Received: 30 September 2021 | Revised: 4 March 2022 | Accepted: 18 April 2022

DOI: 10.1002/sam.11587

WILEY

RESEARCH ARTICLE

Coupled support tensor machine classification for multimodal neuroimaging data

Peide Li[1] | **Seyyid Emre Sofuoglu**[2] | **Selin Aviyente**[2] | **Tapabrata Maiti**[3]

[1]Boehringer Ingelheim Pharmaceuticals, Duluth, Georgia, USA

[2]College of Engineering, Michigan State University, East Lansing, Michigan, USA

[3]College of Natural Science, Michigan State University, East Lansing, Michigan, USA

Correspondence
Seyyid Emre Sofuoglu, College of Engineering, Michigan State University, East Lansing, MI, USA.
Email: sofuoglu@msu.edu

Funding information
Division of Mathematical Sciences, Grant/Award Number: 1924724

Abstract

Multimodal data arise in various applications where information about the same phenomenon is acquired from multiple sensors and across different imaging modalities. Learning from multimodal data is of great interest in machine learning and statistics research as this offers the possibility of capturing complementary information among modalities. Multimodal modeling helps to explain the interdependence between heterogeneous data sources, discovers new insights that may not be available from a single modality, and improves decision-making. Recently, coupled matrix–tensor factorization has been introduced for multimodal data fusion to jointly estimate latent factors and identify complex interdependence among the latent factors. However, most of the prior work on coupled matrix–tensor factors focuses on unsupervised learning and there is little work on supervised learning using the jointly estimated latent factors. This paper considers the multimodal tensor data classification problem. A coupled support tensor machine (C-STM) built upon the latent factors jointly estimated from the advanced coupled matrix–tensor factorization is proposed. C-STM combines individual and shared latent factors with multiple kernels and estimates a maximal-margin classifier for coupled matrix–tensor data. The classification risk of C-STM is shown to converge to the optimal Bayes risk, making it a statistically consistent rule. C-STM is validated through simulation studies as well as a simultaneous analysis on electroencephalography with functional magnetic resonance imaging data. The empirical evidence shows that C-STM can utilize information from multiple sources and provide a better classification performance than traditional single-mode classifiers.

KEYWORDS

classification, coupled tensor decomposition, multimodal data, support tensor machine

1 INTRODUCTION

Advances in clinical neuroimaging and computational bioinformatics have dramatically increased our understanding of various brain functions using multiple modalities such as magnetic resonance imaging (MRI), functional MRI (fMRI), electroencephalography (EEG), and positron emission tomography (PET). Their strong connections to the patients' biological status and disease pathology suggest the great potential of their predictive power in disease diagnostics. Numerous studies using vector- and tensor-based statistical models illustrate how to utilize these imaging data both at the voxel- and region-of-interest (ROI) level and develop efficient biomarkers that predict disease status. For example, Anderson et al. [7] propose a classification model using functional connectivity MRI for autism disease and reach 89% diagnostic accuracy for subjects under 20. Schindlbeck and Eidelberg [64] utilize network models and brain imaging data to develop novel biomarkers for Parkinson's disease. Many works in Alzheimer's disease research such as [21, 27, 37, 49, 52, 54, 55] use EEG, MRI, and PET imaging data to predict patient's cognition and detect early-stage Alzheimer's diseases. Although these studies have provided impressive results, utilizing imaging data from a single modality such as individual MRI sequences are known to have limited predictive capacity, especially in the early phases of the disease. For instance, Li et al. [49] use brain MRI volumes from ROIs to identify patients in early-stage Alzheimer's disease. They use 1-year MRI data from Alzheimer's disease neuroimaging initiative (ADNI) and obtain 77% prediction accuracy. Although such a performance is favorable compared to other existing approaches, the diagnostic accuracy is relatively low due to the limited information from MRI data. In recent years, it has been common to acquire multiple neuroimaging modalities in clinical studies such as simultaneous EEG-fMRI or MRI and fMRI. Even though each modality measures different biological signals, they are interdependent and mutually informative. Learning from multimodal neuroimaging data may help integrate information from multiple sources and facilitate biomarker developments in clinical studies. It also raises the need for novel supervised learning techniques for multimodal data in statistical learning literature.

The existing statistical approaches to multimodal data science are dominated by unsupervised learning methods. These methods analyze multimodal neuroimaging data by performing joint matrix decomposition and extracting common information across different modalities. During optimization, the decomposed factors bridging two or more modalities are estimated to interpret the connections between different modalities. Examples of these methods include matrix-based joint independent component analysis [6, 16, 30, 44, 51, 67], which assume bilinear correlations between factors in different modalities. However, these matrix–vector-based models cannot preserve the multilinear nature of original data and the spatiotemporal correlations across modes as most neuroimaging modalities are naturally in tensor format. Recently, various coupled matrix–tensor decomposition methods have been introduced to address this issue [4–6, 17, 18, 36, 56]. These methods impose different soft or hard multilinear constraints between factors from different modalities providing more flexibility in data modeling.

Current supervised learning approaches for multimodal data mostly concatenate data modalities as extra features without exploring their interdependence. For example, Li et al. [48] and Zhou et al. [77] build generalized regression models by appending tensor and vector predictors linearly for image prediction and classification. Pan et al. [60] develop a discriminant analysis by including tensor and vector predictors in a linear fashion. Li and Li [46] propose an integrative factor regression for multimodal neuroimaging data assuming that data from different modalities can be decomposed into latent factors. More recently, Gahrooei et al. [26] proposed multiple tensor-on-tensor regression for multimodal data, which combines tensor-on-tensor regression from [53] with traditional additive linear model. Another type of integration utilizes kernel tricks and combines information from multimodal data with multiple kernels. Gönen and Alpaydın [29] provide a survey on various multiple kernel learning (MKL) techniques for multimodal data fusion and classification with support vector machines (SVMs). Combining kernels linearly or nonlinearly instead of original data in different modalities provides more flexibility in information integration. Bach [10] proposed a multiple kernel regression model with group lasso penalty, which integrates information by multiple kernels and selects the most predictive data modalities.

Despite these accomplishments, the current approaches have several shortcomings. First, they mainly focus on exploring the interdependence between multimodal imaging data, ignoring the representative and discriminative power of the learned components. Thus, the methods cannot further bridge the imaging data to the patients' biological status, which is not helpful in biomarker development. Second, the supervised techniques such as integrate information primarily by data or feature concatenation without explicitly considering the possible correlations between different modalities. This lack of consideration of interdependence may cause issues like overfitting and parameter identifiability. Third, even though methods from [26, 46] have considered latent structures for multimodal data, these models are designed primarily for linear regression and are not

directly applicable to classification problems. Fourth, the aforementioned multimodal analysis methods are mainly vector based methods, which cannot handle large-size multidimensional data encountered in contemporary data science. As discussed in [14], tensors provide a powerful tool for analyzing multidimensional data in statistics. As a result, developing a novel multimodal tensor-based statistical framework for supervised learning can be of great interest. Finally, although many empirical studies demonstrate the success of using multimodal data, there is a lack of mathematical and statistical clarity to the extent of generalizability and associated uncertainties. The absence of a solid statistical framework for multimodal data analysis makes it impossible to interpret the generalization ability of a certain statistical model.

In this paper, we propose a two-stage coupled support tensor machine (C-STM) for multimodal tensor-based neuroimaging data classification. The proposed model addresses the current issues in multimodal data science and provides a sound statistical framework to interpret the interdependence between modalities and quantify the model consistency and generalization ability. The *major contributions* of this work are as follows:

1. Individual and common latent factors are extracted from multimodal tensor data, for each sample or subject, using advanced coupled matrix–tensor factorization (ACMTF) [3, 5]. The extracted components are then utilized in a statistical framework. Most of the works on ACMTF do not work on each subject separately and the extracted factors are utilized for a signal analysis rather than a subsequent statistical learning framework. Specifically, the work on supervised approaches with CMTF is limited.
2. Building a novel C-STM with both the coupled and noncoupled tensor CP factors for classification. In this regard, MKL approaches are adopted to integrate components from multimodal data.
3. For the validation of our work, we provide both theoretical and empirical evidence. We provide theoretical results such as classification consistency for statistical guarantee. A thorough numerical study has been conducted, including a simulation study and experiments on real data to illustrate the usefulness of the proposed methodology.

A Matlab package is also provided in the supplemental material, including all functions for C-STM classification. The source codes are available at our Github repository.[1]

[1]https://github.com/PeterLiPeide/Coupled_MatrixTensor_SupportTensor_Machine

2 | RELATED WORK

In this section, we review some background and prior work on tensor decompositions and MKL.

2.1 | Notations

In this work, we denote numbers and scalars by letters such as x, y, N. Vectors are denoted by boldface lowercase letters, for example, $\mathbf{x}, \mathbf{y}$. Matrices are denoted by boldface capital letters like $\mathbf{X}, \mathbf{Y}$. Multidimensional tensors are denoted by boldface Euler script letters such as $\mathcal{X}, \mathcal{Y}$. The order of a tensor is the number of dimensions of the data hypercube, also known as ways or modes. For example, a scalar can be regarded as a zeroth-order tensor, a vector is a first-order tensor, and a matrix is a second-order tensor.

Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be a tensor of order N, where $x_{i_1, i_2, \ldots, i_N}$ denotes the $(i_1, i_2, \ldots, i_N)$ th element of the tensor. Vectors obtained by fixing all indices of the tensor except the one that corresponds to nth mode are called mode-n fibers and denoted as $\mathbf{x}_{i_1, \ldots i_{n-1}, i_{n+1}, \ldots, i_N} \in \mathbb{R}^{I_n}$. The mode-$n$ unfolding of $\mathcal{X}$ is defined as $\mathcal{X}_{(n)} \in \mathbb{R}^{I_n \times \prod_{n'=1, n' \neq n}^{N} I_{n'}}$, where the mode-$n$ fibers of the tensor $\mathcal{X}$ are the columns of $\mathcal{X}_{(n)}$ and the remaining modes are organized accordingly along the rows.

2.2 | Canonical/polyadic decomposition

Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}$ be a tensor with d modes. Rank-r canonical/polyadic decomposition of $\mathcal{X}$ is defined as:

$$\mathcal{X} \approx \sum_{k=1}^{r} \mathbf{x}_k^{(1)} \circ \mathbf{x}_k^{(2)} \cdots \circ \mathbf{x}_k^{(d)} = [\![\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(d)}]\!], \quad (1)$$

where $\mathbf{X}^{(j)} \in \mathbb{R}^{I_j \times r}, j = 1, .., d$ are defined as factor matrices whose columns are $\mathbf{x}_r^{(j)}$ and "$\circ$" represents the vector outer product. The second equality in (1) is called Kruskal tensor [40], which is a convenient representation for CP tensors. We denote a Kruskal tensor by $\mathfrak{U}_x = \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(d)}$ or $\mathfrak{U}_x = \zeta; \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(d)}$, where $\zeta \in \mathbb{R}^r$ is a vector whose entries are the weights of rank one tensor components. In the special case of matrices, ζ corresponds to singular values of a matrix. In general, it is assumed that the rank r is small so that Equation (1) is also called low-rank approximation for a tensor $\mathcal{X}$. Such an approximation can be estimated by an alternating least square approach [39].

Although there are other tensor decomposition structures such as Tucker decomposition or tensor train decomposition, the advantage of CP is that the factors extracted by a CP decomposition are unique up to permutation. The

uniqueness of the factors makes CP decomposition more interpretable.

2.3 | Coupled matrix–tensor factorization

Motivated by the fact that joint analysis of data from multiple sources can potentially unveil complex data structures and provide more information, CMTF [2] was proposed for multimodal data fusion. CMTF estimates the underlying latent factors for both tensor and matrix data simultaneously by taking the coupling between tensor and matrix data into account. This feature makes CMTF a promising model in analyzing heterogeneous data, which generally have different structures and modalities.

During latent factor estimation, CMTF solves an objective function that approximates a CP decomposition for the tensor modality and a singular value decomposition for the second modality with the assumption that the factors from one mode of each modality are the same. Given $\mathcal{X}_1 \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_d}$ and $\mathbf{X}_2 \in \mathbb{R}^{I_1 \times J_2}$, without loss of generality assume that the factors from the first mode of the tensor $\mathcal{X}_1$ span the column space of the matrix $\mathbf{X}_2$. CMTF then tries to estimate all factors by minimizing:

$$\mathbf{Q}(\mathfrak{U}_1, \mathbf{V}) = \frac{1}{2}\left\| \mathcal{X}_1 - \left[\!\left[\mathbf{X}_1^{(1)}, \mathbf{X}_1^{(2)}, \ldots, \mathbf{X}_1^{(d)} \right]\!\right] \right\|_{\text{Fro}}^2 + \frac{1}{2} \| \mathbf{X}_2 - \mathbf{X}_2^{(1)} \mathbf{X}_2^{(2)T} \|_{\text{Fro}}^2, \quad \text{s.t. } \mathbf{X}_1^{(1)} = \mathbf{X}_2^{(1)}, \tag{2}$$

where $\mathbf{X}_p^{(m)}$ are the factor matrices for modality p and mode m. The factor matrices $\mathbf{X}_1^{(1)} = \mathbf{X}_2^{(1)}$ are the coupled factors between tensor and matrix data. An illustration of this coupling is given in Figure 1. These factor matrices can also be represented in Kruskal form, $\mathfrak{U}_1 = \left[\!\left[\mathbf{X}_1^{(1)}, \mathbf{X}_1^{(2)}, \ldots, \mathbf{X}_1^{(d)} \right]\!\right]$ and $\mathfrak{U}_2 = \left[\!\left[\mathbf{X}_2^{(1)}, \mathbf{X}_2^{(2)} \right]\!\right]$. By minimizing the objective function $\mathbf{Q}(\mathfrak{U}_1, \mathfrak{U}_2)$, CMTF estimates latent factors for the tensor and matrix data jointly which allows it to utilize information from both modalities. Acar et al. [2] use a gradient descent algorithm to optimize the objective function (2). Although this model is formulated for the joint decomposition of a dth order tensor and a matrix, extensions to two or more tensors with couplings across multiple modes are possible.

In real data, couplings across different modalities might include shared or modality-specific (individual) components. Shared components correspond to those columns of the factor matrices that contribute to the decomposition of both modalities, while individual components carry information unique to the corresponding modality. Although CMTF provides a successful framework for joint data analysis, it often fails to obtain a unique estimation for shared or individual components. As a result, any further statistical analysis and learning from CMTF estimation will suffer from the uncertainty in latent factors. To address this issue, Acar et al. [3] proposed ACMTF by introducing a sparsity penalty to the weights of latent factors in the objective function (2), and restricting the norm of the columns of the factors to be unity to provide uniqueness up to a permutation. This modification provides a more precise estimation for latent factors compared to CMTF [3, 4]. In our framework, we utilize ACMTF to extract the latent factors which are in turn used to build a classifier for multimodal data.

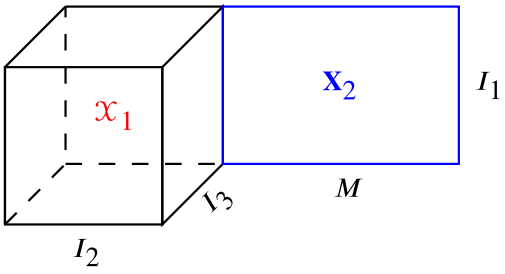


FIGURE 1 Illustration of coupled tensor matrix model

2.4 | CP-STM for tensor classification

CP-STM has been previously studied by He et al. [32, 33] and Tao et al. [69] and uses CP tensor to construct STM types of model. Assume there is a collection of data $T_n = \{(\mathcal{X}_1 y_1), (\mathcal{X}_2 y_2), \ldots, (\mathcal{X}_n y_n)\}$, where $\mathcal{X}_t \in \mathcal{X} \subset \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_d}$ are d-way tensors. $\mathcal{X}$ is a compact tensor space, which is a subspace of $\mathbb{R}^{I_1 \times I_2 \times \ldots \times I_d}$. $y_t \in \{1, -1\}$ are binary labels. CP-STM assumes the tensor predictors are in CP format, and can be classified by the function which minimizes the objective function

$$\min \quad \lambda \|\mathbf{f}\|^2 + \frac{1}{n} \sum_{t=1}^{n} \mathcal{L}(\mathbf{f}(\mathcal{X}_t), y_t). \tag{3}$$

By using tensor kernel function

$$K(\mathcal{X}_1, \mathcal{X}_2) = \sum_{l,m=1}^{r} \prod_{j=1}^{d} K^{(j)}\left(\mathbf{x}_{1,l}^{(j)}, \mathbf{x}_{2,m}^{(j)} \right), \tag{4}$$

where $\mathcal{X}_1 = \sum_{l=1}^{r} \mathbf{x}_{1l}^{(1)} \circ \ldots \circ \mathbf{x}_{1l}^{(d)}$ and $\mathcal{X}_2 = \sum_{l=1}^{r} \mathbf{x}_{2l}^{(1)} \circ \ldots \circ \mathbf{x}_{2l}^{(d)}$. The STM classifier can be written as

$$\mathbf{f}(\mathcal{X}) = \sum_{t=1}^{n} \alpha_t y_t K(\mathcal{X}_t, \mathcal{X}) = \boldsymbol{\alpha}^T \mathbf{D}_y \mathbf{K}(\mathcal{X}) \tag{5}$$

where $\mathcal{X}$ is a new d-way rank-r tensor, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^T$ is the coefficient vector, $\mathbf{D}_y$ is a diagonal matrix whose diagonal elements are $y_1, \ldots, y_n$ and $\mathbf{K}(\mathcal{X}) = [K(\mathcal{X}_1\mathcal{X}), \ldots, K(\mathcal{X}_n\mathcal{X})]^T$ is a column vector, whose values are kernel values computed between training and test data. We denote the collection of functions in the form of (5) with $\mathcal{H}$, which is a functional space also known as reproducing kernel Hilbert space (RKHS). The optimal classifier CP-STM $\mathbf{f} \in \mathcal{H}$ can be estimated by plugging function (5) into objective function (3), and minimize it with hinge or squared hinge loss. The coefficients of the optimal CP-STM model are denoted by $\boldsymbol{\alpha}^*$. The classification model is statistically consistent if the tensor kernel function satisfying the universal approximating property, which is shown by Li and Maiti [45].

2.5 | Multiple kernel learning

MKL creates new kernels using a linear or nonlinear combination of single kernels to measure inner products between data. Statistical learning algorithms such as SVM and kernel regression can then utilize the new combined kernels instead of single kernels to obtain better learning results and avoid the potential bias from kernel selection [29]. A more important and related reason for using MKL is that different kernels can take inputs from various data representations possibly from different sources or modalities. Thus, combining kernels and using MKL is one possible way of integrating multiple information sources.

Given a collection of kernel functions $\{K_1(\cdot,\cdot), \ldots, K_m(\cdot,\cdot)\}$, a new kernel function can be constructed by

$$K(\cdot,\cdot) = \mathbf{f}_{\boldsymbol{\eta}}\left(\{K_1(\cdot,\cdot), \ldots, K_m(\cdot,\cdot)\} \,|\boldsymbol{\eta}\right), \tag{6}$$

where $\mathbf{f}_\eta$ is a linear or nonlinear function and $\boldsymbol{\eta}$ is a vector whose elements are the weights for the kernel combination. Linear combination methods are the most popular in MKL, where the kernel function is parameterized as

$$\begin{aligned} K(\cdot,\cdot) &= \mathbf{f}_{\boldsymbol{\eta}}\left(\{K_1(\cdot,\cdot), \ldots, K_m(\cdot,\cdot)\} \,|\boldsymbol{\eta}\right) \\ &= \sum_{l=1}^{m} \eta_l K_l(\cdot,\cdot). \end{aligned} \tag{7}$$

The weight parameters η_l can be simply assumed to be the same (unweighted) [12, 61], or be determined by looking at some performance measures for each kernel or data representation [63, 68]. There are few more advanced approaches such as optimization-based, Bayesian approaches, and boosting approaches that can also be adopted [13, 19, 25, 28, 38, 43, 71]. In this work, we only consider linear combination (7), and select the weight parameters in a heuristic data-driven way to construct our C-STM model.

3 | METHODOLOGY

Let $T_n = \left\{\left(\mathcal{X}_{1,1}, \mathbf{X}_{1,2}, y_1\right), \ldots, \left(\mathcal{X}_{n,1}, \mathbf{X}_{n,2}, y_n\right)\right\}$ be training data, where each sample $t \in \{1, \ldots, n\}$ has two data modalities $\mathcal{X}_{t,1}, \mathbf{X}_{t,2}$ and a corresponding binary label $y_t \in \{1, -1\}$. In this work, following [2], we assume that the first data modality is a third-order tensor, $\mathcal{X}_{t,1} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, and the other is a matrix, $\mathbf{X}_{t,2} \in \mathbb{R}^{I_4 \times I_3}$. The third mode of $\mathcal{X}_{t,1}$ and the second mode of $\mathbf{X}_{t,2}$ are assumed to be coupled for each t, that is, the factor matrix is assumed to be fully or partially shared across these modes. Utilizing this coupling, one can extract factors that better represent the underlying structure of the data, and preserve and utilize the discriminative power of the factors from both modalities. Our approach, C-STM, consists of two stages: multimodal tensor factorization, that is, ACMTF, and C-STM as illustrated in Figure 2. In this section, we present both stages and the corresponding procedures.

3.1 | Multimodal tensor factorization

In this work, the first aim is to perform a joint factorization across two modalities for each training sample, t. Let $\mathfrak{U}_{t,1} = \left[\!\left[\zeta; \mathbf{X}_{t,1}^{(1)}, \mathbf{X}_{t,1}^{(2)}, \mathbf{X}_{t,1}^{(3)}\right]\!\right]$ denote the Kruskal tensor of $\mathcal{X}_{t,1}$, and $\mathfrak{U}_{t,2} = \left[\!\left[\sigma; \mathbf{X}_{t,2}^{(1)}, \mathbf{X}_{t,2}^{(2)}\right]\!\right]$ denote the singular value decomposition of $\mathbf{X}_{t,2}$. The weights of the columns of each factor matrix $\mathbf{X}_{t,p}^{(m)}$, where p is the index for modality and m denotes the mode, are denoted by ζ and σ and the norms of these columns are constrained to be 1 to avoid redundancy. The objective function of ACMTF [3, 5] is then given by:

$$\begin{aligned} \mathbf{Q}\left(\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2}\right) = \gamma \left\| \mathcal{X}_{t,1} - \left[\!\left[\zeta; \mathbf{X}_{t,1}^{(1)}, \mathbf{X}_{t,1}^{(2)}, \mathbf{X}_{t,1}^{(3)}\right]\!\right] \right\|_{\text{Fro}}^2 \\ + \gamma \parallel \mathbf{X}_{t,2} - \mathbf{X}_{t,2}^{(1)} S \mathbf{X}_{t,2}^{(2)^\top} \parallel_{\text{Fro}}^2 \\ + \beta \parallel \zeta \parallel_0 + \beta \parallel \sigma \parallel_0 \\ \text{s.t.} \quad \mathbf{X}_{t,1}^{(3)} = \mathbf{X}_{t,2}^{(2)} \\ \parallel \mathbf{x}_{t,1,k}^{(1)} \parallel_2 = \parallel \mathbf{x}_{t,1,k}^{(2)} \parallel_2 = \parallel \mathbf{x}_{t,1,k}^{(3)} \parallel_2 \\ = \parallel \mathbf{x}_{t,2,k}^{(1)} \parallel_2 = \parallel \mathbf{x}_{t,2,k}^{(2)} \parallel_2 = 1, \\ \forall k \in \{1, \ldots, r\}, \end{aligned} \tag{8}$$

where Σ is a diagonal matrix whose elements are the singular values σ of the matrix $\mathbf{X}_{t,2}$ and $\mathbf{x}_{t,m,k}^{(j)} \in \mathbb{R}^{I_j}$ denotes the columns of the factor matrices for the object $\mathcal{X}_{t,m}$. The objective function in (8) includes penalties for the number of nonzero weights in both tensor and matrix

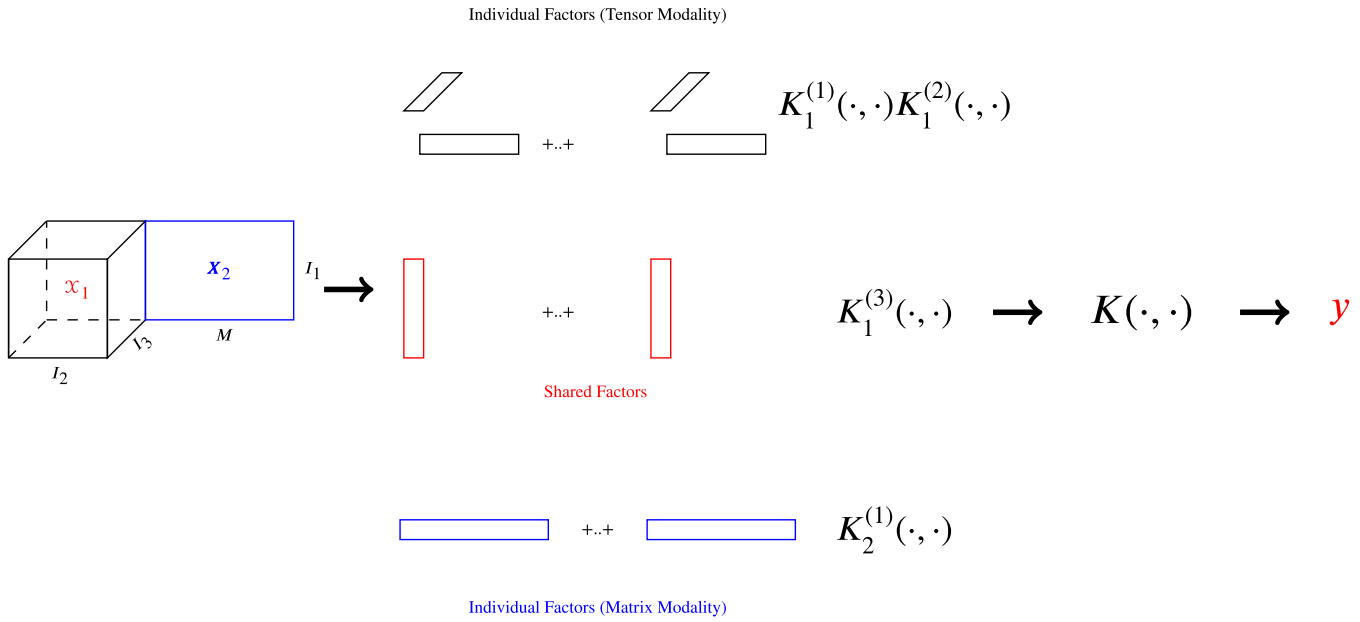


FIGURE 2 Coupled support tensor machine (C-STM) model pipeline

decomposition. Thus, the model identifies the shared and individual components. These factors are then considered as different data representations for multimodal data, and used to predict the labels y_t in C-STM classifier.

3.2 | Coupled support tensor machine

C-STM uses the idea of MKL and considers the coupled and uncoupled factors from ACMTF decomposition as various data representations. As a result, we use three different kernel functions to measure their similarity, that is, inner products. One can think of these three kernels inducing three different feature maps transforming multimodal factors into different feature spaces. In each feature space, the corresponding kernel measures the similarity between factors in this specific data modality. The similarities of multimodal factors are then integrated by combining the kernel measures through a nonlinear combination. This combination should be able to take individual and shared components into account separately for better adaptability depending on the size and corruptions on the data as the coupled modes are likely to be better estimated than the individual modes. Thus, we use tensor kernels for individual modes of each modality and combine these with the kernels of the coupled modes as illustrated in Figure 2. The kernel function for C-STM is defined as

$$\begin{aligned} K\left(\left(\mathcal{X}_{t,1}\mathbf{X}_{t,2}\right),\left(\mathcal{X}_{i,1}\mathbf{X}_{i,2}\right)\right) &= K\left(\left(\mathfrak{U}_{t,1}\mathfrak{U}_{t,2}\right),\left(\mathfrak{U}_{i,1}\mathfrak{U}_{i,2}\right)\right) \\ &= \sum_{k,l=1}^{r} w_1 K_1^{(1)}\left(\mathbf{x}_{t,1,k}^{(1)},\mathbf{x}_{i,1,l}^{(1)}\right) K_1^{(2)}\left(\mathbf{x}_{t,1,k}^{(2)},\mathbf{x}_{i,1,l}^{(2)}\right) \\ &\quad + w_2 K_1^{(3)}\left(\mathbf{x}_{t,1,k}^{(3)*},\mathbf{x}_{i,1,l}^{(3)*}\right) \\ &\quad + w_3 K_2^{(1)}\left(\mathbf{x}_{t,2,k}^{(1)},\mathbf{x}_{i,2,l}^{(1)}\right) \end{aligned} \tag{9}$$

for two pairs of decomposed tensor matrix factors $\left(\mathfrak{U}_{t,1},\mathfrak{U}_{t,2}\right)$ and $\left(\mathfrak{U}_{i,1},\mathfrak{U}_{i,2}\right)$. $\mathbf{x}_{t,1,k}^{(3)*}$ is the average of the estimated shared factors $\frac{1}{2}\left[\mathbf{x}_{t,1,k}^{(3)}+\mathbf{x}_{t,2,k}^{(2)}\right]$. This kernel is inspired by the idea of MKL with linear combination of multiple kernels for multimodal data. Few more details regarding chosing such kernel combination are provided in Section 6.1. w_1, w_2, and w_3 are the three weight parameters combining the three kernel functions. As discussed in [29], there is no unique choice for determining these weights, in this paper, we adopt a cross-validation approach as explained in the Appendix C.2.

With the kernel function in (9), C-STM model tries to estimate a bivariate decision function $\mathbf{f}$ from a collection of functions $\mathcal{H}$ such that

$$\mathbf{f} = \arg\min \quad \lambda \cdot \|\mathbf{f}\|^2 + \frac{1}{n}\sum_{t=1}^{n}\mathcal{L}\left(\mathbf{f}(\mathcal{X}_t), y_t\right), \tag{10}$$

where $\mathcal{L}(\mathcal{X}_t, y_t) = \max(0, 1 - \mathbf{f}(\mathcal{X}_t)\cdot y_t)$ is Hinge loss. $\mathcal{H}$ is defined as the collection of all functions in the form of

$$\begin{aligned} \mathbf{f}(\mathcal{X}_1,\mathbf{X}_2) &= \sum_{t=1}^{n}\alpha_t y_t K\left(\left(\mathcal{X}_{t,1}\mathbf{X}_{t,2}\right),\left(\mathcal{X}_1\mathbf{X}_2\right)\right) \\ &= \alpha^T \mathbf{D}_y \mathbf{K}\left(\mathcal{X}_1,\mathbf{X}_2\right) \end{aligned} \tag{11}$$

due to the well-known representer theorem [8] for any pair of testing data $(\mathcal{X}_1,\mathbf{X}_2)$ and for $\boldsymbol{\alpha}\in\mathbb{R}^n$. For all

possible values of $\boldsymbol{\alpha}$, Equation (11) defines the data collection $\mathcal{H}$. $\mathbf{D}_y$ is a diagonal matrix whose diagonal elements are labels from the training data T_n. $\mathbf{K}(\mathcal{X}_1, \mathbf{X}_2)$ is a $n \times 1$ vector whose t-th element is $K\left(\left(\mathcal{X}_{t,1}\mathbf{X}_{t,2}\right), \left(\mathcal{X}_1\mathbf{X}_2\right)\right)$. The optimal C-STM decision function, denoted by $\mathbf{f_n} = \boldsymbol{\alpha}^{*T}\mathbf{D}_y\mathbf{K}(\mathcal{X}_1, \mathbf{X}_2)$, can be estimated by solving the quadratic programming problem

$$\begin{aligned} &\min_{\alpha \in \mathbb{R}^n} \quad \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{D}_y\mathbf{K}\mathbf{D}_y\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha}, \\ &\text{s.t.} \quad \boldsymbol{\alpha}^T\mathbf{y} = 0, \\ &0 \preccurlyeq \boldsymbol{\alpha} \preccurlyeq \frac{1}{2n\lambda}, \end{aligned} \tag{12}$$

where $\mathbf{K}$ is the kernel matrix constructed by function (9). Problem (12) is the dual problem of (10), and its optimal solution $\boldsymbol{\alpha}^*$ also minimizes the objective function (10) when plugging functions in the form of (11). For a new pair of test points $(\mathcal{X}_1, \mathbf{X}_2)$, the class label is predicted as sign $(\mathbf{f_n}(\mathcal{X}_1, \mathbf{X}_2))$.

4 | MODEL ESTIMATION

In this section, we first present the estimation procedure for coupled matrix–tensor decomposition (8), and then combine it with the classification procedure to summarize the algorithm for C-STM.

To satisfy the constraints in the objective function (8), the function $\mathbf{Q}\left(\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2}\right)$ is converted to a differentiable and unconstrained form given by:

$$\begin{aligned} \mathbf{Q}\left(\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2}\right) = {} & \gamma \left\| \mathcal{X}_{t,1} - \left[\!\left[\zeta; \mathbf{X}_{t,1}^{(1)}, \mathbf{X}_{t,1}^{(2)}, \mathbf{X}_{t,1}^{(3)}\right]\!\right] \right\|_{\text{Fro}}^2 \\ & + \gamma \parallel \mathbf{X}_{t,2} - \mathbf{X}_{t,2}^{(1)} S \mathbf{X}_{t,2}^{(2)^\top} \parallel_{\text{Fro}}^2 \\ & + \xi \parallel \mathbf{X}_{t,1}^{(3)} - \mathbf{X}_{t,2}^{(2)} \parallel_{\text{Fro}}^2 \\ & + \sum_{k=1}^{r} \left[\beta\sqrt{\zeta_r^2 + \varepsilon} + \beta\sqrt{\sigma_r^2 + \varepsilon} \right. \\ & + \theta \left[\left(\parallel \mathbf{x}_{t,1,k}^{(1)} \parallel_2 - 1 \right)^2 \right. \\ & + \left(\parallel \mathbf{x}_{t,1,k}^{(2)} \parallel_2 - 1 \right)^2 \\ & + \left(\parallel \mathbf{x}_{t,1,k}^{(3)} \parallel_2 - 1 \right)^2 + \left(\parallel \mathbf{x}_{t,2,k}^{(1)} \parallel_2 - 1 \right)^2 \\ & \left. \left. + \left(\parallel \mathbf{x}_{t,2,k}^{(2)} \parallel_2 - 1 \right)^2 \right] \right], \end{aligned} \tag{13}$$

where ℓ_1 norm penalties in (8) are replaced with differentiable approximations; ξ and θ are Lagrange multipliers and $\varepsilon > 0$ is a very small number. This unconstrained optimization problem can be solved by nonlinear conjugate gradient descent [2, 5, 56].

Let $\mathcal{T}_t$ be the full (created by converting Kruskal tensor, or the factor matrices into multidimensional array form) tensor of $\mathfrak{U}_{t,1}$, and $\mathbf{M}_t = \mathbf{X}_{t,2}^{(1)} S \mathbf{X}_{t,2}^{(2)^\top}$, the partial derivative of each latent factor can be derived as follows:

$$\begin{aligned} \frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2}\right)}{\delta \mathbf{X}_{t,1}^{(1)}} = {} & \gamma \left(\mathcal{T}_t - \mathcal{X}_{t,1}\right)_{(1)} \left(\zeta^\top \odot \mathbf{X}_{t,1}^{(3)} \odot \mathbf{X}_{t,1}^{(2)}\right) \\ & + \theta \left(\mathbf{X}_{t,1}^{(1)} - \overline{\mathbf{X}}_{t,1}^{(1)}\right), \end{aligned} \tag{14}$$

$$\begin{aligned} \gamma \frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2}\right)}{\delta \mathbf{X}_{t,1}^{(2)}} = {} & \left(\mathcal{T}_t - \mathcal{X}_{t,1}\right)_{(2)} \left(\zeta^\top \odot X_{t,1}^{(3)} \odot \mathbf{X}_{t,1}^{(1)}\right) \\ & + \theta \left(\mathbf{X}_{t,1}^{(2)} - \overline{\mathbf{X}}_{t,1}^{(2)}\right), \end{aligned} \tag{15}$$

$$\begin{aligned} \gamma \frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2}\right)}{\delta \mathbf{X}_{t,1}^{(3)}} = {} & \left(\mathcal{T}_t - \mathcal{X}_{t,1}\right)_{(3)} \left(\zeta^\top \odot X_{t,1}^{(2)} \odot \mathbf{X}_{t,1}^{(1)}\right) \\ & + \xi \left(\mathbf{X}_{t,1}^{(3)} - \mathbf{X}_{t,2}^{(2)}\right) + \theta \left(\mathbf{X}_{t,1}^{(3)} - \overline{\mathbf{X}}_{t,1}^{(3)}\right), \end{aligned} \tag{16}$$

$$\gamma \frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2}\right)}{\delta \mathbf{X}_{t,2}^{(1)}} = \left(\mathbf{M}_t - \mathbf{X}_{t,2}\right) \mathbf{X}_{t,2}^{(2)} \boldsymbol{\Sigma} + \theta \left(\mathbf{X}_{t,2}^{(1)} - \overline{\mathbf{X}}_{t,2}^{(1)}\right), \tag{17}$$

$$\begin{aligned} \gamma \frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2}\right)}{\delta \mathbf{X}_{t,2}^{(2)}} = {} & \left(\mathbf{M}_t - \mathbf{X}_{t,2}\right)^\top \mathbf{X}_{t,2}^{(1)} \boldsymbol{\Sigma} + \\ & \tau \left(\mathbf{X}_{t,2}^{(2)} - \mathbf{X}_{t,1}^{(3)}\right) + \theta \left(\mathbf{X}_{t,2}^{(2)} - \overline{\mathbf{X}}_{t,2}^{(2)}\right), \end{aligned} \tag{18}$$

$$\begin{aligned} \frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2}\right)}{\delta \sigma_k} = {} & \mathbf{x}_{t,2,k}^{(1)\top} \left(\mathbf{M}_t - \mathbf{X}_{t,2}\right) \mathbf{x}_{t,2,k}^{(2)} \\ & + \frac{\beta}{2} \frac{\sigma_k}{\sqrt{\sigma_k^2 + \varepsilon}}, \quad k \in \{1, \ldots, r\}, \end{aligned} \tag{19}$$

$$\begin{aligned} \frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2}\right)}{\delta \zeta_k} = {} & \operatorname{vec}\left(\mathcal{T}_t - \mathcal{X}_{t,1}\right)^\top \\ & \times \left(\mathbf{x}_{t,1,k}^{(3)} \odot \mathbf{x}_{t,1,k}^{(2)} \odot \mathbf{x}_{t,1,k}^{(1)}\right) \\ & + \frac{\beta}{2} \frac{\zeta_k}{\sqrt{\zeta_k^2 + \varepsilon}}, \quad k \in \{1, \ldots, r\}, \end{aligned} \tag{20}$$

where (vec(.)) is a vectorization operator that stacks all elements of the operand in a column vector, $\mathcal{T}_{(j)}$ denotes the mode-j unfolding of a tensor $\mathcal{T}$, and $\odot$ denotes Khatri–Rao product. $\overline{\mathbf{M}}$ is a normalized matrix whose columns have unit ℓ_2 norms.

By combining all of the partial derivatives, the partial derivative of the objective function is given by:

$$\nabla \mathbf{Q}\left(\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2}\right) = \left[\frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}\mathfrak{U}_{t,2}\right)}{\delta \mathbf{X}_{t,1}^{(1)}}, \frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}\mathfrak{U}_{t,2}\right)}{\delta \mathbf{X}_{t,1}^{(2)}}, \right.$$

$$\frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}\mathfrak{U}_{t,2}\right)}{\delta \mathbf{X}_{t,1}^{(3)}}, \frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}\mathfrak{U}_{t,2}\right)}{\delta \mathbf{X}_{t,2}^{(2)}}, \frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}\mathfrak{U}_{t,2}\right)}{\delta \zeta_1}, \cdots, \frac{\delta \mathbf{Q}\left(\mathfrak{U}_{t,1}\mathfrak{U}_{t,2}\right)}{\delta \sigma_1}, \cdots \Bigg]^{\top}$$

which is a $2r+5$ dimensional vector. As mentioned in [5], a nonlinear conjugate gradient method with Hestenes–Stiefel updates is used to optimize (13). The procedure is described in Algorithm 1.

Once the factors for all data pairs in the training set T_n are extracted, we can create the kernel matrix using the kernel function in (9). By solving the quadratic programming problem (12), we can obtain the optimal decision function $\mathbf{f_n}$. This two-stage procedure for C-STM estimation is summarized in Algorithm 2.

5 | THEORY

In this section, we provide some preliminary theoretical results to validate the C-STM model. The first proposition provides a sketch of proof of convergence for the coupled matrix–tensor decomposition.

Proposition 1. Suppose for every pair of multimodal data $\mathcal{X}_{t,1}$ and $\mathbf{X}_{t,2}$, the optimal latent factor estimate is the optimal solution for the objective function (8), which is denoted by $\mathfrak{U}_{t,1}^*, \mathfrak{U}_{t,2}^*$. The conjugate gradient descent Algorithm 1 converges to stable estimates for tensor and matrix components $\mathfrak{U}_{t,1}^*, \mathfrak{U}_{t,2}^*$, where:

$$D\left(\left(\mathfrak{U}_{t,1}^{\tau}\mathfrak{U}_{t,2}^{\tau}\right), \left(\mathfrak{U}_{t,1}^{*}\mathfrak{U}_{t,2}^{*}\right)\right) \to 0, \quad \text{as } \tau \to \infty, \tag{22}$$

Algorithm 1. ACMTF decomposition

Input: Multimodal data $(\mathcal{X}_1, \mathbf{X}_2)$, r, η, S (Upper limit for the number of iterations)
Output: $\mathfrak{U}_{t,1}^*, \mathfrak{U}_{t,2}^*$
$\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2} = \mathfrak{U}_{t,1}^0, \mathfrak{U}_{t,2}^0$ ▷ Initial value
$\mathbf{\Delta}_0 = -\nabla \mathbf{Q}(\mathfrak{U}_{t,1}^0, \mathfrak{U}_{t,2}^0)$
$\varphi_0 = \arg\min_{\varphi} \mathbf{Q}\left[(\mathfrak{U}_{t,1}^0, \mathfrak{U}_{t,2}^0) + \varphi \mathbf{\Delta}_0\right]$
$\mathfrak{U}_{t,1}^1, \mathfrak{U}_{t,2}^1 = (\mathfrak{U}_{t,1}^0, \mathfrak{U}_{t,2}^0) + \varphi_0 \mathbf{\Delta}_0$
$\mathbf{g}_0 = \mathbf{\Delta}_0$
while s < S and $\|\mathbf{Q}(\mathfrak{U}_{t,1}^s, \mathfrak{U}_{t,2}^s) - \mathbf{Q}(\mathfrak{U}_{t,1}^{s-1}, \mathfrak{U}_{t,2}^{s-1})\| \geqslant \eta$ **do**
  $\mathbf{\Delta}_{s+1} = -\nabla \mathbf{Q}(\mathfrak{U}_{t,1}^s, \mathfrak{U}_{t,2}^s)$
  $\mathbf{g}_{s+1} = \mathbf{\Delta}_{s+1} + \frac{\mathbf{\Delta}_{s+1}^{\top}(\mathbf{\Delta}_{s+1} - \mathbf{\Delta}_s)}{-\mathbf{g}_s^{\top}(\mathbf{\Delta}_{s+1} - \mathbf{\Delta}_s)}\mathbf{g}_s$
  $\varphi_{s+1} = \arg\min_{\varphi} \mathbf{Q}\left[(\mathfrak{U}_{t,1}^s, \mathfrak{U}_{t,2}^s) + \varphi \mathbf{g}_{s+1}\right]$
  $\mathfrak{U}_{t,1}^{s+1}, \mathfrak{U}_{t,2}^{s+1} = (\mathfrak{U}_{t,1}^s, \mathfrak{U}_{t,2}^s) + \varphi_{s+1}\mathbf{g}_{s+1}$
end while

Algorithm 2. Coupled support tensor machine

procedure C-STM
  Input: Training set $T_n = \{(\mathcal{X}_{1,1}, \mathbf{X}_{1,2}, y_1), ..., (\mathcal{X}_{n,1}, \mathbf{X}_{n,2}, y_n)\}$, $\mathbf{y}$, kernel function K, r, λ, η, S
  for t = 1, 2,...n **do**
    $\mathfrak{U}_{t,1}^*, \mathfrak{U}_{t,2}^* = \text{ACMTF}\big((\mathcal{X}_{t,1}, \mathbf{X}_{t,2}), r, \eta, \text{S}\big)$
  end for
  Create initial matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$
  for t = 1,...,n **do**
    for i = 1,...,n **do**
      $\mathbf{K}[i,t] = K\big((\mathfrak{U}_{t,1}, \mathfrak{U}_{t,2}), (\mathfrak{U}_{i,1}, \mathfrak{U}_{i,2})\big)$ ▷ Kernel values
      $\mathbf{K}[i,t] = \mathbf{K}[t,i]$
    end for
  end for
  Solve the quadratic programming problem (12) and find the optimal $\boldsymbol{\alpha}^*$.
  Output: $\boldsymbol{\alpha}^*$
end procedure

where τ is the number of iterations and $D(.,.)$ is a distance measure between Kruskal tensor sets such as ℓ_2 distance between factors with an appropriate selection of permutations. This proposition is a direct result of the convergence property of nonlinear conjugate gradient descent algorithm with line search [31, 58, 73, 74, 78]. The convergence rate of nonlinear conjugate gradient descent is linear. For detailed convergence properties of nonlinear conjugate gradient descent with Hestenes–Stiefel updates on nonconvex objectives, the readers are referred to [58, 62].

The next result discusses the statistical property of C-STM. Let us assume the risk of a decision function, $\mathbf{f}$, is $\mathcal{R}(\mathbf{f}) = \mathbb{E}_{\mathcal{X}\times\mathcal{Y}}[\mathbf{1}\{\mathbf{f}(\mathcal{X}) \neq y\}]$, where $\mathcal{X} \subset \mathbb{R}^{I_1 \times .. \times I_d}$ is a subspace of $\mathbb{R}^{I_1 \times .. \times I_d}$. $\mathcal{Y} = \{1, -1\}$. The expectation is taken over the joint distribution defined on $\mathcal{X} \times \mathcal{Y}$, which is a data domain. The function $\mathbf{1}\{\cdot\}$ is an indicator function measuring the loss of classification function $\mathbf{f}$. It is also known as the "zero–one" loss since its value is zero when the decision function provides correct prediction and is one otherwise. If there is a $\mathbf{f}^* : \mathcal{X} \to \mathcal{Y}$ from the collection of all measurable functions such that $\mathbf{f}^* = \arg\min \mathcal{R}(\mathbf{f})$, its risk is called the Bayes risk for the classification problem with data from $\mathcal{X} \times \mathcal{Y}$. We denote the Bayes risk as $\mathcal{R}^* = \mathcal{R}\left(\mathbf{f}^*\right)$. With different training sets T_n, we can estimate a sequence of decision functions $\mathbf{f_n}$ under the same training procedure. This sequence of decision function $\{\mathbf{f_n}\}$ is called a decision rule. Obviously, C-STM is a decision rule if different training sets are provided. A decision rule is statistically consistent if $\mathcal{R}\left(\mathbf{f_n}\right)$ converges to the Bayes

risk $\mathcal{R}^*$ as the size of training data n increases [20]. The consistency property is desirable for classification rules, because a consistent rule guarantees to reconstruct the whole data distribution with more training data/observations. The reconstruction here means the Bayes risk of the classification problem will be eventually the same as the risk of estimated classifier with sufficient training data, and thus will be known. Our next result shows that C-STM is a statistically consistent decision rule.

Proposition 2. Given the tensor and matrix factors for all data in the domain, the classification risk of C-STM, $\mathcal{R}(\mathbf{f_n})$, converges to the optimal Bayes risk almost surely, that is,

$$\mathcal{R}(\mathbf{f_n}) \to \mathcal{R}^* \quad \text{a.s.} \quad n \to \infty$$

if the following conditions are satisfied:

AS.1 The loss function $\mathcal{L}$ is self-calibrated [66], and is $C(W)$ local Lipschitz continuous in the sense that for $|a| \leq W < \infty$ and $|b| \leq W < \infty$

$$|\mathcal{L}(a,y) - \mathcal{L}(b,y)| \leq C(W)|a-b|$$

In addition, we need $\sup_{y\in\{1,-1\}} \mathcal{L}(0,y) \leq L_0 < \infty$.

AS.2 The kernel functions $K_1^{(1)}(\cdot,\cdot)$, $K_1^{(2)}(\cdot,\cdot)$, $K_1^{(3)}(\cdot,\cdot)$, and $K_2^{(1)}(\cdot,\cdot)$ used to compose the coupled tensor kernel (9) are regular vector-based kernels satisfying the universal approximating property. A kernel has this property if it satisfies the following condition. Suppose $\mathcal{X}$ is a compact subset of the Euclidean space $\mathbb{R}^p$, and $C(\mathcal{X}) = \{\mathbf{f} : \mathcal{X} \to \mathbb{R}\}$ is the collection of all continuous functions defined on $\mathcal{X}$. The kernel function is also defined on $\mathcal{X} \times \mathcal{X}$, and its RKHS is $\mathcal{H}$. Then $\forall \mathbf{g} \in C(\mathcal{X})$, $\exists \mathbf{f} \in \mathcal{H}$ such that $\forall \varepsilon > 0$

$$\|\mathbf{g} - \mathbf{f}\|_\infty = \sup_{\mathbf{x}\in\mathcal{X}} |\mathbf{g}(\mathbf{x}) - \mathbf{f}(\mathbf{x})| \leq \varepsilon.$$

AS.3 The kernel functions $K_1^{(1)}(\cdot,\cdot)$, $K_1^{(2)}(\cdot,\cdot)$, $K_1^{(3)}(\cdot,\cdot)$, and $K_2^{(1)}(\cdot,\cdot)$ used to composite the coupled tensor kernel (9) are all bounded, and are satisfying

$$\sqrt{\sup K(\cdot,\cdot)} \leq K_{\max} < \infty,$$

for every kernel function mentioned above.

AS.4 The hyper-parameter in the regularization term $\lambda = \lambda_n$ satisfies:

$$\lambda_n \to 0 \quad \text{as} \quad n \to \infty$$
$$n\lambda_n \to \infty \quad \text{as} \quad n \to \infty.$$

This proposition is an extension of our previous result for the statistical consistency of CP-STM. The proof of this proposition is provided in Appendix A.

6 | SIMULATION STUDY

We present a simulation study to demonstrate the benefit of utilizing C-STM with multimodal data in classification problems. To show the advantage of using multimodalities in C-STM, we include CP-STM from [32], constrained multilinear discriminant analysis (CMDA), and direct general tensor discriminant analysis (DGTDA) from [47] as competitors. These existing approaches can only take a single tensor–matrix as the feature for classification. As a result, they are not able to enjoy the multimodalities in the simulated data. We apply these approaches on every single data modality in our simulated data, and compare their classification performance with C-STM which uses multimodal data.

We generate synthetic data using the idea from [23]. Suppose the two data modalities in our classification problems are

$$\mathcal{X}_{t,1} = \sum_{k=1}^{3} \mathbf{x}_{k,t,1}^{(1)} \circ \mathbf{x}_{k,t,1}^{(2)} \circ \mathbf{x}_{k,t,1}^{(3)},$$
$$\mathbf{X}_{t,2} = \sum_{k=1}^{3} \mathbf{x}_{k,t,2}^{(1)} \circ \mathbf{x}_{k,t,2}^{(2)}, \tag{23}$$

where $\mathcal{X}_{t,1}$ are three-way tensors in the size of 30 by 20 by 10. $\mathbf{X}_{t,2}$ are matrices in the size of 50 by 10. Both of them have CP ranks equal to 3. To generate data for the simulation study, we first generate the latent factors (vectors) from various multivariate normal distributions, and then convert these factors into full tensors $\mathcal{X}_{t,1}$ and matrices $\mathbf{X}_{t,2}$ using Equation (23). The multivariate normal distributions we used to generate columns of the latent factors in Equation (23) are specified in Table 1 below. In Table 1, we use $c = 1, 2$ to denote data from two different classes.

There are eight different cases in our simulation study. In Cases 1–5, one of the tensor factors and the matrix factors are generated from different multivariate normal distributions for data in different classes. This means the tensor and matrix data both contain certain class information (discriminant power) which are different in different data modalities. Notice that the discriminant power in one of the tensor factor remains the same among Cased 1–5, while the power in the matrix factor increases. Cases 6 and 7 assume the class information exists only in a single data modality. In Case 6, only one of the tensor factors are generated from different distributions for data in different classes. This factor then becomes the matrix factor in Class 7. In Case 8, the shared factors are sampled from different distributions, meaning that both tensor and matrix data modalities have class information. However, such class information are from the shared factors are the same between different modalities.

TABLE 1 Distribution specifications for simulation study **MVN** stands for multivariate normal distribution. **I** indicates identity matrices. Bold numbers are vectors whose elements are all equal to the numbers

		Tensor factors		Shared factors	Matrix factors
Simulation	c	$\mathbf{x}_{k,t,1}^{(1)}$	$\mathbf{x}_{k,t,1}^{(2)}$	$\mathbf{x}_{k,t,1}^{(3)} = \mathbf{x}_{k,t,2}^{(2)}$	$\mathbf{x}_{k,t,2}^{(1)}$
Case 1	1	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$
	2	$\mathbf{MVN}(\mathbf{1.5}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1.25}, \mathbf{I})$
Case 2	1	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$
	2	$\mathbf{MVN}(\mathbf{1.5}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1.5}, \mathbf{I})$
Case 3	1	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$
	2	$\mathbf{MVN}(\mathbf{1.5}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1.75}, \mathbf{I})$
Case 4	1	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$
	2	$\mathbf{MVN}(\mathbf{1.5}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{2}, \mathbf{I})$
Case 5	1	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$
	2	$\mathbf{MVN}(\mathbf{1.5}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{2.25}, \mathbf{I})$
Case 6	1	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$
	2	$\mathbf{MVN}(\mathbf{2}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$
Case 7	1	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$
	2	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{2}, \mathbf{I})$
Case 8	1	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$
	2	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{2}, \mathbf{I})$	$\mathbf{MVN}(\mathbf{1}, \mathbf{I})$

For each simulation case, we generate 50 pairs of tensor and matrix from both classes, collecting 100 pairs of observations in total. We then perform a random training and testing set separation by randomly choosing 20 samples as the testing set, and use the remaining data as the training set. The random selection of testing set is conducted in a stratified sampling manner such that the proportion of samples from each class remains the same in both training and testing sets. For all models, we report the model prediction accuracy, the proportion of correct predictions over total predictions, on the testing set as the performance metric. The random training and testing set separation is repeated for 50 times and the average prediction accuracy of these 50 repetitions for all the cases are reported in Figure 3. In addition, the SDs are illustrated by the error bars in the figure. The results of CP-STM, CMDA, and DGTDA with tensor data are denoted by CP-STM1, CMDA1, and DGTDA1, respectively, in the figure. The results using matrix data are denoted by CP-STM2, CMDA2, and DGTDA2.

From Figure 3, we can conclude that our C-STM has a more favorable performance in this multimodal classification problem comparing with other competitors. Its accuracy rates are significantly larger than other methods in most cases. Particularly, we can see that the accuracy rates of C-STM (orange) are increasing from Case 1 to Case 5, while the accuracy rates of CP-STM using tensor data remain the same. This is because the difference between class mean vectors for the first tensor factor does not change from Case 1 to Case 5. However, the gap between class mean vectors in matrix factor increases. Due to this fact, both C-STM and CP-STM (yellow) which utilize matrix data are getting better performance from Case 1 to Case 5. More importantly, C-STM always outperforms CP-STM with matrix data as it enjoys the extra class information from multimodalities. In Case 6 and Case 7 where class information are in single data modalities, the advantage of C-STM is not as significant as the previous cases, though its performances are slightly better than CP-STM. This indicates C-STM can provide robust classification results even when extra data modalities do not provide any other class information, as it can extract more accurate estimates of the factors in the decomposition step. In Case 8 where the class information is from the shared factors, C-STM recovers the shared factors accurately and provides significantly better classification accuracy. Through this simulation, we showed that C-STM has a clear advantage of using multimodal data in classification problems, and is robust to redundant data modalities.

6.1 | Kernel selection

In this section, we evaluate and justify the choice of the kernel function presented in (9). In this formulation, the

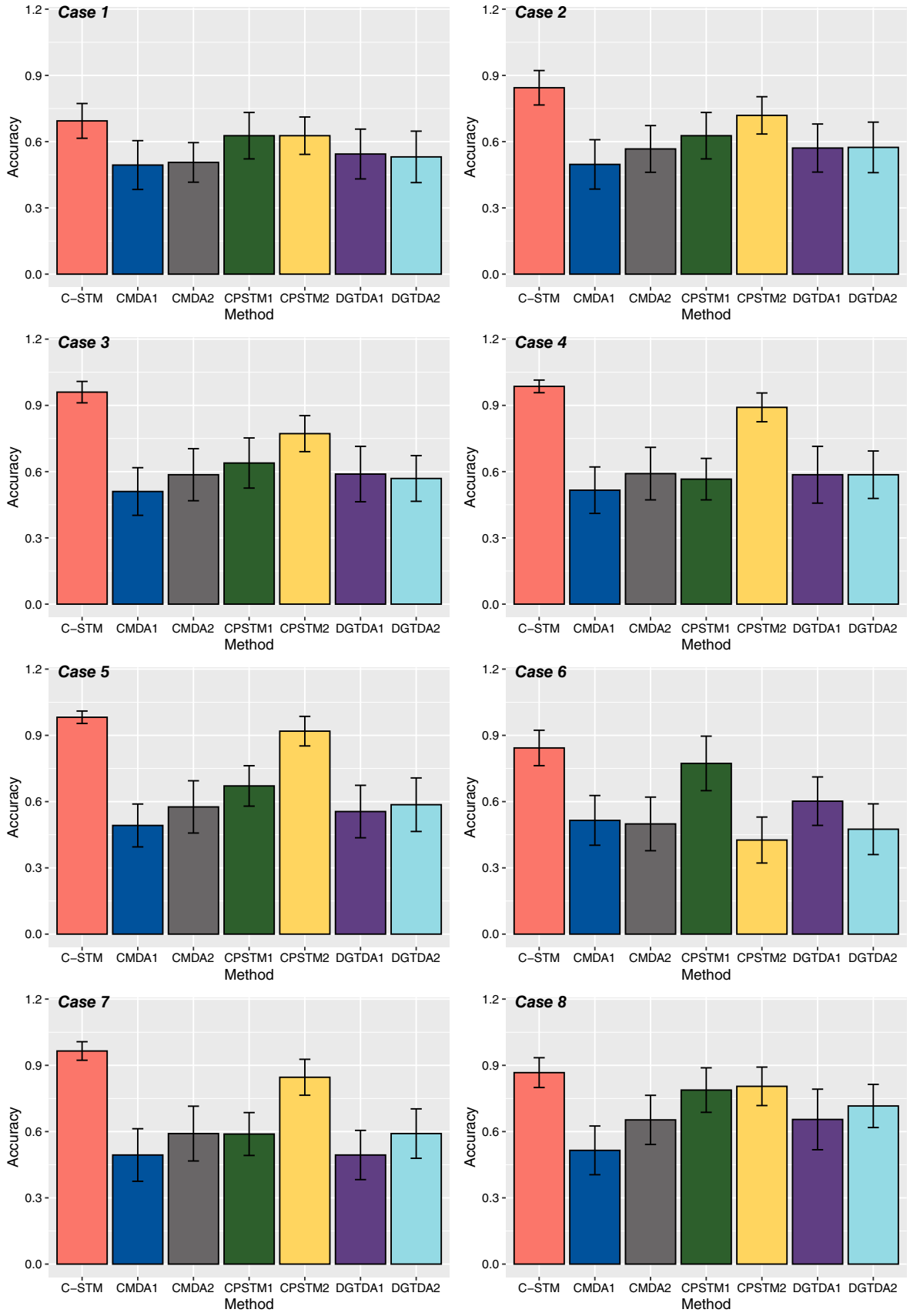


FIGURE 3 Simulation result: average accuracy rates shown in bar plots; SD of accuracy rates shown by error bars

individual and coupled modes for each modality are first separated and then the individual modes of each modality are combined as a tensor kernel function. The kernels from the individual modes are then added to those from the coupled modes to obtain the final form where the weights for the individual and coupled parts can be optimized as discussed in the Appendix. This kernel formulation separates the coupled and individual information and integrates them as a linear combination. Although this is not the only way to integrate kernels, the relatively simple structure of such combination provides us with several benefits such as interpretability, convenient parameter tuning, and generalizability for multimodal data. With Equation (9), it is possible to explain the contribution of different data modalities to discrimination power by looking at the weights parameters. Further, with linear combination of kernels, the weight parameters can be tuned with the different approaches introduced in [29] such as group lasso. Even though we do not adopt these tuning techniques in this work, it still shows the advantage of choosing such a combination and can be the foundation for future work. Lastly, this kernel combination can be extended for data with more modalities easily since kernels are appended linearly.

Besides the aforementioned reasons, we also provide numerical experiments to illustrate the performance of our choice against other kernel combination choices. In these experiments, the factor sizes are the same ($\mathcal{X}_1 \in \mathbb{R}^{40\times40\times40}$, and $\mathbf{X}_2 \in \mathbb{R}^{40\times40}$, $r = 3$) so that the kernels are balanced across the modes. We consider two cases, that is, Case 8 in Table 1, and Case 9 where the columns of the latent factors corresponding to all individual and coupled modes of the second class are from the distribution $\mathbf{MVN}(\mathbf{2}, \mathbf{I})$. Although there can be many different kernel combinations, we select four particular formulations for comparison as they can be a basis for other choices. The particular formulations are the weighted combination of individual kernels from all modes (K2), the weighted combination of the tensor kernels corresponding to the two modalities (K3) and the tensor kernel corresponding to all modes across modalities (K4). The formulations for the different kernels are given in Table 2. We report average classification accuracy across 50 simulations, where the simulated tensors are randomly initialized.

In Table 3, we can see that the kernel selection Schemes K1 and K2 perform the best. K3 performs slightly worse as it is not as flexible as the previous two. Finally, K4 performs the worst as it is affected by all modes simultaneously, and cannot generalize well. Although the difference in performance between the kernels is not significant, K3 and K4 cannot determine whether the observed class differences are due to an individual mode or a coupled mode. Thus, K1 and K2 are better in terms of explaining the results. For Case 8, in most cases, cross-validation across a range of weight parameters for K1 and K2 yields $w_2 = 1$ and $w_3 = 1$, respectively, and the remaining weights are equal to zero. This directly identifies the source of discriminability and allows for better interpretability, which is not possible for K3 and K4. Finally, K1 has less number of parameters than K2 and this can be advantageous in cases with high number of modalities. The smaller number of parameters makes cross-validation simpler, while still allowing for some interpretability.

TABLE 2 Various kernel combination schemes. Note that $K_2^{(2)} = K_1^{(3)}$

	Combination scheme
K1	$w_1 K_1^{(1)} K_1^{(2)} + w_2 K_1^{(3)} + w_3 K_2^{(1)}$
K2	$w_1 K_1^{(1)} + w_2 K_1^{(2)} + w_3 K_1^{(3)} + w_4 K_2^{(1)}$
K3	$w_1 K_1^{(1)} K_1^{(2)} K_1^{(3)} + w_2 K_2^{(1)} K_2^{(2)}$
K4	$K_1^{(1)} K_1^{(2)} K_1^{(3)} K_2^{(1)}$

TABLE 3 Classification accuracy using different kernel combinations

	K1	K2	K3	K4
Case 9	0.91 ± 0.036	0.9 ± 0.043	0.87 ± 0.038	0.82 ± 0.045
Case 8	0.90 ± 0.039	0.9 ± 0.038	0.88 ± 0.036	0.83 ± 0.06

7 | TRIAL CLASSIFICATION FOR SIMULTANEOUS EEG-FMRI DATA

In this section, we present the application of the proposed method on simultaneous EEG-fMRI data. The simultaneous EEG-fMRI is one of the most popular noninvasive multimodal brain imaging techniques to study human brain function. EEG records electrical activity from the scalp resulting from ionic current within the neurons of the brain. Its millisecond temporal resolution makes it possible to record event-related potentials that occur in response to visual, auditory and sensory stimuli [1, 70]. Although EEG provides high temporal resolution, its spatial resolution is limited by the number of electrodes placed on the scalp and thus provides less spatial resolution compared to other neuroimaging modalities such as MRI and PET. As a result, it has been commonplace to record EEG data in conjunction with a high spatial resolution modality. As another powerful tool in studying human brain function, blood oxygenation level-dependent (BOLD) fMRI provides signals with much higher spatial resolution to reflect hemodynamic changes in blood oxygenation level at all voxels related to neuronal activities

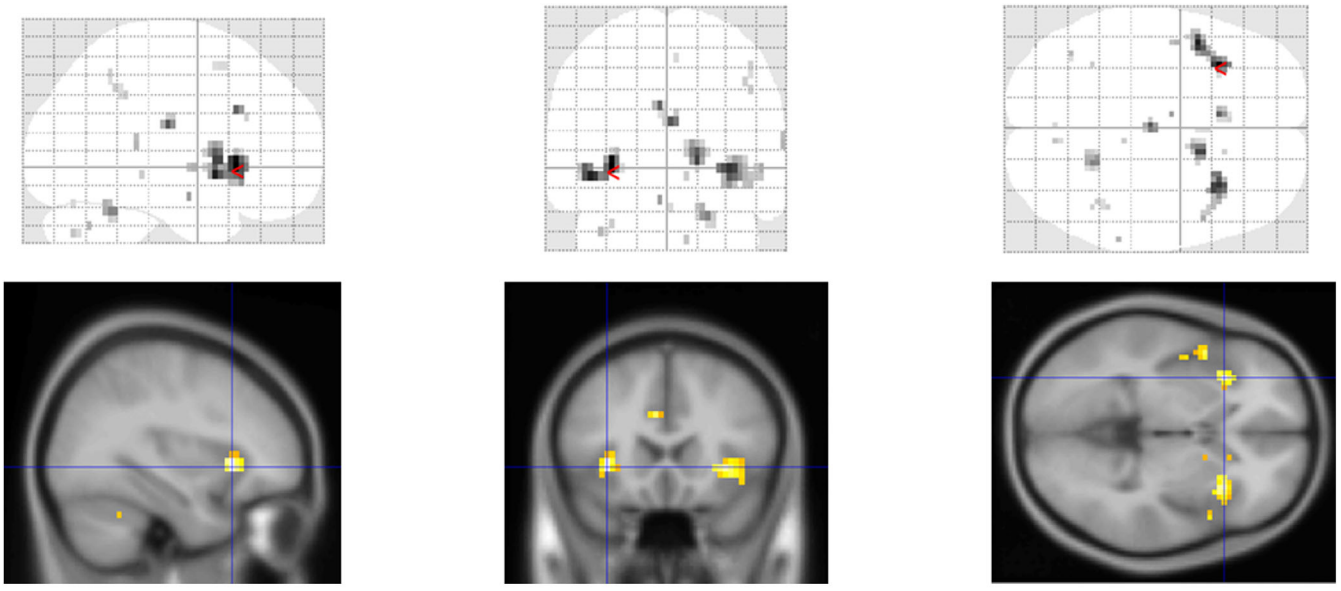
[11, 24, 41, 59]. Recording simultaneous EEG and fMRI can provide high-resolution information at both the spatial and temporal dimensions at the same time. Thus, developing novel machine learning techniques to utilize such multimodal data is of great significance. In this application, we apply our C-STM model to a binary trial classification problem on a simultaneous EEG-fMRI data.

The data are obtained from the study of Walz et al. [72]. In this study, there are 17 individuals (6 females, average age 27.7) participated in 3 runs each of analogous visual and auditory oddball paradigms. The 375 (125 per run) total stimuli per task were presented for 200 ms each with a 2–3 s uniformly distributed variable inter-trial interval. A trial is defined as a time window in which subjects receive stimuli and make responses. In the visual task, a large red circle on isoluminant gray backgrounds was considered as the target stimuli, and a small green circle were the standard stimuli. For the auditory task, the standard and oddball stimuli were, respectively, 390 Hz pure tones and broadband sounds which sound like "laser guns." During the experiment, the stimuli were presented to all subjects, and their EEG and fMRI data are collected simultaneously and continuously. We obtain the EEG and fMRI data from OpenNeuro website (https://openneuro.org/datasets/ds000116/versions/00003). We utilize both EEG and fMRI in this data set with our C-STM model to class stimulus types in all the trials. Through our numerical study, we want to demonstrate the fact our C-STM model enjoys the advantage of data multimodality and provides more accurate class predictions. The data from Subject 4 are dropped since its fMRI data are corrupted. Due to the fact that the number of trials from each subject is different, we further provide Table B1 in Appendix B to show the number of trials for each subject.

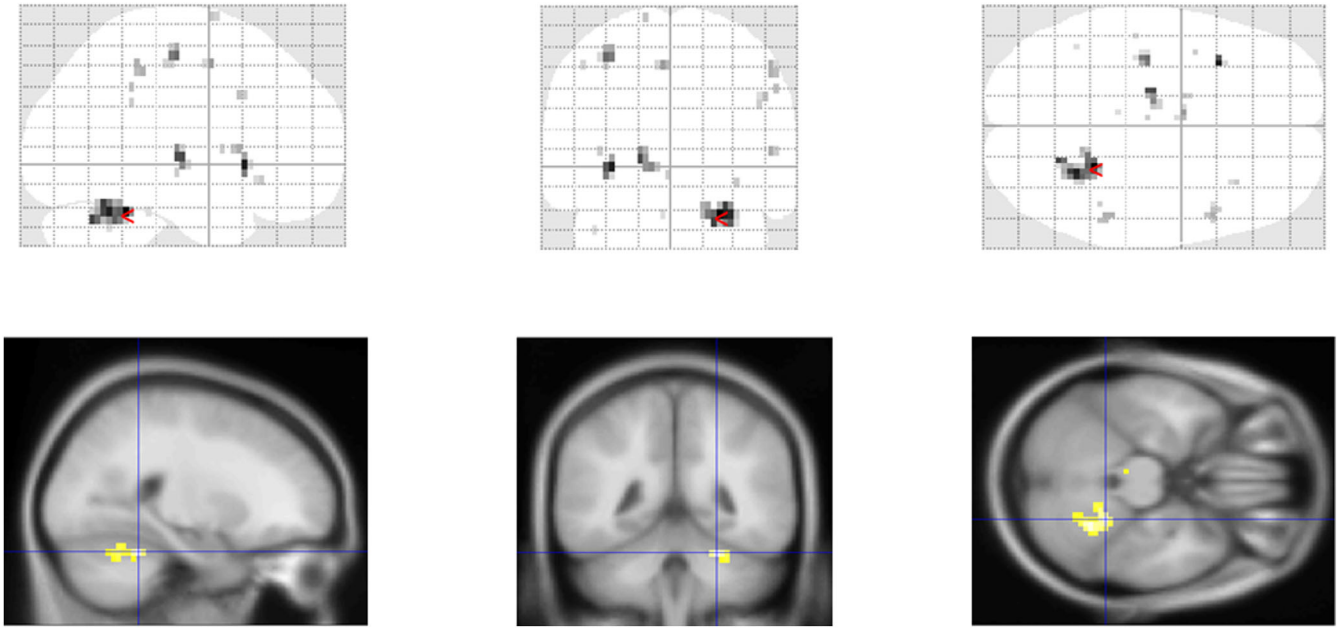
We preprocess both the EEG and fMRI data with statistical parametric mapping (SPM 12) [9] and Matlab. The EEG data are collected by a custom built MR-compatible EEG system with 49 channels. Walz et al. [72] provide a version of re-referenced EEG data with 34 channels which are used in our experiment. This version of EEG data is sampled at 1000 Hz and is downsampled to 200 Hz at the beginning of pre-processing. We then remove both low-frequency and high-frequency noise in the data using SPM filter functions. As the last step of EEG preprocessing, we define trials from brain imaging data structure files [57] and extract EEG data epochs recorded within the trial-related time windows. The time window for each trial is considered to go from 100 ms before the stimulus onset until 500 ms after the stimulus. For each trial, we construct a three-mode tensor corresponding to the EEG data for all subjects where the modes represent channel × time × subject. We denote it as $\mathcal{X}_{t,1} \in \mathbb{R}^{34\times121\times16}$. The fMRI data are collected by 3 T Philips Achieva MR Scanner with 170 volumes (TR = 2 s) per session. Each 3D volume contains 32 slices. The voxel size in the image is $3\times3\times4$ mm. For each subject, we realign all the fMRI volumes from multiple sessions to the mean volume, and co-register the participant's T1-weighted anatomical scan to the mean fMRI volume. Next, we normalize all the fMRI volumes to match the MNI brain template [42] by creating segments from co-registered T1-weighted scan, and keep the voxel size as $3\times3\times4$ mm. All normalized fMRI volumes are then smoothed by 3D Gaussian kernels with full width at half maximum (FWHM) parameter being $8\times8\times8$. After the preprocessing, we further perform a regular statistical analysis [50, 75] to extract fMRI volumes from visual and auditory stimulus related voxels. Such data are also known as ROI data. We describe the ROI voxel identification and data extraction in Appendix B. We extract fMRI volumes from 178 voxels (in Figure 4a) for auditory oddball tasks, and 112 voxels for auditory tasks. As a result, fMRI data are modeled by matrices whose rows and columns stand for voxels and subjects: $\mathcal{X}_{t,2} \in \mathbb{R}^{16\times178}$ for auditory task data, and $\mathcal{X}_{t,2} \in \mathbb{R}^{16\times112}$ for visual task data. There is no time mode in fMRI data because the trial duration is less than the repetition time of fMRI (time for obtaining a single 3D volume fMRI). For each trial, there is only one 3D scan of fMRI collected from a single subject. The ROI data then become a vector for this subject in the trial as we extract volumes from the ROI.

To classify trials with oddball and standard stimulus, we collect 140 multimodal data samples $(\mathcal{X}_{t,1}, \mathcal{X}_{t,2})$ from auditory tasks, and 100 samples from visual tasks. For both types of tasks, the numbers of oddball and standard trials are equal. We consider the trials with oddball stimulus as the positive class, and the trials with standard stimulus as the negative class. Like the procedures in our simulation study, we select 20% of data as testing set, and use the remaining 80% for model estimation and validation. The classification accuracy, precision (positive predictive rate), sensitivity (true positive rate), and specificity (true negative rate) of classifiers are calculated using the test set at each experiment. The experiment is repeated multiple times, and the average accuracy, precision, sensitivity, and specificity, and their SDs (in subscripts) are reported in Table 4. The single mode classifiers CP-STM, CMDA, and DGTDA are also applied on either EEG or fMRI data as a comparison. The single-mode classifiers applied on EEG data are denoted by appending the number "1" after their names, and those applied on fMRI data are denoted by appending the number "2." The area under the curve (AUC) for all the classifiers is also reported in Table 4.

The results in Table 4 show that the trial classification accuracy for C-STM using multimodal data is better than any classifier based on single modality with a



(a) Auditory Task



(b) Visual Task

FIGURE 4 Region of interest (ROI)

significant improvement in terms of average accuracy rates and average AUC values. This improvement is observed for classification of both auditory and visual tasks. This observation agrees to the conclusion from our simulation study. Similar to our simulation study, the tensor discriminant analysis does not work as well as CP-STM and C-STM. In addition, it is obvious that the performance of tensor discriminant analysis using fMRI data is better than using EEG data. This is within our expectation, since the regions we extracted from fMRI data are identified by group-level fMRI statistical analysis (see Appendix B). The data in these regions have already shown significant differences between different trials in the traditional study, and thus are easy to classify. On the other hand, there is no prior analysis and feature extraction procedure applied on EEG data, leaving a low signal to noise ratio in EEG data. However, C-STM still can take advantage of using EEG data and further increase the classification accuracy, highlighting its robustness and potential in processing noisy multimodal tensor data.

TABLE 4 Real data result: simultaneous EEG-fMRI data trial classification (mean of performance metrics with SDs in subscripts)

Task	Method	Accuracy	Precision	Sensitivity	Specificity	AUC
Auditory	C-STM	$\mathbf{0.89}_{0.05}$	$0.83_{0.07}$	$1.00_{0.00}$	$0.77_{0.11}$	$\mathbf{0.89}_{0.06}$
	CP-STM1	$0.80_{0.08}$	$0.71_{0.11}$	$1.00_{0.00}$	$0.60_{0.12}$	$0.78_{0.06}$
	CP-STM2	$0.83_{0.06}$	$0.76_{0.07}$	$0.99_{0.05}$	$0.65_{0.11}$	$0.82_{0.05}$
	CDMA1	$0.55_{0.10}$	$0.51_{0.09}$	$0.96_{0.09}$	$0.20_{0.21}$	$0.55_{0.06}$
	CDMA2	$0.67_{0.09}$	$0.61_{0.11}$	$0.92_{0.07}$	$0.46_{0.14}$	$0.70_{0.08}$
	DGTDA1	$0.55_{0.09}$	$0.51_{0.09}$	$0.94_{0.07}$	$0.23_{0.12}$	$0.59_{0.06}$
	DGTDA2	$0.67_{0.09}$	$0.60_{0.10}$	$0.90_{0.09}$	$0.46_{0.13}$	$0.68_{0.08}$
Visual	C-STM	$\mathbf{0.86}_{0.06}$	$0.82_{0.09}$	$0.93_{0.07}$	$0.77_{0.12}$	$\mathbf{0.86}_{0.06}$
	CP-STM1	$0.76_{0.08}$	$0.66_{0.11}$	$1.00_{0.00}$	$0.54_{0.12}$	$0.78_{0.05}$
	CP-STM2	$0.77_{0.08}$	$0.70_{0.11}$	$0.98_{0.08}$	$0.58_{0.17}$	$0.77_{0.07}$
	CDMA1	$0.53_{0.12}$	$0.52_{0.11}$	$0.94_{0.11}$	$0.11_{0.18}$	$0.54_{0.08}$
	CDMA2	$0.65_{0.13}$	$0.61_{0.14}$	$0.91_{0.09}$	$0.43_{0.19}$	$0.66_{0.09}$
	DGTDA1	$0.56_{0.11}$	$0.54_{0.11}$	$0.94_{0.06}$	$0.17_{0.12}$	$0.56_{0.07}$
	DGTDA2	$0.64_{0.10}$	$0.60_{0.13}$	$0.86_{0.10}$	$0.44_{0.18}$	$0.64_{0.07}$

Abbreviations: AUC, area under the curve; CMDA, constrained multilinear discriminant analysis; C-STM, coupled support tensor machine; DGTDA, direct general tensor discriminant analysis; EEG, electroencephalography; fMRI, functional magnetic resonance imaging.

8 | CONCLUSION

In this work, we have proposed a novel C-STM classifier for multimodal data by combining the ACMTF and STM. The most distinctive feature of this classifier is its ability to integrate features across different modalities and structures. The proposed approach can simultaneously take matrix- and tensor-shaped data for classification and can be easily extended to inputs with more than two modalities. The coupled matrix–tensor decomposition unveils the intrinsic correlation structure between data across different modalities, making it possible to integrate information from multiple sources efficiently. Such decomposition also makes the whole method robust and applicable to large-scale noisy data with missing values. The newly designed kernel functions in C-STM provide feature-level information fusion, combining discriminant information from different modalities. Moreover, the kernel formulation makes it possible to utilize the most discriminative features from each modality by tuning the weight parameters in the function. Our theoretical results demonstrate that the C-STM decision rule is statistically consistent.

The most important theoretical extension of our current approach would be the development of excess risk for C-STM. In particular, we are looking for an explicit expression for the excess risk in terms of data factors from multiple modalities to quantify the contribution of every single modality in minimizing the excess risk. By doing so, we are able to interpret the importance of each data modality in classification tasks. In addition, quantifying the uncertainty of tensor and matrix factors estimation and their impact on the excess risk will build the foundation to the next level.

Future work will focus on learning the weight parameters in the kernel function via optimization. As Gönen and Alpaydın [29] introduced, the weights in the kernel function can be further estimated by including a group lasso penalty in the objective function. Such a weight estimation procedure can identify the most significant data components and reduce the burden of parameter selection. In addition, the proposed framework can be extended to multimodal tensors with more than two modalities, and for regression problems.

In conclusion, we believe C-STM offers many encouraging possibilities for multimodal data integration and analysis. Its capability of handling multimodal tensor inputs will make it appropriate in many advanced data applications in neuroscience and medical research. We anticipate that this method will play an important role in a variety of applications.

ACKNOWLEDGMENTS

The authors like to thank the reviewers and the editors for their helpful comments. This work was in part supported by NSF DMS-1924724.

DATA AVAILABILITY STATEMENT

Data for the simulation study is generated by our own procedures and the codes are provided for this. Data for the real data experiments are available online and the identifiers and necessary references are provided within the manuscript.

ORCID

Seyyid Emre Sofuoglu https://orcid.org/0000-0003-3699-0053

REFERENCES

1. R. Abreu, A. Leal, and P. Figueiredo, *EEG-informed fMRI: A review of data analysis methods*, Front. Hum. Neurosci. 12 (2018), 29.
2. E. Acar, T. G. Kolda, and D. M. Dunlavy, All-at-once optimization for coupled matrix and tensor factorizations. *arXiv Preprint arXiv:1105.3422*, 2011
3. E. Acar, Y. Levin-Schwartz, V. D. Calhoun, and T. Adali, *ACMTF for fusion of multi-modal neuroimaging data and identification of biomarkers*, 2017 25th Eur. Signal Process. Conf. (EUSIPCO), IEEE, Kos, Greece, 2017, pp. 643–647.
4. E. Acar, *Tensor-based fusion of EEG and fMRI to understand neurological changes in schizophrenia*, 2017 IEEE Int. Symp. Circ. Syst. (ISCAS), IEEE, Baltimore, MD, USA, 2017, pp. 1–4.
5. E. Acar, E. E. Papalexakis, G. Gürdeniz, M. A. Rasmussen, A. J. Lawaetz, M. Nilsson, and R. Bro, *Structure revealing data fusion*, BMC Bioinform. 15 (2014), no. 1, 1–17.
6. E. Acar, C. Schenker, Y. Levin-Schwartz, V. D. Calhoun, and T. Adali, *Unraveling diagnostic biomarkers of schizophrenia through structure-revealing fusion of multi-modal neuroimaging data*, Front. Neurosci. 13 (2019), 416.
7. J. S. Anderson, J. A. Nielsen, A. L. Froehlich, M. B. DuBray, T. J. Druzgal, A. N. Cariello, J. R. Cooperrider, B. A. Zielinski, C. Ravichandran, P. T. Fletcher, A. L. Alexander, E. D. Bigler, N. Lange, and J. E. Lainhart, *Functional connectivity magnetic resonance imaging classification of autism*, Brain 134 (2011), no. 12, 3742–3754.
8. A. Argyriou, C. A. Micchelli, and M. Pontil, *When is there a representer theorem? Vector versus matrix regularizers*, J. Mach. Learn. Res. 10 (2009), 2507–2529.
9. J. Ashburner, G. Barnes, C.-C. Chen, J. Daunizeau, G. Flandin, K. Friston, S. Kiebel, J. Kilner, V. Litvak, R. Moran, W. Penny, K. Stephan, P. Zeidman, D. Gitelman, R. Henson, C. Hutton, V. Glauche, J. Mattout, and C. Phillips, *Spm12 manual*, Wellcome Trust Centre for Neuroimaging, London, UK, 2014 p. 2464.
10. F. R. Bach, *Consistency of the group lasso and multiple kernel learning*, J. Mach. Learn. Res. 9 (2008), no. 6.
11. J. Belliveau, D. Kennedy, R. McKinstry, B. Buchbinder, R. Weisskoff, M. Cohen, J. Vevea, T. Brady, and B. Rosen, *Functional mapping of the human visual cortex by magnetic resonance imaging*, Science 254 (1991), no. 5032, 716–719.
12. A. Ben-Hur and W. S. Noble, *Kernel methods for predicting protein–protein interactions*, Bioinformatics 21 (2005), no. suppl_1, i38–i46.
13. K. P. Bennett, M. Momma, and M. J. Embrechts, *Mark: A boosting algorithm for heterogeneous kernel models*, Proc. Eighth ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, Alberta, Canada, 2002, pp. 24–31.
14. X. Bi, X. Tang, Y. Yuan, Y. Zhang, and A. Qu, *Tensors in statistics*, Annu. Rev. Stat. Appl. 8 (2020), 345–368.
15. M. Brett, K. Christoff, R. Cusack, and J. Lancaster, *Using the talairach atlas with the MNI template*, NeuroImage 13 (2001), no. 6, 85.
16. V. D. Calhoun, T. Adali, N. Giuliani, J. Pekar, K. Kiehl, and G. Pearlson, *Method for multimodal analysis of independent source differences in schizophrenia: Combining gray matter structural and auditory oddball functional data*, Hum. Brain Mapp. 27 (2006), no. 1, 47–62.
17. C. Chatzichristos, M. Davies, J. Escudero, E. Kofidis, and S. Theodoridis, *Fusion of EEG and fMRI via soft coupled tensor decompositions*, 2018 26th Eur. Signal Process. Conference (EUSIPCO), Rome, Italy, IEEE, 2018, pp. 56–60.
18. C. Chatzichristos, E. Kofidis, L. De Lathauwer, S. Theodoridis, and S. Van Huffel, Early soft and flexible fusion of EEG and fMRI via tensor decompositions. *arXiv Preprint arXiv:2005.07134*, 2020.
19. M. Christoudias, R. Urtasun, and T. Darrell, *Bayesian localized multiple kernel learning*, University of California, Berkeley, Berkeley, CA, 2009.
20. L. Devroye, L. Györfi, and G. Lugosi, *A probabilistic theory of pattern recognition*, Vol 31, Springer Science & Business Media, Berlin, Germany, 2013.
21. Y. Ding, J. H. Sohn, M. G. Kawczynski, H. Trivedi, R. Harnish, N. W. Jenkins, D. Lituiev, T. P. Copeland, M. S. Aboian, C. Mari Aparici, S. C. Behr, R. R. Flavell, S.-Y. Huang, K. A. Zalocusky, L. Nardo, Y. Seo, R. A. Hawkins, M. H. Pampaloni, D. Hadley, and B. L. Franc, *A deep learning model to predict a diagnosis of alzheimer disease by using 18f-fdg pet of the brain*, Radiology 290 (2019), no. 2, 456–464.
22. R. Durrett, *Probability: Theory and examples*, Vol 49, Cambridge University Press, Cambridge, United Kingdom, 2019.
23. H. Fanaee-T and J. Gama, Simtensor: A synthetic tensor data generator. *arXiv Preprint arXiv:1612.03772*, 2016.
24. M. Filippi, N. DeStefano, V. Dousset, and J. C. McGowan, *MR imaging in white matter diseases of the brain and spinal cord*, Springer, Berlin, Germany, 2005.
25. G. Fung, M. Dundar, J. Bi, and B. Rao, *A fast iterative algorithm for fisher discriminant using heterogeneous kernels*, Proc. Twenty-First Int. Conf. Mach. Learn., 2004, p. 40.
26. M. R. Gahrooei, H. Yan, K. Paynabar, and J. Shi, *Multiple tensor-on-tensor regression: An approach for modeling processes with heterogeneous sources of data*, Technometrics 63 (2021), no. 2, 147–159.
27. G. Gavidia-Bovadilla, S. Kanaan-Izquierdo, M. Mataró-Serrat, A. Perera-Lluna, and ADNI, *Early prediction of alzheimer's disease using null longitudinal model-based classifiers*, PLoS One 12 (2017), no. 1, e0168011.
28. M. Girolami and M. Zhong, *Data integration for classification problems employing Gaussian process priors*, Advances in Neural Information Processing Systems 19, Proc. 2006 Conf., Vancouver, Canada, vol. 19, MIT Press, 2007, p. 465.
29. M. Gönen and E. Alpaydın, *Multiple kernel learning algorithms*, J. Mach. Learn. Res. 12 (2011), 2211–2268.
30. A. R. Groves, C. F. Beckmann, S. M. Smith, and M. W. Woolrich, *Linked independent component analysis for multimodal data fusion*, NeuroImage 54 (2011), no. 3, 2198–2217.

31. W. W. Hager and H. Zhang, *A survey of nonlinear conjugate gradient methods*, Pacific J. Optim. 2 (2006), no. 1, 35–58.
32. L. He, X. Kong, P. S. Yu, X. Yang, A. B. Ragin, and Z. Hao, *Dusk: A dual structure-preserving kernel for supervised tensor learning with applications to neuroimages*, Proc. 2014 SIAM Int. Conf. Data Mining, SIAM, Shenzen, China, 2014, pp. 127–135.
33. L. He, C.-T. Lu, G. Ma, S. Wang, L. Shen, P. S. Yu, and A. B. Ragin, *Kernelized support tensor machines*, Proc. 34th Int. Conf. Mach. Learn., Sydney, Australia, vol. 70, JMLR.org, 2017, pp. 1442–1451.
34. R. N. Henson, H. Abdulrahman, G. Flandin, and V. Litvak, *Multimodal integration of M/EEG and f/MRI data in spm12*, Front. Neurosci. 13 (2019), 300.
35. M. H. Kamstrup-Nielsen, L. G. Johnsen, and R. Bro, *Core consistency diagnostic in parafac2*, J. Chemom. 27 (2013), no. 5, 99–105.
36. E. Karahan, P. A. Rojas-Lopez, M. L. Bringas-Vega, P. A. Valdés-Hernández, and P. A. Valdes-Sosa, *Tensor analysis and fusion of multimodal brain images*, Proc. IEEE 103 (2015), no. 9, 1531–1559.
37. A. Khazaee, A. Ebrahimzadeh, and A. Babajani-Feremi, *Application of advanced machine learning methods on resting-state fMRI network for identification of mild cognitive impairment and Alzheimer's disease*, Brain Imaging Behav. 10 (2016), no. 3, 799–817.
38. M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien, *Efficient and accurate LP-norm multiple kernel learning*, NIPS 22 (2009), 997–1005.
39. T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev. 51 (2009), no. 3, 455–500.
40. J. B. Kruskal, *Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl. 18 (1977), no. 2, 95–138. https://doi.org/10.1016/0024-3795(77)90069-6.
41. K. K. Kwong, J. W. Belliveau, D. A. Chesler, I. E. Goldberg, R. M. Weisskoff, B. P. Poncelet, D. N. Kennedy, B. E. Hoppel, M. S. Cohen, and R. Turner, *Dynamic magnetic resonance imaging of human brain activity during primary sensory stimulation*, Proc. Natl. Acad. Sci. 89 (1992), no. 12, 5675–5679.
42. J. L. Lancaster, D. Tordesillas-Gutiérrez, M. Martinez, F. Salinas, A. Evans, K. Zilles, J. C. Mazziotta, and P. T. Fox, *Bias between MNI and Talairach coordinates analyzed using the ICBM-152 brain template*, Hum. Brain Mapp. 28 (2007), no. 11, 1194–1205.
43. G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, *Learning the kernel matrix with semidefinite programming*, J. Mach. Learn. Res. 5 (2004), 27–72.
44. X. Lei, P. A. Valdes-Sosa, and D. Yao, *EEG/fMRI fusion based on independent component analysis: Integration of data-driven and model-driven methods*, J. Integr. Neurosci. 11 (2012), no. 03, 313–337.
45. P. Li and T. Maiti, *Universal consistency of support tensor machine*, 2019 IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA), Washington, DC, IEEE, 2019, pp. 608–609.
46. Q. Li and L. Li, Integrative factor regression and its inference for multimodal data analysis. *arXiv Preprint arXiv:1911.04056*, 2019.
47. Q. Li and D. Schonfeld, *Multilinear discriminant analysis for higher-order tensor data classification*, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2014), no. 12, 2524–2537.
48. X. Li, D. Xu, H. Zhou, and L. Li, *Tucker tensor regression and neuroimaging analysis*, Stat. Biosci. 10 (2018), no. 3, 520–545.
49. Y. Li, L. Zhang, A. Bozoki, D. C. Zhu, J. Choi, and T. Maiti, *Early prediction of Alzheimer's disease using longitudinal volumetric MRI data from ADNI*, Health Serv. Outcomes Res. Methodol. 20 (2020), no. 1, 13–39.
50. M. A. Lindquist, *The statistical analysis of fMRI data*, Stat. Sci. 23 (2008), no. 4, 439–464.
51. J. Liu, G. Pearlson, A. Windemuth, G. Ruano, N. I. Perrone-Bizzozero, and V. Calhoun, *Combining fMRI and SNP data to investigate connections between brain function and genetics using parallel ICA*, Hum. Brain Mapp. 30 (2009), no. 1, 241–255.
52. S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, and D. Feng, *Early diagnosis of Alzheimer's disease with deep learning*, 2014 IEEE 11th Int. Symp. Biomed. Imaging (ISBI), Beijing, China, IEEE, 2014, pp. 1015–1018.
53. E. F. Lock, *Tensor-on-tensor regression*, J. Comput. Graph. Stat. 27 (2018), no. 3, 638–647.
54. X. Long, L. Chen, C. Jiang, L. Zhang, and A. D. N. Initiative, *Prediction and classification of Alzheimer disease based on quantification of MRI deformation*, PLoS One 12 (2017), no. 3, e0173372.
55. J. C. Morris, C. M. Roe, E. A. Grant, D. Head, M. Storandt, A. M. Goate, A. M. Fagan, D. M. Holtzman, and M. A. Mintun, *Pittsburgh compound B imaging and prediction of progression from cognitive normality to symptomatic Alzheimer disease*, Arch. Neurol. 66 (2009), no. 12, 1469–1475.
56. R. Mosayebi and G.-A. Hossein-Zadeh, *Correlated coupled matrix tensor factorization method for simultaneous EEG-fMRI data fusion*, Biomed. Signal Process. Control 62 (2020), 102071.
57. G. Niso, K. J. Gorgolewski, E. Bock, T. L. Brooks, G. Flandin, A. Gramfort, R. N. Henson, M. Jas, V. Litvak, J. T. Moreau, R. Oostenveld, J.-M. Schoffelen, F. Tadel, J. Wexler, and S. Baillet, *Meg-BIDS, the brain imaging data structure extended to magnetoencephalography*, Sci. Data 5 (2018), no. 1, 1–5.
58. J. Nocedal and S. Wright, *Numerical optimization*, Springer Science & Business Media, Berlin, Germany, 2006.
59. S. Ogawa, T.-M. Lee, A. R. Kay, and D. W. Tank, *Brain magnetic resonance imaging with contrast dependent on blood oxygenation*, Proc. Nat. Acad. Sci. 87 (1990), no. 24, 9868–9872.
60. Y. Pan, Q. Mai, and X. Zhang, *Covariate-adjusted tensor classification in high dimensions*, J. Am. Stat. Assoc. 114:527 (2018), 1–15.
61. P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy, *Gene functional classification from heterogeneous data*, Proc. Fifth Annu. Int. Conf. Comput. Biol., Quebec, Canada, 2001, pp. 249–255.
62. M. J. Powell, *Nonconvex minimization calculations and the conjugate gradient method. Numerical analysis*, Springer, Berlin, Germany, 1984, 122–141.
63. S. Qiu and T. Lane, *A framework for multiple kernel support vector regression and its applications to siRNA efficacy prediction*, IEEE/ACM Trans. Comput. Biol. Bioinform. 6 (2008), no. 2, 190–199.
64. K. A. Schindlbeck and D. Eidelberg, *Network imaging biomarkers: Insights and clinical applications in Parkinson's disease*, Lancet Neurol. 17 (2018), no. 7, 629–640.
65. S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*, Cambridge University Press, Cambridge, 2014.

66. I. Steinwart and A. Christmann, *Support vector machines*, Springer Science & Business Media, Berlin, Germany, 2008.
67. J. Sui, G. Pearlson, A. Caprihan, T. Adali, K. A. Kiehl, J. Liu, J. Yamamoto, and V. D. Calhoun, *Discriminating schizophrenia and bipolar disorder by fusing fMRI and DTI in a multimodal CCA+ joint ICA model*, NeuroImage 57 (2011), no. 3, 839–855.
68. H. Tanabe, T. B. Ho, C. H. Nguyen, and S. Kawasaki, *Simple but effective methods for combining kernels in computational biology*, 2008 IEEE Int. Conf. Res. Innov. Vis. Fut. Comput. Commun. Technol., Ho Chi Minh City, Vietnam, IEEE, 2008, pp. 71–78.
69. D. Tao, X. Li, W. Hu, S. Maybank, and X. Wu, *Supervised tensor learning*, Fifth IEEE Int. Conf. Data Mining (ICDM'05), Rio de Janerio, Brazil, IEEE, 2005, 8 pp.
70. M. Teplan, *Fundamentals of EEG measurement*, Measure. Sci. Rev. 2 (2002), no. 2, 1–11.
71. M. Varma and D. Ray, *Learning the discriminative power-invariance trade-off*, 2007 IEEE 11th Int. Conf. Comput. Vis., Rio de Janerio, Brazil, IEEE, 2007, pp. 1–8.
72. J. M. Walz, R. I. Goldman, M. Carapezza, J. Muraskin, T. R. Brown, and P. Sajda, *Simultaneous EEG-fMRI reveals temporal evolution of coupling between supramodal cortical attention networks and the brainstem*, J. Neurosci. 33 (2013), no. 49, 19212–19222.
73. P. Wolfe, *Convergence conditions for ascent methods*, SIAM Rev. 11 (1969), no. 2, 226–235.
74. P. Wolfe, *Convergence conditions for ascent methods. ii: Some corrections*, SIAM Rev. 13 (1971), no. 2, 185–188.
75. K. J. Worsley, C. H. Liao, J. Aston, V. Petre, G. Duncan, F. Morales, and A. Evans, *A general statistical analysis for fMRI data*, NeuroImage 15 (2002), no. 1, 1–15.
76. T. Zhang, *Statistical behavior and consistency of classification methods based on convex risk minimization*, Ann. Stat. 32 (2004), no. 1, 56–85.
77. H. Zhou, L. Li, and H. Zhu, *Tensor regression with applications in neuroimaging data analysis*, J. Am. Stat. Assoc. 108 (2013), no. 502, 540–552.
78. G. Zoutendijk, Computational methods in nonlinear programming Studies in Optimization 1. (1970), 125.

How to cite this article: P. Li, S. E. Sofuoglu, S. Aviyente, and T. Maiti, *Coupled support tensor machine classification for multimodal neuroimaging data*, Stat. Anal. Data Min.: ASA Data Sci. J. (2022), 1–22. https://doi.org/10.1002/sam.11587

APPENDIX A. PROOF OF THEOREM 2

Proof. To prove the proposition 2, we introduce few more notations here. Let $\mathcal{L}$ be the loss function satisfying the condition **AS.2**. We denote the classification risk for an arbitrary decision function, $\mathbf{f}$, as

$$\mathcal{R}_{\mathcal{L}}(\mathbf{f}) = \mathbb{E}_{\mathcal{X}\times\mathcal{Y}}\mathcal{L}(y, \mathbf{f}(\mathcal{X})) = \int \mathcal{L}(y, \mathbf{f}(\mathcal{X}))d\mathbb{P}.$$

The expectation is taken over the joint distribution of $\mathcal{X} \times \mathcal{Y}$. Notice that this risk notation, $\mathcal{R}_{\mathcal{L}}(\mathbf{f})$, is different from our notation $\mathcal{R}(\mathbf{f})$ in Section 5 since we use the Lipschitz continuous loss $\mathcal{L}$ instead of the "zero–one" loss to measure the classification error. $\mathcal{L}$ is also called surrogate loss for classification problems. Examples of such surrogate loss functions include hinge loss and squared hinge loss. Comparison of these loss functions and their statistical properties can be found in [76]. If we denote the Bayes risk under the surrogate loss $\mathcal{L}$ as $\mathcal{R}^*_{\mathcal{L}}$, that is, $\mathcal{R}^*_{\mathcal{L}} = \min \mathcal{R}_{\mathcal{L}}(\mathbf{f})$ for all measurable function f, then the result from [76] says $\mathcal{R}_{\mathcal{L}}(\mathbf{f_n}) \to \mathcal{R}^*_{\mathcal{L}}$ indicates $\mathcal{R}(\mathbf{f_n}) \to \mathcal{R}^*$ for any decision rule $\{\mathbf{f_n}\}$. This conclusion holds as long as the surrogate loss is "self-calibrated" [66]. Since we use hinge loss in our problem, and hinge loss is known to be Lipschitz and self-calibrated, our assumption **AS.2** holds in our discussion. Thus, we only need to show $\mathcal{R}_{\mathcal{L}}(\mathbf{f_n}) \to \mathcal{R}^*_{\mathcal{L}}$ for the proof of our Proposition 2.

Given the tuning parameter λ satisfying condition **AS.4**, we denote

$$\mathbf{f}^{\lambda}_{\mathbf{n}} = \underset{\mathbf{f}\in\mathcal{H}}{\arg\min} \quad \lambda \cdot \|\mathbf{f}\|^2 + \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}\left(\mathbf{f}(\mathcal{X}_i), y_i\right),$$

where $\mathcal{H}$ is the RKHS generated by the kernel function (9). As we mentioned in Section 3, $\mathcal{H}$ is also known as the collection of functions which are in the form of Equation (11). Now we further assume

$$\mathbf{f}^{\lambda} = \underset{\mathbf{f}\in\mathcal{H}}{\arg\min} \quad \lambda \cdot \|\mathbf{f}\|^2 + \mathcal{R}_{\mathcal{L}}(\mathbf{f}).$$

Then $\mathbf{f}^{\lambda}$ is the optimal decision function from $\mathcal{H}$ such that it minimizes the expected risk. Comparing $\mathbf{f}^{\lambda}_{\mathbf{n}}$ with $\mathbf{f}^{\lambda}$, we can understand that $\mathbf{f}^{\lambda}$ is the version of $\mathbf{f}^{\lambda}_{\mathbf{n}}$ when the size of training data is as large as possible. If we denote $\mathcal{R}_{\mathcal{L},T_n}(\mathbf{f}) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}\left(\mathbf{f}(\mathcal{X}_i), y_i\right)$, then $\mathcal{R}_{\mathcal{L},T_n}(\mathbf{f})$ is a sample estimate of $\mathcal{R}_{\mathcal{L}}(\mathbf{f})$. With $\mathbf{f}^{\lambda}$, we can show that

$$\begin{aligned}|\mathcal{R}_{\mathcal{L}}\left(\mathbf{f}^{\lambda}_{\mathbf{n}}\right) - \mathcal{R}^*_{\mathcal{L}}| \le |\mathcal{R}_{\mathcal{L}}\left(\mathbf{f}^{\lambda}_{\mathbf{n}}\right) \\ - \mathcal{R}_{\mathcal{L}}\left(\mathbf{f}^{\lambda}\right)| + |\mathcal{R}_{\mathcal{L}}\left(\mathbf{f}^{\lambda}\right) - \mathcal{R}^*_{\mathcal{L}}|\end{aligned}$$

through triangular inequality. Since the Bayes risk under loss function $\mathcal{L}$ is defined as $\mathcal{R}^* = \min_{\mathbf{f}:\mathcal{X}\to\mathcal{Y}} \mathcal{R}(\mathbf{f})$ over all functions defined on $\mathcal{X}$, we can immediately show that

$$|\mathcal{R}\left(\mathbf{f}^{\lambda}\right) - \mathcal{R}^*| \le \mathbb{E}_{(\mathcal{X}\times\mathcal{Y})}|\mathcal{L}\left(y, \mathbf{f}^{\lambda}(\mathcal{X})\right) - \mathcal{L}\left(y, \mathbf{f}^{*}(\mathcal{X})\right)|$$

$$\leq C(K_{\max}) \sup |\mathbf{f}^{\lambda} - \mathbf{f}^{*}|$$
$$\leq C(K_{\max}) \cdot \varepsilon. \tag{A1}$$

This is the result of using condition **AS.1** and **AS.2** in the Proposition 2. $\mathbf{f}^{\lambda}$ is in the RKHS and thus bounded by some constant depending on K_{max}. $\mathbf{f}^{*}$ is also continuous on compact subspace $\mathcal{X}$ (because all the tensor components considered are bounded in condition **AS.1**) and thus is bounded. The universal approximating property in condition **AS.3** makes Equation (A1) vanishes as ε goes to zero. Thus, the consistency result can be established if we show $|\mathcal{R}\left(\mathbf{f}_{\mathbf{n}}^{\lambda}\right) - \mathcal{R}\left(\mathbf{f}^{\lambda}\right)|$ converges to zero. This can be done with Rademacher complexity (see chap. 26 in [65]).

From the objective function (10), we have

$$\mathcal{R}_{\mathcal{L},T_n}(\mathbf{f_n}) + \lambda_n \|\mathbf{f_n}\|^2 \leq L_0 \tag{A2}$$

under condition **AS.2** when we simply let $\mathbf{f} = 0$ as a naive classifier. Thus, $\left\|\mathbf{f_n}\right\| \leq \sqrt{\frac{L_0}{\lambda_n}}$. Let $M_n = \sqrt{\frac{L_0}{\lambda_n}}$. $\mathbf{f}_{\varepsilon} \in \mathcal{H}$ such that $\mathcal{R}_{\mathcal{L}}(\mathbf{f}_{\varepsilon}) \leq \mathcal{R}_{\mathcal{L}}\left(\mathbf{f}^{\lambda}\right) + \frac{\varepsilon}{2}$. $\left\|\mathbf{f}_{\varepsilon}\right\| \leq M_n$ when n is sufficiently large. Due to condition **AS.4**, $\lambda_n \to 0$, making $M_n \to \infty$. Further notice that we introduce $\mathbf{f}_{\varepsilon}$ since it is independent of n. As a result, its norm, even though is bounded by M_n, is a constant and is not changing with respect to n. By Rademacher complexity, the following inequality holds with probability at least $1 - \delta$, where $0 < \delta < 1$:

$$\mathcal{R}_{\mathcal{L}}\left(\mathbf{f}_{\mathbf{n}}^{\lambda}\right) \leq \mathcal{R}_{\mathcal{L},T_n}\left(\mathbf{f}_{\mathbf{n}}^{\lambda}\right) + \frac{2C(K_{\max})M_n}{\sqrt{n}} + (L_0 + C(K_{\max})M_n)\sqrt{\frac{\log 2/\delta}{2n}},$$

$\mathbf{f}_{\varepsilon}$ is not the optimal in training data

$$\leq \mathcal{R}_{\mathcal{L},T_n}(\mathbf{f}_{\varepsilon}) + \lambda_n \|\mathbf{f}_{\varepsilon}\|^2 - \lambda_n \left\|\mathbf{f}_{\mathbf{n}}^{\lambda}\right\|^2 + \frac{2C(K_{\max})M_n}{\sqrt{n}} + (L_0 + C(K_{\max})M_n)\sqrt{\frac{\log 2/\delta}{2n}},$$

$$\text{Drop}\left(\lambda_n \left\|\mathbf{f}_{\mathbf{n}}^{\lambda}\right\|^2 > 0\right) \leq \mathcal{R}_{\mathcal{L},T_n}(\mathbf{f}_{\varepsilon}) + \lambda_n \|\mathbf{f}_{\varepsilon}\|^2 + \frac{2C(K_{\max})M_n}{\sqrt{n}} + (L_0 + C(K_{\max})M_n)\sqrt{\frac{\log 2/\delta}{2n}},$$

Rademacher Complexity again

$$\leq \mathcal{R}_{\mathcal{L}}(\mathbf{f}_{\varepsilon}) + \lambda_n \|\mathbf{f}_{\varepsilon}\|^2 + \frac{4C(K_{\max})M_n}{\sqrt{n}} + 2(L_0 + C(K_{\max})M_n)\sqrt{\frac{\log 2/\delta}{2n}}.$$

Let $\delta = \frac{1}{n^2}$, and N large such that for all $n > N$,

$$\lambda_n \|\mathbf{f}_{\varepsilon}\|^2 + \frac{4C(K_{\max})M_n}{\sqrt{n}} + 2(L_0 + C(K_{\max})M_n)\sqrt{\frac{\log 2/\delta}{2n}} \leq \frac{\varepsilon}{2}.$$

The inequality exists because $\left\|\mathbf{f}_{\varepsilon}\right\|$ is a constant with respect to n, and all other terms are converging to zero. Thus,

$$\mathcal{R}_{\mathcal{L}}\left(\mathbf{f}_{\mathbf{n}}^{\lambda}\right) \leq \mathcal{R}_{\mathcal{L}}(\mathbf{f}_{\varepsilon}) + \frac{\varepsilon}{2} \leq \mathcal{R}_{\mathcal{L}}\left(\mathbf{f}^{\lambda}\right) + \varepsilon$$

with probability $1 - \frac{1}{n^2}$. We conclude that

$$\mathbb{P}\left(\mathcal{R}_{\mathcal{L}}\left(\mathbf{f}_{\mathbf{n}}^{\lambda}\right) - \mathcal{R}_{\mathcal{L}}\left(\mathbf{f}^{\lambda}\right) | \geq \varepsilon\right) \to 0 \tag{A3}$$

for any arbitrary ε. This establishes the weak consistency of CP-STM. For strong consistency, we consider for each n

$$\sum_{n=1}^{\infty} \mathbb{P}\left(\mathcal{R}_{\mathcal{L}}\left(\mathbf{f}_{\mathbf{n}}^{\lambda}\right) - \mathcal{R}_{\mathcal{L}}\left(\mathbf{f}^{\lambda}\right) | \geq \varepsilon\right) \leq N - 1 + \sum_{n=1}^{\infty} \frac{1}{n^2} \leq \infty.$$

By Borel–Cantelli lemma [22], $\mathcal{R}_{\mathcal{L}}\left(\mathbf{f}_{\mathbf{n}}^{\lambda}\right) \to \mathcal{R}_{\mathcal{L}}\left(\mathbf{f}^{\lambda}\right)$ almost surely. The proof is finished.

APPENDIX B. DATA PRE-PROCESSING FOR SECTION 7

We provide further details about our EEG-fMRI data preprocessing and fMRI data extraction in this section. Most of the processing steps are referred from [34]. Information about the number of trials per subject can be seen in Table B1.

B.1.fMRI Data

The fMRI data processing includes three major steps, which are preprocessing, ROIs identification, and data extraction. We describe all these steps here. All the steps are performed by SPM 12 in Matlab. There are five steps in the image preprocessing part including realignment, co-registration, segment, normalization, and smoothing.

- *Realignment*: It is a procedure to align all the 3D BOLD volumes recorded along the time to remove artifacts caused by head motions, and also to estimate head position. For each task, there are three sessions of fMRI scans, providing 510 scans in total for each subject.

TABLE B1 EEG-fMRI data: number of trials per subject

Tasks	Auditory oddball	Auditory standard	Visual oddball	Visual standard
Subject 1	75	299	75	299
Subject 2	70	287	70	287
Subject 3	74	296	74	296
Subject 5	74	299	74	299
Subject 6	75	290	75	290
Subject 7	73	295	73	295
Subject 8	72	297	72	297
Subject 9	75	297	75	298
Subject 10	72	298	72	298
Subject 11	70	293	70	293
Subject 12	74	299	74	299
Subject 13	71	297	71	297
Subject 14	75	296	75	296
Subject 15	72	295	72	295
Subject 16	74	293	74	293
Subject 17	73	295	73	295

These scans are realigned within subject to the average of these 510 scans. (average across time) In SPM, we create three independent sessions to load all the fMRI runs, and choose not to reslice all the images at this step. The reslicing will be done in normalization step. Avoiding extra reslicing can avoid introducing new artifacts. The mean scan is created in this step for co-registration.

- *Co-registration*: Since all the fMRI scans are aligned to the mean scan, we have to transform the T1-weighted anatomical scan to match their orientation. Reason for doing this is that all the data will finally be transformed to a standardized space. Estimating such a transformation with T1-weighted scan can provide a high accuracy, since anatomical scans have higher resolutions. Matching the orientation of T1-weighted scan with all the fMRI scans makes it possible to apply the transformation estimated from T1 scan directly on fMRI data. In this step, we let the mean fMRI scan to be stationary, and move T1 anatomical scan to match it. A resliced T1 weighted scan is created in this step.
- *Segment*: This step estimate a deformation transformation mapping data into MNI 152 template space [15, 42]. A forward deformation field is created in this step.
- *Normalization*: In this step, the forward deformation is applied to all realigned fMRI scans, transforming all the data into MNI template space. The voxel size is set to be $3 \times 3 \times 4$ mm, which is the same as the original images.
- *Smoothing*: All normalized fMRI volumes are then smoothed by 3D Gaussian kernels with FWHM parameter being $8 \times 8 \times 8$.

This preprocessing procedure is applied to auditory and visual fMRI scans separately and independently.

For each task, the processed fMRI are used to for statistical analysis introduced in [50, 75]. These models are basic linear mixed effect model with auto-regression covariance structure. Since these models are standard and are out of the scope of this dissertation, we do not introduce them in this part. For the first-level (subject-level) analysis, we use the model to estimate two contrast images: standard stimulus over baseline and oddball stimulus over baseline. These two are difference of average BOLD signals during stimulus time and that during no stimulus (baseline) time. They can be understand as the estimate $\widehat{\beta}$ in a regression model $y = x\beta + \varepsilon$. These contrasts are then pooled together in the group-level analysis. For each voxel, the group-level analysis performs a T-test to compare the BOLD signals in standard contrasts and oddball contrasts. For voxels whose test results are significant, SPM highlighted them as the ROI. The ROIs of auditory and visual tasks are presented in Figures B1 and B2 with p-values.

The coordinates of these activate voxels are also provided in the statistical analysis results. To extract ROI data, we can use "spm_get_data" function in SPM 12. Since we are classifying trials, we only take one fMRI scan for each trial. This is because the trial duration (0.6 s) is less than the repetition time (2 s) of fMRI data. For each trial, we take the k-th fMRI scan where "$k = \text{round}(\text{onset}/\text{TR}) + 1$". This option is also inspired by SPM codes.

B.2.EEG data

The preprocessing of EEG data is relatively easy comparing to fMRI, since all EEG are already converted to MAT file and are re-referenced. Thus, we only need to resample it using Matlab signal processing toolbox to a lower sampling rates, which is 200 Hz in our case. Then, we use function "ft_preproc_lowpassfilter" and "ft_preproc_highpassfilter" from SPM 12 toolbox to filter the data. Finally, we split EEG into epochs which starts 100 ms before the onset and ends 500 ms after the onset. According to Henson et al. [34], such a duration is long enough to capture the event-related potential for EEG data.

APPENDIX C. PARAMETER SELECTION

C.1 Multimodal tensor factorization

The proposed model requires the selection of three different parameters, namely, γ, β, and rank r. To select these parameters, we closely follow best practices outlined in

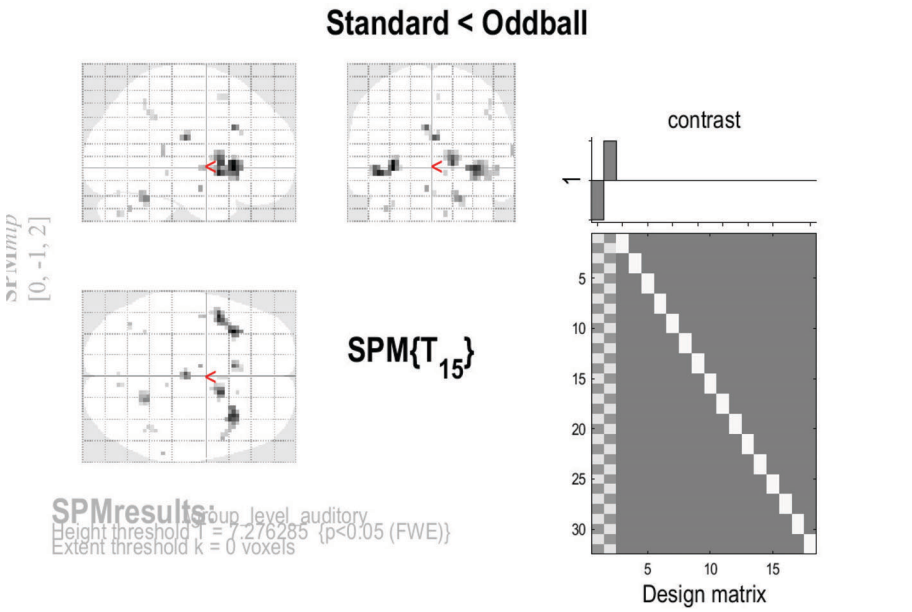


FIGURE B1 Auditory functional magnetic resonance imaging (fMRI) group-level analysis

Statistics: ***p-values adjusted for search volume***

set-level		cluster-level				peak-level					mm	mm	mm
p	c	$p_{FWE\text{-}corr}$	$q_{FDR\text{-}corr}$	k_E	p_{uncorr}	$p_{FWE\text{-}corr}$	$q_{FDR\text{-}corr}$	T	(Z_E)	p_{uncorr}			
0.00018		0.000	0.000	47	0.000	0.000	0.191	11.77	5.83	0.000	-33	20	-2
						0.000	0.191	11.00	5.67	0.000	-45	8	-6
		0.000	0.000	47	0.000	0.000	0.191	10.62	5.59	0.000	33	20	-2
						0.010	0.765	8.29	5.01	0.000	51	11	-2
						0.017	0.782	7.95	4.91	0.000	45	17	-10
		0.000	0.000	28	0.000	0.001	0.196	10.30	5.52	0.000	15	8	2
		0.000	0.002	9	0.001	0.001	0.230	9.93	5.43	0.000	0	-19	22
		0.000	0.003	8	0.001	0.001	0.249	9.68	5.37	0.000	-6	20	30
		0.000	0.000	17	0.000	0.003	0.421	9.04	5.21	0.000	21	-52	-26
		0.015	0.179	1	0.179	0.008	0.677	8.50	5.07	0.000	-6	-7	-18
		0.005	0.090	2	0.065	0.013	0.767	8.11	4.95	0.000	63	-37	10
		0.005	0.090	2	0.065	0.013	0.767	8.11	4.95	0.000	-45	-4	-2
		0.005	0.090	2	0.065	0.016	0.782	7.98	4.92	0.000	-39	-64	-34
		0.005	0.090	2	0.065	0.022	0.812	7.80	4.86	0.000	9	-73	-38
		0.015	0.179	1	0.179	0.028	0.812	7.65	4.81	0.000	-33	-52	-30
		0.001	0.034	4	0.013	0.028	0.812	7.64	4.81	0.000	45	-43	38
		0.015	0.179	1	0.179	0.028	0.812	7.63	4.81	0.000	9	-34	-6
		0.005	0.090	2	0.065	0.032	0.812	7.56	4.78	0.000	45	-52	50
		0.015	0.179	1	0.179	0.034	0.825	7.51	4.77	0.000	15	23	-2
		0.015	0.179	1	0.179	0.042	0.887	7.39	4.73	0.000	0	-73	-22
		0.002	0.063	3	0.028	0.043	0.887	7.37	4.72	0.000	3	14	54

table shows 3 local maxima more than 8.0mm apart

Height threshold: T = 7.28, p = 0.000 (0.050)
Extent threshold: k = 0 voxels
Expected voxels per cluster, <k> = 0.588
Expected number of clusters, <c> = 0.08
FWEp: 7.276, FDRp: Inf, FWEc: 1, FDRc: 4

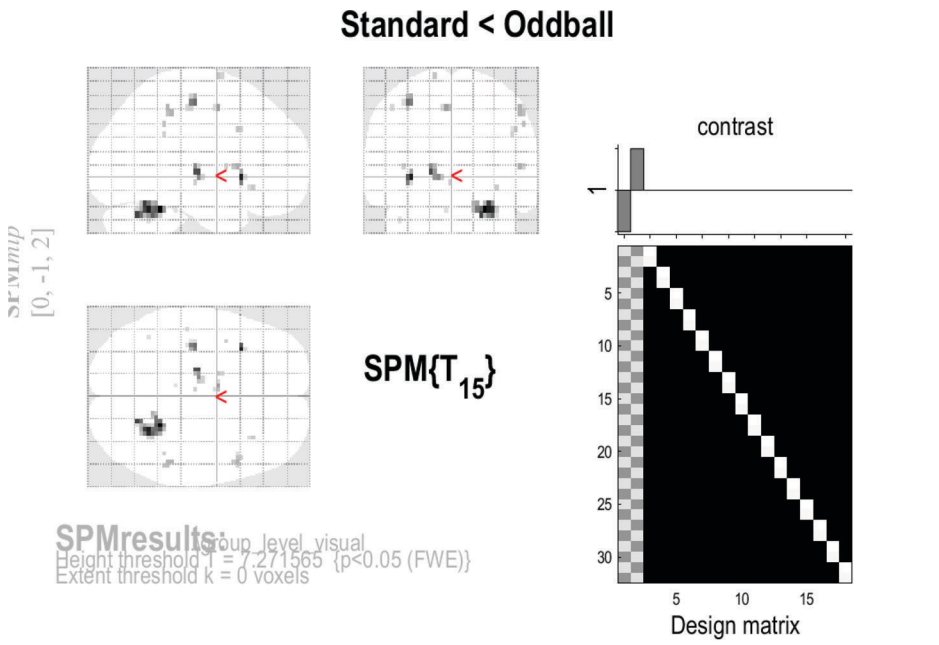
Degrees of freedom = [1.0, 15.0]
FWHM = 13.4 13.4 12.6 mm mm mm; 4.5 4.5 3.1 {voxels}
Volume: 1327176 = 36866 voxels = 515.1 resels
Voxel size: 3.0 3.0 4.0 mm mm mm; (resel = 62.78 voxels)

previous work on CMTF [2], ACMTF [5], and CCMTF [56]. First of all, one of these parameter can be set to 1 as a pivot, and following previous work, we set $\gamma = 1$. The selection of rank r is directly related to the selection of β. As β enforces sparsity over the singular values, it directly minimizes the rank. With sufficiently large r, we can estimate the low-rank part through optimization. For the selection of r in real data, we set $r = 5$ following the work of Mosayebi and Hossein-Zadeh [56], where it was shown through CORCONDIA tests [35] that $r = 3$ is sufficiently large for oddball data. In the case of the simulation study, $r = 5$ is again sufficiently large as the data were generated from rank $r = 3$ factors. Finally, based on our empirical results and the results presented in [56] we set $\beta = 0.001$ using k-fold cross-validation.

C.2 Coupled support tensor machine

The parameters in C-STM include kernel weights w_1, w_2, w_3 and regularization parameter λ in the optimization. The weight parameters, normalized such that the ℓ_2-norm is equal to 1, and λ are selected using fivefold cross-validation. The overall classification accuracy in our validation set serves as the performance metric and helps us determine the best combination of weights and λ.

The selection of weight parameters $w1, w2, w3$ is indeed a problem of how to combine kernels from different modalities. It is straightforward to calculate kernels from every data modality, however, combining them appropriately and effectively would be challenging unless we can find out the weight for each kernel. This problem has been widely studied in the literature of MKL. In



Statistics: ***p-values adjusted for search volume***

set-level		cluster-level				peak-level					mm mm mm		
p	c	$p_{FWE\text{-}corr}$	$q_{FDR\text{-}corr}$	k_E	p_{uncorr}	$p_{FWE\text{-}corr}$	$q_{FDR\text{-}corr}$	T	(Z_E)	p_{uncorr}			
0.00017		0.000	0.000	56	0.000	0.001	0.429	10.15	5.49	0.000	24	-46	-26
						0.001	0.429	9.69	5.38	0.000	27	-55	-26
						0.003	0.481	9.20	5.26	0.000	21	-64	-30
		0.000	0.010	6	0.002	0.001	0.429	9.68	5.38	0.000	-33	17	-2
		0.000	0.000	12	0.000	0.003	0.481	9.10	5.23	0.000	-18	-16	2
		0.000	0.000	12	0.000	0.006	0.620	8.70	5.12	0.000	-36	-19	54
		0.001	0.028	4	0.010	0.013	0.910	8.11	4.95	0.000	-6	-1	50
		0.002	0.047	3	0.022	0.018	0.916	7.91	4.89	0.000	51	14	6
		0.000	0.016	5	0.005	0.019	0.916	7.87	4.88	0.000	51	-37	46
		0.002	0.047	3	0.022	0.027	0.916	7.65	4.81	0.000	48	14	34
		0.015	0.161	1	0.161	0.028	0.916	7.63	4.81	0.000	54	-43	38
		0.005	0.093	2	0.055	0.030	0.916	7.60	4.80	0.000	-15	-1	74
		0.005	0.093	2	0.055	0.032	0.916	7.55	4.78	0.000	33	26	-10
		0.015	0.161	1	0.161	0.034	0.916	7.51	4.77	0.000	-3	-19	-6
		0.015	0.161	1	0.161	0.036	0.916	7.48	4.76	0.000	-39	5	6
		0.015	0.161	1	0.161	0.036	0.916	7.48	4.76	0.000	45	-43	30
		0.015	0.161	1	0.161	0.042	0.971	7.38	4.73	0.000	-48	-25	50
		0.015	0.161	1	0.161	0.047	0.993	7.31	4.70	0.000	-6	-34	-26
		0.015	0.161	1	0.161	0.050	0.993	7.28	4.69	0.000	-42	-55	-34

table shows 3 local maxima more than 8.0mm apart

Height threshold: T = 7.27, p = 0.000 (0.050)
Extent threshold: k = 0 voxels
Expected voxels per cluster, <k> = 0.538
Expected number of clusters, <c> = 0.09
FWEp: 7.272, FDRp: Inf, FWEc: 1, FDRc: 3

Degrees of freedom = [1.0, 15.0]
FWHM = 13.0 13.0 12.1 mm mm mm; 4.3 4.3 3.0 {voxels}
Volume: 1316988 = 36583 voxels = 559.8 resels
Voxel size: 3.0 3.0 4.0 mm mm mm; (resel = 57.29 voxels)

FIGURE B2 Visual functional magnetic resonance imaging (fMRI) group-level analysis

[29], the authors summarize that the existing methods of kernel weight selection can be divided into five categories, including fixed rules, heuristic approaches, optimization approaches, Bayesian approaches, and boosting approaches. As there is no consensus on the best way to choose the weights, we adopt a cross-validation approach as explained in the Appendix of the revised manuscript to identify the kernel weights. The overall classification accuracy in our validation set serves as the performance metric and helps us determine the best combination of weights.

The generalization of our method to more than two modalities would be straightforward for tuning the weights. This is because the tuning problem has been widely studied in MKL research. There is no restriction on the number of kernels one can include in MKL framework. The weight selection techniques in MKL can be adapted to our framework.

The optimization problem defined in (10) is an ordinary SVM problem once the kernel values are calculated through Equation (9). Thus, for more information about the estimation procedure, λ selection as well as the consistency results readers are referred to the existing SVM literature [66].