PLoS Computational Biology*, vol. 16, no. 1, p. e1007148, 2020.

[15] J. D. Bennett, S. E. John, D. B. Grayden, and A. N. Burkitt, "Universal neurophysiological interpretation of eeg brain-computer interfaces," in *2021 9th International Winter Conference on Brain-Computer Interface (BCI)*. IEEE, 2021, pp. 1–6.

[16] C. Budding, F. Eitel, K. Ritter, and S. Haufe, "Evaluating saliency methods on artificial data with different background types," *arXiv preprint arXiv:2112.04882*, 2021.

[17] C. Molnar, G. König, J. Herbinger, T. Freiesleben, S. Dandl, C. A. Scholbeck, G. Casalicchio, M. Grosse-Wentrup, and B. Bischl, "General pitfalls of model-agnostic interpretation methods for machine learning models," in *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*. Springer, 2020, pp. 39–68.

[18] P. Honeine and C. Richard, "Preimage problem in kernel-based machine learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 77–88, 2011.

[19] J.-Y. Kwok and I.-H. Tsang, "The pre-image problem in kernel methods," *IEEE transactions on neural networks*, vol. 15, no. 6, pp. 1517–1525, 2004.

[20] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel pca and de-noising in feature spaces," *Advances in neural information processing systems*, vol. 11, 1998.

[21] T. Takahashi and T. Kurita, "Robust de-noising by kernel pca," in *Artificial Neural Networks—ICANN 2002: International Conference Madrid, Spain, August 28–30, 2002 Proceedings 12*. Springer, 2002, pp. 739–744.

[22] T. J. Abrahamsen and L. K. Hansen, "Input space regularization stabilizes pre-images for kernel pca de-noising," in *2009 IEEE International Workshop on Machine Learning for Signal Processing*. IEEE, 2009, pp. 1–6.

[23] M. S. Treder, "Mvpa-light: a classification and regression toolbox for multi-dimensional data," *Frontiers in Neuroscience*, vol. 14, p. 289, 2020.

[24] D. G. Wakeman and R. N. Henson, "A multi-subject, multi-modal human neuroimaging dataset," *Scientific data*, vol. 2, no. 1, pp. 1–10, 2015.

[25] W. Skrandies, "Global field power and topographic similarity," *Brain topography*, vol. 3, pp. 137–141, 1990.

[26] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of erp components—a tutorial," *NeuroImage*, vol. 56, no. 2, pp. 814–825, 2011.

[27] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously." *J. Mach. Learn. Res.*, vol. 20, no. 177, pp. 1–81, 2019.

[28] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[29] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[30] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[31] N. Boutros, H. Nasrallah, R. Leighty, M. Torello, P. Tueting, and S. Olson, "Auditory evoked potentials, clinical vs. research applications," *Psychiatry Research*, vol. 69, no. 2-3, pp. 183–195, 1997.

[32] J. Kropotov, "Chapter 3.1-sensory systems and attention modulation," *Functional neuromarkers for psychiatry. Academic Press; Boston, Massachusetts, USA*, pp. 140–142, 2016.

[33] M. Negrini, D. Brkić, S. Pizzamiglio, I. Premoli, and D. Rivolta, "Neurophysiological correlates of featural and spacing processing for face and non-face stimuli," *Frontiers in Psychology*, vol. 8, p. 333, 2017.

[34] B. Rossion and C. Jacques, "Does physical interstimulus variance account for early electrophysiological face sensitive responses in the human brain? ten lessons on the n170," *Neuroimage*, vol. 39, no. 4, pp. 1959–1979, 2008.

[35] Y. Wang, H. Huang, H. Yang, J. Xu, S. Mo, H. Lai, T. Wu, and J. Zhang, "Influence of eeg references on n170 component in human facial recognition," *Frontiers in neuroscience*, vol. 13, p. 705, 2019.

[36] R. N. Henson, E. Mouchlianitis, and K. J. Friston, "Meg and eeg data fusion: simultaneous localisation of face-evoked responses," *Neuroimage*, vol. 47, no. 2, pp. 581–589, 2009.

[37] J. Liu, A. Harris, and N. Kanwisher, "Stages of processing in face perception: an meg study," *Nature neuroscience*, vol. 5, no. 9, pp. 910–916, 2002.

[38] F. Tadel, E. Bock, G. Niso, J. C. Mosher, M. Cousineau, D. Pantazis, R. M. Leahy, and S. Baillet, "Meg/eeg group analysis with brainstorm," *Frontiers in neuroscience*, p. 76, 2019.

[39] Y. Xu, J. Liu, and N. Kanwisher, "The m170 is selective for faces, not for expertise," *Neuropsychologia*, vol. 43, no. 4, pp. 588–597, 2005.