BTTDA: Block-Term Tensor Discriminant Analysis for Brain-Computer Interfacing

A. Van Den Kerchove[1,2,*], H. Si-Mohammed[2], F. Cabestaing[2], M.M. Van Hulle[1]

[1] KU Leuven, Leuven Brain Institute, Leuven.AI,
Laboratory for Neuro- and Psychophysiology,
Campus Gasthuisberg O&N2, Herestraat 49 bus 1021, BE-3000 Leuven, Belgium

[2] Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

[*] arne.vandenkerchove@kuleuven.be

November 9, 2025

Abstract

Brain-computer interfaces (BCIs) allow direct communication between the brain and external devices, frequently using electroencephalography (EEG) to record neural activity. Dimensionality reduction and structured regularization are essential for effectively classifying task-related brain signals, including event-related potentials (ERPs) and motor imagery (MI) rhythms. Current tensor-based approaches, such as Tucker and PARAFAC decompositions, often lack the flexibility needed to fully capture the complexity of EEG data. This study introduces Block-Term Tensor Discriminant Analysis (BTTDA): a novel tensor-based and supervised feature extraction method designed to enhance classification accuracy by providing flexible multilinear dimensionality reduction. Extending Higher Order Discriminant Analysis (HODA), BTTDA uses a novel and interpretable forward model for HODA combined with a deflation scheme to iteratively extract discriminant block terms, improving feature representation for classification. BTTDA and a sum-of-rank-1-terms variant PARAFACDA were evaluated on publicly available ERP (second-order tensors) and MI (third-order tensors) EEG datasets from the MOABB benchmarking framework. Benchmarking revealed that BTTDA and PARAFACDA significantly outperform the traditional HODA method in ERP decoding, resulting in state-of-the art performance (ROC-AUC=91.25%). For MI, decoding results of HODA, BTTDA and PARAFACDA were subpar, but BTTDA still significantly outperformed HODA (64.52%>61.00%). The block-term structure of BTTDA enables interpretable and more efficient dimensionality reduction without compromising discriminative power. This offers a promising and adaptable approach for feature extraction in BCI and broader neuroimaging applications.

Keywords *tensor discriminant analysis, brain-computer interface, block-term decomposition, multilinear decoding, event-related potentials, motor imagery*

1 Introduction

Brain-computer interfaces (BCIs) have the potential to bypass defective neural pathways by providing an alternative communication channel between the brain and an external device. These interfaces find

applications in the development of neuroprosthetics, assistive technologies and rehabilitation [1]. To achieve their functionality, BCIs record and process neural data, with electroencephalography (EEG) the most popular recording method in the field.

A BCI usually operates by identifying specific, task-related activity in the recorded EEG data, which can then be coupled to output or actions. This methodology often gives rise to classification problems [2]. Some well-known examples include the P300 speller [3], where momentary visual stimuli evoke characteristic event-related potentials (ERPs) modulated by attention, and motor imagery (MI) [4], where different (imagined) limb movements evoke event-related synchronizations/desynchronizations (ERS/Ds) with different spatial patterns. As a consequence BCI decoding (ERP vs. non-attended ERP, left vs. right limb MI, ...) involves a calibration phase training a classifier on labeled EEG data and an operation phase where the trained classifier is applied to unseen EEG data.

The duration of the calibration session should ideally be minimized to enhance user experience. This results in small, subject- and session-specific training datasets which make BCI classification methods vulnerable to overfitting in the presence of high-dimensional data. One possible countermeasure is applying a dimensionality reduction technique which extracts a lower-dimensional set features relevant to the classification problem at hand.

1.1 Tensors & tensor methods

Because of the multichannel time series format of EEG and other BCI functional neuroimaging methods, recorded data naturally exist as multiway data, capturing information in both the spatial and the temporal domain. Preprocessing transformations can further expand the data into additional analytic domains. Common examples include time-frequency transformation, time-binning, or integrating information across multiple subjects or conditions. This in turn results in high-dimensional datasets which are usually flattened into a set of sample vectors, stripping the original data of its structure. A more suited approach relies on this intrinsic multiway structure of neural data [5] to represent the data as *tensors*, multiway arrays, with each domain corresponding to a tensor *mode*. Tensors provide a structured data representation for this highly dimensional multiway data. This in turn paves the way to the development of tensor methods which can counteract some of the drawbacks of the dimensionality problem. Tensor methods are machine learning or dimensionality reduction techniques that consider each tensor mode separately, reducing a given problem into partial, per-mode problems.

Tensor methods often decompose tensors into a lower dimensional structure of a core tensor and factor tensors. The most common approaches adhere to either the Tucker structure or the PARAFAC structure. A Tucker decomposition reduces an input tensor of order K with dimensions $(D_1, D_2, \ldots, D_K)$ to a dense tensor with dimensions $(R_1, R_2, \ldots, R_K)$, with $R_k \leq D_k$ for $k = 1, 2, \ldots K$, using a set of per-mode factor matrices. Effective unsupervised tensor decomposition and approximation in the Tucker format can be achieved using the Higher Order Singular Value Decomposition (HOSVD) [6, 7]. Alternatively, the Parallel Factor Analysis (PARAFAC) structure can be used. Here, the tensor is decomposed into a sum of rank-1 tensors, each the product of a scalar and a vector per mode. This is equivalent to a Tucker structured decomposition with all core elements off the hyperdiagonal set to 0, as shown in fig. 1. One way of obtaining an unsupervised PARAFAC decomposition is through the Canonical Polyadic Decomposition [8, 9]. These decomposition methods can be regarded as feature extraction methods for a BCI classification problem, with the flattened core tensors as feature vectors. Extracted features can subsequently be classified to predict class labels, most commonly using linear discriminant analysis (LDA) or a support vector machine (SVM).

While commonly used, these Tucker or PARAFAC structures might still not be able to efficiently represent relevant neural information in a compressed format. The block-term tensor structure is a generalization of both the Tucker and PARAFAC structures. It represents the tensor as a sum of Tucker structured terms. If the number of terms is equal to 1, it is equivalent to the Tucker structure; if the

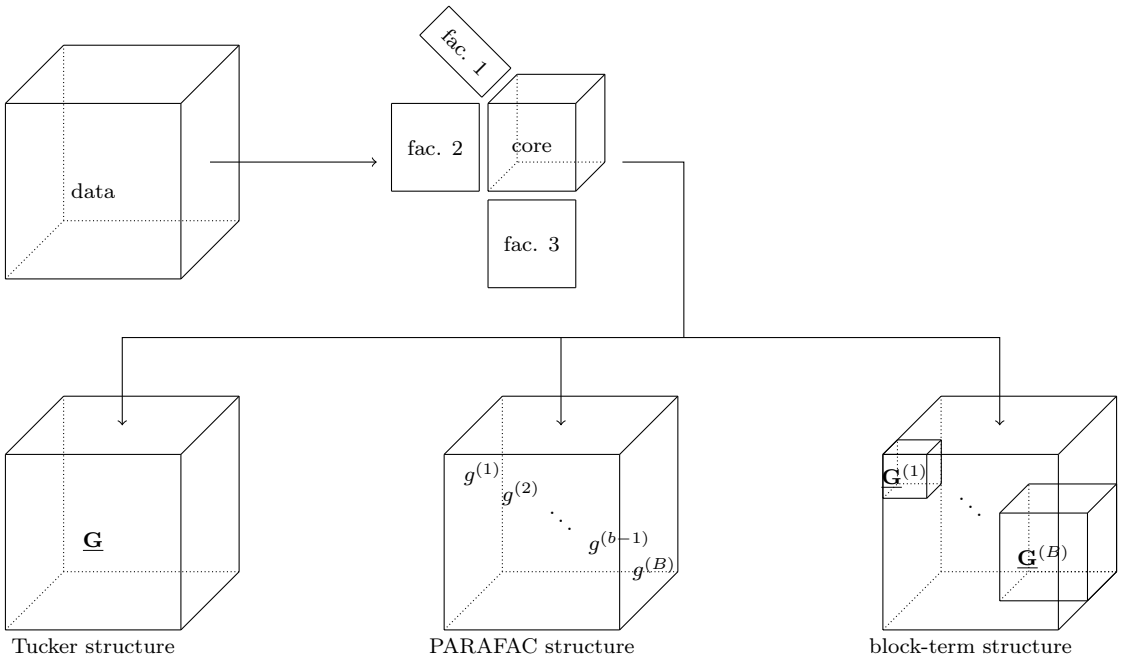

Figure 1: A tensor decomposition finds core tensor and factor matrices from input tensor. This core tensor can have several structures. In the Tucker structure, the core is a dense tensor $\underline{\mathbf{G}}$. The PARAFAC structure expresses the core as a sum of B rank-1 terms, each with a scalar core $g^{(b)}$. The block-term structure expresses the core as a sum of B smaller, Tucker-structured blocks $\underline{\mathbf{G}}^{(b)}$. Both the PARAFAC and block-term structures are more sparse than the full Tucker structure, yet the block-term structure is more flexible as it allows blocks of variable tensor dimensionality instead of fixed rank-1 terms.

dimensions of each term are equal to 1, it is equivalent to the PARAFAC structure. The block-term structure (fig. 1, right) is more flexible than either the Tucker or the PARAFAC structures, since it is not constrained to solutions that must be expressed as either one of these structures and their chosen hyperparameters. Due to its flexibility, the block-term structure can strike a better balance between extracting a maximal amount of relevant features and a minimal amount of irrelevant features. However, this increased flexibility comes at the cost of a higher number of hyperparameters, as now both the number of terms and the dimension of each term need to be specified. A block-term structured core tensor can be obtained in an unsupervised way using the Block-Term Decomposition (BTD) [10, 11, 12, 13]. Performance of methods leveraging either the Tucker and PARAFAC structures are heavily dependent on the prior choice of hyperparameters describing the desired reduced dimension or the number of rank-1 terms.

1.2 Supervised tensor decompositions for BCI

If the decompositions are not full rank, the Tucker, PARAFAC and block-term structures are not unique and can be obtained by optimizing different criteria. Given the low signal-to-noise ratio and specific, task-related output expected in a BCI application, supervised feature extraction and machine learning techniques are favored [2] over the unsupervised decomposition methods presented above. A decomposition that is helpful for classification should ideally optimize the discriminability between classes in the resulting core tensors, which can then be considered as extracted features. In this philosophy, the Tucker decomposition can also be obtained using Higher

Order Discriminant Analysis (HODA) [14, 15, 16], which optimizes class separability in the Fisher sense, analogous to linear discriminant analysis.

Variants of HODA have been applied to BCI problems such as the decoding of ERPs [17, 18] and MI [19, 20] with positive results [2]. Recent work proposes optimization of the objective function and introduces regularization [21, 22, 23]. Discriminant tensor features have also been extracted in the PARAFAC structure through manifold optimization [16]. However, it is not immediately obvious if either the Tucker or PARAFAC structure are most suited to represent the neural data of interest for the BCI paradigm and for decoding.

More recent studies have shown that supervised decoders adopting a more flexible structure can improve BCI performance. Promising results have been achieved for regression tasks using Higher Order Partial Least Squares (HOPLS) [24, 25] and Block-Term Tensor Regression (BTTR) [26, 27]. BTTR has also been adapted into the classification variant Block-Term Tensor Classification (BTTC) [28] but this methodology leaves room for improvement: instead of optimizing features directly for class separability, BTTC regresses to dummy 2-valued independent variable. Thus, the method cannot be extended to a multi-class setting. Furthermore, structures employed in these regression approaches could still be considered as relatively constrained in comparison. A more flexible approach could rely on a full block-term tensor decomposition of the input data which optimizes discriminability and relies on a low-rank common subspace between the input and classification labels. Huang et al. [29] propose a supervised approach for finding multiple discriminant multilinear spectral filter terms and apply it to motor imagery BCI, but their decomposition is also limited in flexibility, since the solution is restricted to terms with dimension $(R_1, R_2, 1)$, with mode 3 corresponding to the frequency domain.

1.3 Contribution: A block-term structured model for classification

With a proper choice of reduced dimension and number of terms, a block-term decomposition directly optimizing discriminability might be more suited to represent complex neural data in a sparse way, which additionally yields a regularization effect. Multiple parsimonious discriminant block terms with lower dimensions might yield better performance than a single HODA block requiring a higher dimension to capture discriminant information, and by doing so extracts too many irrelevant features. A complementary view on the same approach goes as follows: if HODA with a well-chosen reduced dimension extracts some discriminant features from the input tensor, it is likely that it does not retrieve all useful information due to the restrictions imposed by its Tucker structure. Could HODA therefore not sequentially be applied to extract discriminant Tucker structured terms – potentially with lower dimension – as long as decoding performance increases?

We implement this idea as a novel supervised feature extraction method titled Block-Term Tensor Discriminant Analysis (BTTDA), a generalization of the aforementioned HODA algorithm. BTTDA extracts discriminant features while adhering to a flexible and efficient block-term tensor structure. This work features the following contributions: 1) We develop a forward model for HODA to reconstruct a given input tensor from the extracted features. 2) This allows us to introduce BTTDA as a state-of-the-art BCI feature extraction method based on the block-term tensor structure. 3) We evaluate a BCI decoder based on BTTDA and its special PARAFAC-structured case on decoding benchmarks for both ERP and MI BCI paradigms and compare these to state-of-the-art decoders.

2 Methods

2.1 Notation

Tensors are indicated by bold underlined letters $\underline{\mathbf{X}}$, matrices by bold letters $\mathbf{U}$, fixed scalars by uppercase letters K, and variable scalars as lowercase letters k. The n^{th} sample of a tensor dataset with N samples is written as $\underline{\mathbf{X}}(n)$, the dataset itself as $\{\underline{\mathbf{X}}(n)\}_n^N$. A tensor $\underline{\mathbf{X}} \in \mathbb{R}^{D_1 \times D_2 \times \cdots \times D_K}$ can be unfolded in mode k to a matrix $\mathbf{X}_k \in \mathbb{R}^{(D_k \times \prod_{j \neq k}^{K} D_j)}$, by concatenating

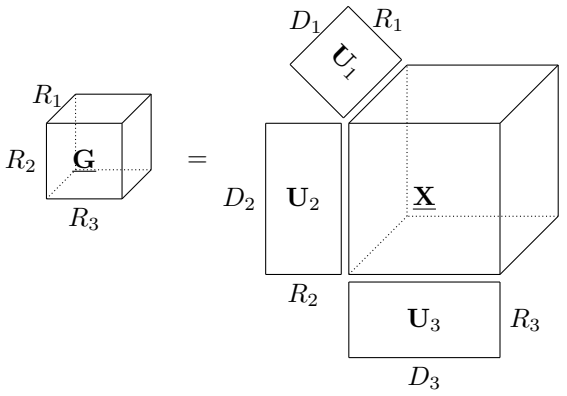

Figure 2: A visualization of the multilinear projection obtained by Higher Order Discriminant Analysis (HODA) applied to a third-order tensor sample $\underline{\mathbf{X}}$ with dimensions (D_1, D_2, D_3). HODA finds projection matrices $\mathbf{U}_k$ such that maximal discriminability between classes is achieved in the projected latent tensors $\underline{\mathbf{G}}$ with reduced dimension (R_1, R_2, R_3).

all mode $j \neq k$ fibers. The tensor-matrix product of tensor $\underline{\mathbf{X}}$ with matrix $\mathbf{U}$ along a given mode k is written as $\underline{\mathbf{X}} \times_k \mathbf{U}_k$. For ease of notation, let $\underline{\mathbf{X}} \times \{\mathbf{U}\} = \underline{\mathbf{X}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_K \mathbf{U}_K$. When skipping one of the modes k, this is written as $\underline{\mathbf{X}} \times_{-k} \{\mathbf{U}\} = \underline{\mathbf{X}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_{k-1} \mathbf{U}_{k-1} \times_{k+1} \mathbf{U}_{k+1} \ldots \times_K \mathbf{U}_K$. $\mathbf{A} \otimes \mathbf{B}$ indicates the Kronecker product of matrices $\mathbf{A}$ and $\mathbf{B}$.

2.2 Higher Order Discriminant Analysis

Higher Order Discriminant Analysis (HODA) [15] is a supervised, tensor-based dimensionality reduction and feature extraction technique. For a set of N tensors of order K $\left\{\underline{\mathbf{X}}(n) \in \mathbb{R}^{D_1 \times D_2 \times \cdots \times D_K}\right\}_n^N$, HODA finds projection matrices $\mathbf{U_k}$ for each mode k which project a given $\underline{\mathbf{X}}$ to a latent tensor $\underline{\mathbf{G}} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_K}$, usually with lower dimensions $(R_1 \leq D_1, R_2 \leq D_2, \ldots, R_K \leq D_K)$ using tensor-matrix mode products:

$$\underline{\mathbf{G}} = \underline{\mathbf{X}} \times \{\mathbf{U}\} \tag{1}$$

as visualized in fig. 2. Since HODA extracts latent features or properties $\underline{\mathbf{G}}$ from the observed data $\underline{\mathbf{X}}$, relying on a task-related criterion, it can be referred to as a *backward model*.

Analogous to the HOSVD, HODA decomposition results in a dense latent tensor $\underline{\mathbf{G}}$ and imposes an orthogonality constraint on each $\mathbf{U}_k$ to ensure uniqueness. However, while HOSVD projection matrices minimize the reconstruction error, HODA optimizes the class discriminability of the reduced tensors $\underline{\mathbf{G}}(n)$ belonging to classes with labels c_n. This is a desirable property in a classification setting where samples are high-dimensional tensors.

HODA optimizes discriminability in the Fisher sense, maximizing the Fisher ratio ϕ between the latent tensors $\underline{\mathbf{G}}(n)$:

$$\phi\left(\{\mathbf{U}\}\right) = \frac{\sum_c^C N_c \left\| \bar{\underline{\mathbf{G}}}(c) - \bar{\bar{\underline{\mathbf{G}}}} \right\|_F^2}{\sum_n^N \left\| \underline{\mathbf{G}}(n) - \bar{\underline{\mathbf{G}}}(c_n) \right\|_F^2} \tag{2}$$

for C classes with each N_c samples. $\bar{\underline{\mathbf{G}}}(c)$ is the mean of latent tensors of class c, and $\bar{\bar{\underline{\mathbf{G}}}}$ the mean of these class mean latent tensors. If the dimensions $(R_1, R_2, \ldots, R_k)$ are set a priori, the objective is now to find the optimal projection matrices:

$$\{\mathbf{U}^*\} = \underset{\{\mathbf{U}\}}{\operatorname{argmax}}\, \phi\left(\{\mathbf{U}\}\right) \tag{3}$$

which is solved through the backward HODA algorithm. To start, $\mathbf{U}_k$ are initialized to orthogonal matrices, e.g., as random orthonormal matrices, by a per-mode singular value decomposition (SVD), or as the partial HOSVD of all stacked tensors in the dataset. At each iteration, the algorithm loops through the modes and fixes all projections but $\mathbf{U}_k$ corresponding to mode k. It then finds a partial latent tensor:

$$\underline{\mathbf{G}}_{-k} = \underline{\mathbf{X}} \times_{-k} \{\mathbf{U}\} \tag{4}$$

Subsequently, a new projection matrix $\mathbf{V}_k$ can be found analogous to Linear Discriminant Analysis by constructing the partial within-class scatter matrix:

$$\mathbf{S}_{-k,\mathrm{w}} = \sum_n^N \tilde{\mathbf{G}}_{-k,k}(n) \cdot \tilde{\mathbf{G}}_{-k,k}^{\intercal}(n) \tag{5}$$

with $\tilde{\underline{\mathbf{G}}}_{-k}(n) = \underline{\mathbf{G}}_{-k}(n) - \bar{\underline{\mathbf{G}}}_{-k}(c_n)$, and the partial between-class scatter matrix:

$$\mathbf{S}_{-k,\mathrm{b}} = \sum_{c}^{C} N_c \tilde{\bar{\mathbf{G}}}_{-k,k}(c) \cdot \tilde{\bar{\mathbf{G}}}^{\intercal}_{-k,k}(c) \tag{6}$$

with $\tilde{\bar{\underline{\mathbf{G}}}}_{-k}(c) = \bar{\underline{\mathbf{G}}}_{-k}(c) - \bar{\underline{\mathbf{G}}}_{-k}$, and solving for the R_k leading eigenvectors in the eigenvalue problem:

$$\mathbf{S}_{-k,\mathrm{b}} - \varphi_k \mathbf{S}_{-k,\mathrm{w}} = \mathbf{V}_k \mathbf{\Lambda} \mathbf{V}_k^{\intercal} \tag{7}$$

with $\varphi_k = \mathrm{Tr}\left(\mathbf{U}_k^{\intercal} \mathbf{S}_{-k,\mathrm{b}} \mathbf{U}_k\right) / \mathrm{Tr}\left(\mathbf{U}_k^{\intercal} \mathbf{S}_{-k,\mathrm{w}} \mathbf{U}_k\right)$ using the $\mathbf{U}_k$ obtained in the previous iteration. Finally, the orthogonal transformation invariant projections $\mathbf{U}_k$ are obtained by calculating the per-mode total scatter matrices:

$$\mathbf{S}_{k,\mathrm{t}} = \sum_{n}^{N} \mathbf{X}_k(n) \cdot \mathbf{X}_k^{\intercal}(n) \tag{8}$$

and finding the R_k leading eigenvectors of:

$$\mathbf{V}_k \mathbf{V}_k^{\intercal} \mathbf{S}_{k,\mathrm{t}} \mathbf{V}_k \mathbf{V}_k^{\intercal} = \mathbf{U}_k \mathbf{\Lambda} \mathbf{U}_k^{\intercal} \tag{9}$$

at each iteration [30]. The iterative process halts when the update of each $\mathbf{U}_k$ is lower than a predetermined threshold ϵ or after a fixed number of iterations $I_{\max}$. The full HODA procedure is summarized in algorithm 1.

To apply HODA in a classification setting, the projections are first learned on a training dataset with known class labels. Next, these projections are used to extract latent tensors from the tensors in the training dataset. These latent training tensors are then reshaped (*vectorized*) into feature vectors $\mathbf{g} = \mathrm{vec}(\underline{\mathbf{G}})$ and used to train a decision classifier with the corresponding class labels. At the evaluation stage, the projections learned from the training dataset are used to extract latent tensors from an unseen test dataset with unknown class labels, which can also be vectorized and passed on to the trained decision classifier.

To avoid overfitting and improve performance in low sample size settings, the HODA problem can be regularized by shrinking the partial within-class scatter matrices [15] with a shrinkage factor α_k at each step such that the eigenvalue problem becomes

$$\mathbf{S}_b^{(-k)} - \varphi \left[(1 - \alpha_k) \mathbf{S}_{-k,\mathrm{w}} + \alpha_k \mathbf{I} \right] = \mathbf{V}_k \mathbf{\Lambda} \mathbf{V}_k^{\intercal} \tag{10}$$

Algorithm 1 The HODA backward solution.

Require: $\{\underline{\mathbf{X}}(n)\}_n^N, \{c_n\}_n^N, (R_1, \ldots, R_K), I_{\max}, \epsilon$
1: $\mathbf{U}_k \leftarrow$ orthonormal matrix $\in \mathbb{R}^{D_k \times R_k}$ $\forall k$
2: $\mathbf{S}_{k,\mathrm{t}} \leftarrow \sum_n^N \mathbf{X}_k(n) \cdot \mathbf{X}_k^{\intercal}(n)$ $\forall k$
3: $i \leftarrow 1$
4: **repeat**
5: **for** $k = 1, 2 \ldots, K$ **do**
6: $\underline{\mathbf{G}}(n)_{-k} \leftarrow \underline{\mathbf{X}}(n) \times_{-k} \{\mathbf{U}\}$ $\forall n$
7: $\mathbf{S}_{-k,\mathrm{w}} \leftarrow \sum_n^N \tilde{\mathbf{G}}_{-k,k}(n) \cdot \tilde{\mathbf{G}}^{\intercal}_{-k,k}(n)$
8: $\mathbf{S}_{-k,\mathrm{b}} \leftarrow \sum_c^C N_c \tilde{\bar{\mathbf{G}}}_{-k,k}(c) \cdot \tilde{\bar{\mathbf{G}}}^{\intercal}_{-k,k}(c)$
9: $\varphi_k \leftarrow \mathrm{Tr}\left(\mathbf{U}_k^{\intercal} \mathbf{S}_{-k,\mathrm{b}} \mathbf{U}_k\right) / \mathrm{Tr}\left(\mathbf{U}_k^{\intercal} \mathbf{S}_{-k\mathrm{w}} \mathbf{U}_k\right)$
10: $\mathbf{V}_k \leftarrow$ largest magnitude R_k eigenvectors of $\mathbf{S}_{-k,\mathrm{b}} - \varphi_k \mathbf{S}_{-k,\mathrm{w}}$
11: $\mathbf{U}_k \leftarrow$ largest magnitude R_k eigenvectors of $\mathbf{V}_k \mathbf{V}_k^{\intercal} \mathbf{S}_{k,\mathrm{t}} \mathbf{V}_k \mathbf{V}_k^{\intercal}$
12: **end for**
13: $i \leftarrow i + 1$
14: **until** $i = I_{\max}$ or $||\mathbf{U}_k^{(i)} - \mathbf{U}_k^{(i-1)}|| < \epsilon$ $\forall k$

As in Linear Discriminant Analysis, the shrinkage parameter α_k can also be estimated in a data-driven way in HODA [22], e.g., using the Ledoit-Wolf procedure [31] at every iteration.

2.3 A forward model for HODA

As a prerequisite to the proposed BTTDA model, we must find a way to reconstruct the original data tensor $\underline{\mathbf{X}}$ as accurately as possible from $\underline{\mathbf{G}}$ after dimensionality reduction. This requires a *forward* model, a generative model that expresses the observed data in terms of given latent properties or features. As indicated earlier, finding the optimal projection matrices U that extract tensors $\underline{\mathbf{G}}$ given input data $\underline{\mathbf{X}}$ as in eq. (1) corresponds to fitting a backward HODA model. A forward model is a method to reconstruct the original data $\underline{\mathbf{X}}$ from the core tensor $\underline{\mathbf{G}}$. Forward models are useful for, e.g., interpretability and data compression, but here reconstruction with minimized reconstruction error is of interest.

A straightforward and computationally efficient candidate for the HODA forward model, visualized

in fig. 3, is given as:

$$\underline{\mathbf{X}} = \underline{\mathbf{G}} \times \{\mathbf{A}^\intercal\} + \underline{\mathbf{E}} = \hat{\underline{\mathbf{X}}} + \underline{\mathbf{E}} \tag{11}$$

with *activation patterns* $\mathbf{A}_k \in \mathbb{R}^{D_k \times R_k}$, reconstructed tensor $\hat{\underline{\mathbf{X}}}$, and error term $\underline{\mathbf{E}}$.

A good forward model should ensure that the norm of the reconstruction error $\|\underline{\mathbf{E}}\|_F$ is minimized. In other words, variation captured in the latent tensor should be maximally captured by the reconstruction term $\hat{\underline{\mathbf{X}}} = \underline{\mathbf{G}} \times \{\mathbf{A}^\intercal\}$, and not by the error term $\underline{\mathbf{E}}$ [32]. Hence, we aim to minimize the expected value of the cross-covariance between the noise term and the extracted latent tensors:

$$\{\mathbf{A}^*\} = \underset{\{\mathbf{A}\}}{\operatorname{argmin}} \, \mathrm{E}\left[\operatorname{vec}\left(\underline{\mathbf{E}}(n)\right) \operatorname{vec}\left(\underline{\mathbf{G}}(n)\right)\right]_n \tag{12}$$

or, equivalently [33, 32],

$$\{\mathbf{A}^*\} = \underset{\{\mathbf{A}\}}{\operatorname{argmin}} \sum_n^N \left[\underline{\mathbf{X}}(n) - \hat{\underline{\mathbf{X}}}(n)\right]^2 \tag{13}$$

$$= \underset{\{\mathbf{A}\}}{\operatorname{argmin}} \sum_n^N \left[\underline{\mathbf{X}}(n) - \underline{\mathbf{G}}(n) \times \{\mathbf{A}\}\right]^2 \tag{14}$$

This least-squares tensor approximation problem can be solved using the Alternating Least Squares algorithm [34], iteratively fixing all but one of the activation patterns such that:

$$\mathbf{A}_k = \underset{\mathbf{A}_k}{\operatorname{argmin}} \sum_n^N \left[\mathbf{X}_k(n) - \mathbf{A}_k \left(\underline{\mathbf{G}}(n) \times_{-k} \{\mathbf{A}\}\right)_k\right]^2 \tag{15}$$

at every iteration, which can be solved directly using ordinary least squares. The activation patterns are initialized to the weights $\{\mathbf{U}\}$ of the backward model. Similar to fitting the backward model, the iterative process for the forward model halts after a fixed number of iterations $I_{\max}$ or when the update of each $\mathbf{A}_k$ is lower than a predetermined threshold ϵ. The full procedure to determine the HODA forward projection is listed in algorithm 2.

Algorithm 2 The HODA forward solution.

Require: $\{\underline{\mathbf{G}}(n)\}_n^N, \{\underline{\mathbf{X}}(n)\}_n^N, I_{\max}, \epsilon$

$\mathbf{A}_k \leftarrow \mathbf{U}_k \ \forall k$
$i \leftarrow 1$
repeat
  for $k = 1, 2 \ldots, K$ **do**
    $\mathbf{X}_{-k}(n) \leftarrow \underline{\mathbf{G}}(n) \times_{-k} \{\mathbf{A}\} \ \forall n$
    $\mathbf{A}_k \leftarrow \operatorname{argmin}_{\mathbf{A}_k} \sum_n^N \left[\mathbf{X}_k(n) - \mathbf{A}_k \mathbf{X}_{-k}(n)\right]^2$
  end for
  $i \leftarrow i + 1$
until $i = I_{\max}$ or $||\mathbf{A}_k^{(i)} - \mathbf{A}_k^{(i-1)}|| < \epsilon \ \forall k$

2.4 Block-Term Tensor Discriminant Analysis

After defining the forward model, we can construct the proposed block-term tensor model. Assuming the latent tensors $\underline{\mathbf{G}}$ obtained by the backward projection of HODA do not achieve perfect class separation, the error term $\underline{\mathbf{E}}$ in eq. (11) contains some discriminative information. This, in turn, can be exploited to improve classifier performance. Useful features can then be extracted from $\underline{\mathbf{E}} = \underline{\mathbf{X}} - \hat{\underline{\mathbf{X}}}$ by further projecting it onto another core tensor $\underline{\mathbf{G}}^{(2)}$, assuming $\underline{\mathbf{G}}$ as $\underline{\mathbf{G}}^{(1)}$.

We thus extend the HODA feature extraction scheme to Block-Term Tensor Discriminant Analysis (BTTDA). BTTDA finds multiple discriminative blocks, such that its forward model adheres to the block-term tensor structure:

$$\underline{\mathbf{X}} = \sum_b^B \underline{\mathbf{G}}^{(b)} \times \left\{\mathbf{A}^{(b)}\right\} + \underline{\mathbf{E}} \tag{16}$$

for B extracted latent tensors $\underline{\mathbf{G}}^{(b)}$ and residual error term $\underline{\mathbf{E}}$. The BTTDA model is further illustrated by fig. 4. The block-term structure of this model implies that it is a generalization of both the Tucker-structured HODA and PARAFAC-structured discriminant feature extraction. If B in eq. (16) is set to one, BTTDA is equivalent to HODA; if at each term b the dimension of the core tensor are $(R_1^{(b)} = R_2^{(b)} = \ldots = R_k^{(b)} = 1)$, a PARAFAC structure is assumed and the resulting discriminant model is titled PARAFAC Discriminant Analysis (PARAFACDA).

Since BTTDA is specified above as a forward model, a backward procedure is required which finds the latent tensors $\underline{\mathbf{G}}^{(b)}$ given $\underline{\mathbf{X}}$ to BTTDA for feature

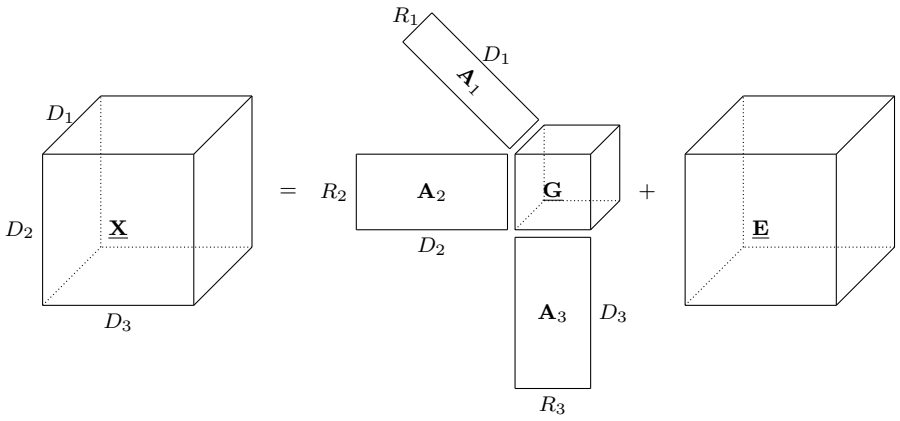

Figure 3: The forward projection for HODA. By calculating activation patterns $\mathbf{A}_k$, the original tensor $\underline{\mathbf{X}}$ can approximately be reconstructed from projected latent tensor $\underline{\mathbf{G}}$. The reconstruction is accurate up to an error term $\underline{\mathbf{E}}$. $\mathbf{A}_k$ are chosen such that the variability captured in the latent tensor is maximally explained by the reconstructed tensor $\hat{\underline{\mathbf{X}}}$ and not by the error term $\underline{\mathbf{E}}$.

extraction. The extracted features represented by the latent tensors $\underline{\mathbf{G}}^{(b)}$ can be computed through a deflation scheme summarized in algorithm 3. For

Algorithm 3 BTTDA feature extraction.

Require: $\{\underline{\mathbf{X}}(n)\}_n^N, \{c_n\}_n^N, \{(R_1^{(b)}, R_2^{(b)}, \ldots, R_K^{(b)})\}_b^B$
1: $\underline{\mathbf{E}}(n) \leftarrow \underline{\mathbf{X}}(n)\ \forall n$
2: **for** $b = 1, 2, \ldots, B$ **do**
3: $\{\mathbf{U}^{(b)}\} \leftarrow$ HODA on $\{\underline{\mathbf{E}}(n)\}_n^N$ and $\{c_n\}_n^N$ with rank $(R_1^{(b)}, R_2^{(b)}, \ldots, R_K^{(b)})$
4: $\underline{\mathbf{G}}^{(b)}(n) \leftarrow \underline{\mathbf{E}}(n) \times \{\mathbf{U}^{(b)}\}\ \forall n$
5: $\{\mathbf{A}^{(b)}\} \leftarrow$ Forward HODA on $\{\underline{\mathbf{G}}^{(b)}(n)\}_n^N$ and $\underline{\mathbf{E}}$
6: $\hat{\underline{\mathbf{E}}}(n) \leftarrow \underline{\mathbf{G}}^{(b)}(n) \times \{\mathbf{A}^{\intercal(b)}\}\ \forall n$
7: $\underline{\mathbf{E}}(n) \leftarrow \underline{\mathbf{E}}(n) - \hat{\underline{\mathbf{E}}}(n) \forall n$
8: **end for**

each block b, the latent tensor is extracted using the HODA backward projection from the residual error term of the previous block $\underline{\mathbf{E}}^{(b-1)}$ as in eq. (1):

$$\underline{\mathbf{G}}^{(b)} = \underline{\mathbf{E}}^{(b-1)} \times \left\{\mathbf{U}^{(b)}\right\} \tag{17}$$

This residual error term is calculated by finding the difference between the previous error and its reconstruction after backward and forward HODA projection:

$$\underline{\mathbf{E}}^{(b)} = \underline{\mathbf{E}}^{(b-1)} - \hat{\underline{\mathbf{E}}}^{(b-1)} \tag{18}$$

$$= \underline{\mathbf{E}}^{(b-1)} - \underline{\mathbf{G}}^{(b)} \times \left\{\mathbf{A}^{\intercal(b)}\right\} \tag{19}$$

with $\underline{\mathbf{E}}^{(0)} = \underline{\mathbf{X}}$.

The resulting latent tensors can be vectorized and concatenated into one single feature vector per input tensor:

$$\mathbf{g} = \left[\text{vec}\left(\underline{\mathbf{G}}^{(1)}\right)\ \text{vec}\left(\underline{\mathbf{G}}^{(2)}\right)\ \cdots\ \text{vec}\left(\underline{\mathbf{G}}^{(B)}\right)\right] \tag{20}$$

so that they can be classified in a similar manner to HODA.

2.5 Model and feature selection

Similar to the unsupervised BTD, the performance of BTTDA is heavily dependent on the number of blocks B and their corresponding dimensions $\{(R_1^{(b)}, R_2^{(b)}, \ldots, R_K^{(b)})\}_b^B$. If these are not known a priori or can not set based on insights into the data generation process, a model selection step is necessary in order to determine the optimal values for $R_k^{(b)}$ and B. These hyperparameters can be set through cross-validated hyperparameter tuning, although computationally expensive.

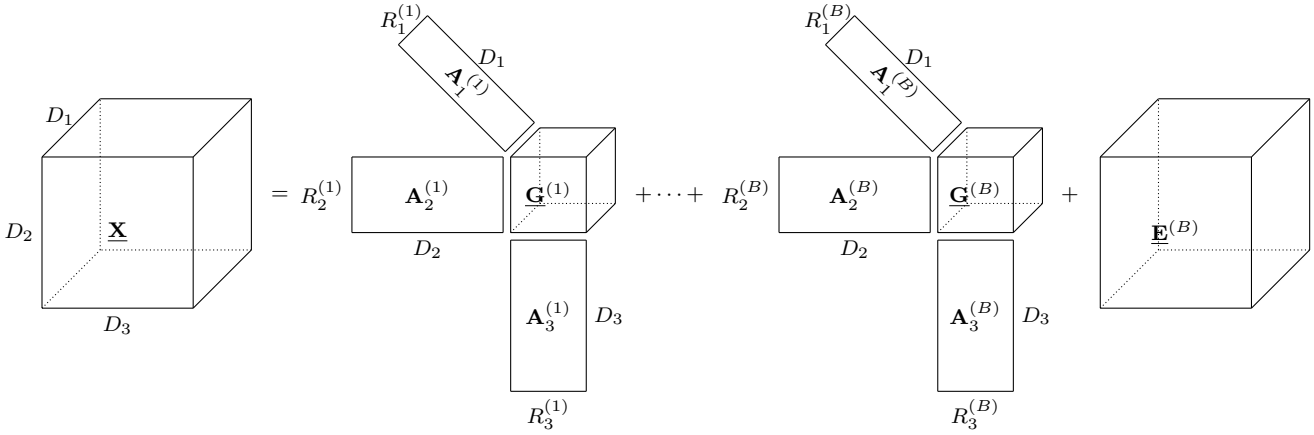

Figure 4: A forward model for Block-Term Tensor Discriminant Analysis (BTTDA). BTTDA can extract more features than HODA by iteratively finding a latent tensor $\underline{\mathbf{G}}^{(b)}$ in a deflation scheme. The HODA backward projection is first applied. Next, the input data is reconstructed via the HODA forward model and the difference between the two is found. Finally, this process is repeated with this difference as input data, until a desired number of blocks B has been found.

To reduce the hyperparameter search space, we introduce a single hyperparameter $\theta \in [0, 1]$ which replaces the block dimensions $\{(R_1^{(b)}, R_2^{(b)}, \ldots, R_K^{(b)})\}_b^B$. The new hyperparameter θ then controls the sparsity of the BTTDA solution, with $\theta = 0$ corresponding to the PARAFACDA model with blocks of dimension $(1, 1, \ldots, 1)$, and $\theta = 1$ corresponding to blocks of full rank $(D_1, D_2, \ldots, D_K)$. For $0 < \theta < 1$, the dimension of block b can be determined analogous to the method described by Phan and Cichocki [15]. Here, R_k are chosen based on the number of components needed to explain a certain proportion of the variability in a mode of the input data for a Tucker-structured decomposition. For the HODA model used in BTTDA, this can be achieved using the eigenvalues of the per-mode total scatter matrix of tensor $\underline{\mathbf{E}}^{(b-1)}$

$$\mathbf{S}_{k,\mathrm{t}}^{(b)} = \sum_n^N \mathbf{E}_k^{(b-1)}(n) \cdot \mathbf{E}_k^{(b-1)\intercal}(n) = \mathbf{W}_k^{(b)} \mathbf{\Lambda}_k^{(b)} \mathbf{W}_k^{(b)\intercal} \tag{21}$$

such that

$$R_k^{(b)} = \underset{R \in 1,\ldots,D_k}{\operatorname{argmin}} \frac{\sum_r^R \lambda_{k,r}^{(b)}}{\sum_r^{D_k} \lambda_{k,r}^{(b)}} > \theta \tag{22}$$

Finally, HODA, and by extension BTTDA, can extract a substantial amount of redundant features. These should be dropped after projection and before proceeding to the classification step [15]. In BTTDA in particular, redundant features can accumulate over the number of blocks, hampering performance. Furthermore, discriminant features across blocks can be heavily correlated since all blocks are independently optimizing the same discriminability criterion.

To tackle these issues, extracted features are first decorrelated and scaled using a whitening principle components analysis (PCA) transformation, retaining all principal components. Relevant PCA components can be identified by calculating the univariate Fisher score $\phi(i)$ for each component i after PCA, calculated as

$$\phi(i) = \frac{\sum_c^C N_c \left[\bar{g}_i(c) - \bar{\bar{g}}_i\right]^2}{\sum_n^N \left[g_i(n) - \bar{g}_i(c_n)\right]^2} \tag{23}$$

Only features where $\phi(i) > 1$, i.e., between-class variance is greater than within-class variance, are retained. If there are no extracted features with $\phi(i) > 1$, only the feature with the highest $\phi(i)$ is retained.

3 Experiments

3.1 Datasets and decoders

We evaluated our proposed model in two offline EEG-based BCI decoding problems: the event-related potential (ERP) and motor imagery (MI) paradigms using the openly available Mother of All BCI Benchmarks (MOABB) datasets (version 1.2.0) [35]. MOABB is widely accepted as a suitable benchmark for decoders aimed at classical BCI problems, allowing fair comparison of machine learning classifiers independent from data preprocessing. Details about these datasets are found in table B1. The ERP decoding task focuses on distinguishing target from non-target ERPs, while the MI tasks consists of distinguishing different imagined or performed limb movements. Within-session classification performance was assessed using stratified 5-fold cross-validation. Performance was calculated as the area under the receiver operating characteristic curve (ROC-AUC) for binary classification problems and accuracy for multi-class problems, in line with MOABB benchmarking framework. Average performance scores are balanced over dataset by taking the mean of the per-dataset average performance scores.

To use HODA, BTTDA, and PARAFACDA as a decoder, they are paired with LDA to classify the extracted features (HODA+LDA). Hyperparameters candidates $\theta \in \{0, 0.1, 0.2, \dots 1\}$ for all three decoders and $b \in \{1, 2 \dots, 16\}$ in the case PARAFACDA+LDA and BTTDA+LDA were tuned each evaluation fold using nested, stratified 5-fold cross-validation. Other HODA hyperparameters were set to $\epsilon = 1 \times 10^{-6}$ and $I_{\max} = 128$.

Differences in classification score between these proposed decoders were statistically verified using one-sided Wilcoxon rank-sum tests performed per dataset and decoder pair on the cross-validated scores per subject and session. Following the MOABB evaluation framework, meta-analyses for all ERP and MI datasets respectively were performed using the Stouffer method and effect size was determined as the standardized mean difference (SMD) between classification scores.

As additional comparison with other commonly used decoders, we selected a subset of the decoders evaluated by Chevallier et al. [36]. These decoders have been thoroughly evaluated on the MOABB benchmark to identify them as generally accepted state-of-the-art methods. For the ERP task, these included the Riemannian Geometry-based classifiers ERPCov+MDM, ERPCovSVD+MDM, XDAWNCov+MDM, XDAWNCov+TS+SVM and the linear classifier. For the MI task, the comparison methods were selected from Riemannian methods ACM+TS+SVM, FgMDM, TS+EL, and the deep learning classifiers EEGTCNet and ShallowConvNet. We refer to Chevallier et al. [36] for the description, implementation details and references of these methods.

3.2 Event-Related Potentials

ERPs are spatiotemporal features, with each sample forming a second-order tensor with $K = 2$ modes (a matrix), representing EEG channels and time samples per epoch.

The ERP datasets listed in table B1 are first processed according to the MOABB framework. EEG signals were recorded at the sample rate given by table B1 and band-pass filtered between 1 Hz and 24 Hz. The signals were cut into epochs starting from stimulus onset with a dataset-specific length given by table B1. For HODA+LDA, PARAFACDA+LDA, and BTTDA+LDA decoders, epochs were further downsampled to 48 Hz.

When considering grand average ROC-AUC over all evaluated ERP datasets as reported in table 1, the full BTTDA+LDA model (avg. ROC-AUC: 91.25±6.77%) outperforms PARAFAC+LDA (90.94±6.90%), and both in turn outperform HODA+LDA (88.89±7.04%). The meta-analysis shown in fig. 5 revealed the following significant effects: BTTDA+LDA > HODA+LDA ($p = 5.65 \times 10^{-65}$, SMD=1.17), PARAFACDA+LDA > HODA+LDA ($p = 2.47 \times 10^{-58}$, SMD=1.06), and BTTDA+LDA > PARAFAC+LDA ($p = 4.90 \times 10^{-15}$, SMD=0.50). Both BTTDA+LDA and PARAFACDA+LDA always significantly outperform HODA+LDA. BTTDA+LDA significantly outperforms PARAFACDA+LDA in 9 out of 14 datasets.

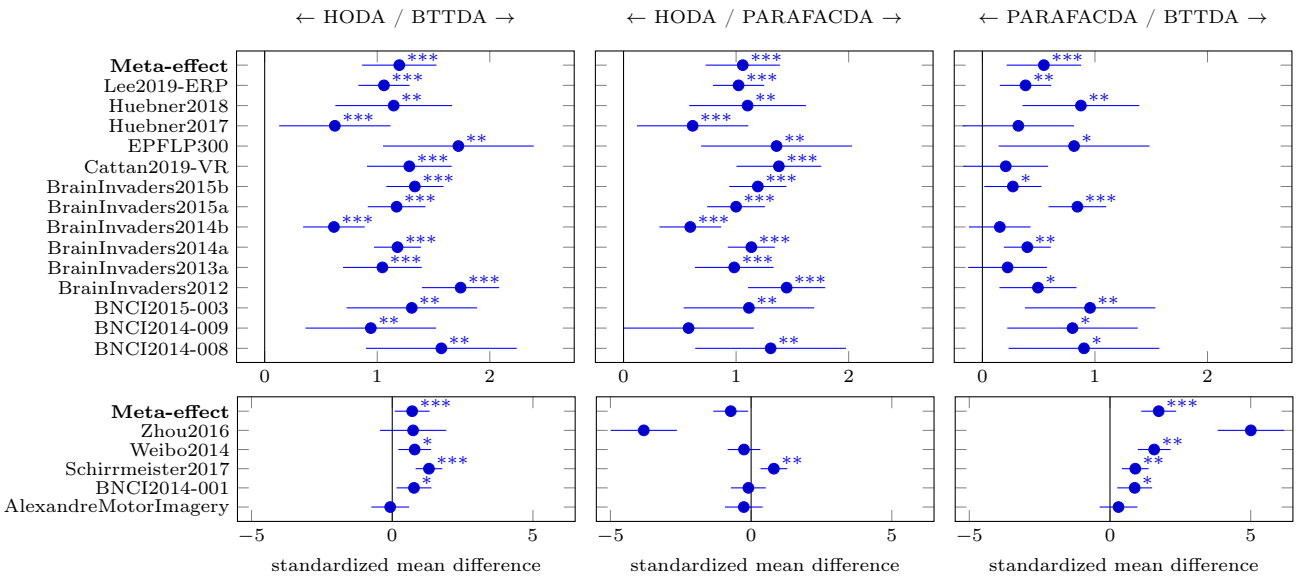

Figure 5: Meta-analysis of decoder classification performance comparisons per dataset. Analyses were performed on ROC-AUC score for ERP datasets (top) and accuracy for MI datasets (bottom). For the evaluated ERP datasets, BTTDA always outperforms HODA. BTTDA outperforms HODA for 3 out of 5 MI datasets. $***$: $p < 0.001$; $**$: $p < 0.01$, $*$: $p < 0.05$.

Significance and effect sizes for all evaluated ERP datasets are reported in table C2.

Compared to the state-of-the-art XDAWN-Cov+TS+SVM decoder, BTTDA+LDA scores better in 8 out of 14 datasets, combined with a moderate increase in grand average ROC-AUC ($91.25 > 90.82$).

Full cross-validation results can be retrieved from additional file item 2.

3.3 Motor Imagery

For MI, discriminatory information is encoded in the EEG data as ERS/Ds. Contrary to the time-domain analyses performed on ERPs, ERS/Ds are discerned in the power expressed in the time-frequency domain. For the MI task, we transform the EEG signal into the time-frequency domain, forming third-order tensors, with $K = 3$ modes respectively representing channels, frequencies, and time bins.

To achieve this, the EEG signals in the MI datasets listed in table B1 are first processed using the MOABB motor imagery pipeline. EEG signals were recorded at the sample rate given by table B1 and band-pass filtered between 8 Hz and 32 Hz. The signals were then cut into epochs starting from stimulus onset with a dataset-specific length given by table B1. Custom postprocessing to convert epochs to third-order tensors extracted the magnitude of the complex Morlet-wavelet transform with 17 logarithmically spaced frequencies from 8 Hz to 32 Hz and a varying number of cycles logarithmically spaced from 4 to 16. Finally, the magnitude envelope was downsampled to 32 Hz using an anti-aliasing filter and decimation.

In line with the MOABB method, only the first three classes per dataset were used. When considering grand average classification accuracies over all evaluated MI datasets as reported in table 2, the full BTTDA+LDA model (avg. accuracy: $64.52 \pm 12.23\%$) outperforms PARAFACDA+LDA ($58.89 \pm 11.27\%$) and HODA+LDA ($61.00 \pm 11.11\%$). The meta-analysis shown in fig. 5 revealed the following significant effects: BTTDA+LDA > HODA+LDA

Pipelines	BNCI2014-008	BNCI2014-009	BNCI2015-003	BrainInvaders2012	BrainInvaders2013a
ERPCov+MDM	74.30 ± 9.77	81.16 ± 10.13	76.79 ± 10.95	78.77 ± 10.32	80.59 ± 9.36
ERPCov(svdn4)+MDM	75.42 ± 9.91	84.52 ± 8.83	76.93 ± 11.26	79.02 ± 10.53	82.07 ± 8.46
XDAWN+LDA	82.24 ± 5.26	64.03 ± 3.91	78.62 ± 7.19	64.41 ± 4.14	76.74 ± 7.16
XDAWNCov+MDM	77.62 ± 9.81	92.04 ± 5.97	83.08 ± 7.55	88.22 ± 5.90	90.97 ± 5.52
XDAWNCov+TS+SVM	85.61 ± 4.43	93.43 ± 5.11	82.95 ± 8.57	90.99 ± 4.79	**92.71 ± 4.92**
HODA+LDA	85.20 ± 4.62	93.23 ± 4.07	82.74 ± 7.14	86.74 ± 5.33	89.78 ± 6.12
PARAFACDA+LDA	85.99 ± 4.73	94.03 ± 4.92	84.71 ± 7.54	90.80 ± 4.87	91.78 ± 5.81
BTTDA+LDA	**86.20 ± 4.61**	**94.40 ± 4.59**	**84.99 ± 7.37**	**91.27 ± 4.40**	92.08 ± 5.50

Pipelines	BrainInvaders2014a	BrainInvaders2014b	BrainInvaders2015a	BrainInvaders2015b	Cattan2019-VR
ERPCov+MDM	71.62 ± 11.17	78.57 ± 12.36	80.02 ± 10.07	75.04 ± 15.85	80.76 ± 10.07
ERPCov(svdn4)+MDM	72.11 ± 11.64	76.48 ± 12.83	77.92 ± 10.33	77.09 ± 15.81	80.67 ± 9.47
XDAWN+LDA	66.60 ± 7.54	83.73 ± 10.62	76.02 ± 10.46	77.22 ± 13.73	67.16 ± 6.11
XDAWNCov+MDM	80.88 ± 11.01	91.58 ± 10.02	92.57 ± 5.03	83.48 ± 12.05	88.53 ± 7.34
XDAWNCov+TS+SVM	85.77 ± 9.75	**91.88 ± 9.94**	93.05 ± 4.98	**84.56 ± 12.09**	90.68 ± 6.29
HODA+LDA	83.76 ± 9.50	87.23 ± 11.00	90.96 ± 5.68	82.19 ± 11.68	88.21 ± 8.59
PARAFACDA+LDA	87.24 ± 9.55	90.00 ± 9.70	92.95 ± 4.51	84.33 ± 12.39	91.22 ± 7.95
BTTDA+LDA	**87.71 ± 9.36**	90.41 ± 10.59	**93.44 ± 4.38**	84.47 ± 12.26	**91.47 ± 7.37**

Pipelines	EPFLP300	Huebner2017	Huebner2018	Lee2019-ERP	Average
ERPCov+MDM	71.97 ± 10.88	94.47 ± 8.26	95.15 ± 3.72	74.43 ± 13.26	79.55 ± 10.76
ERPCov(svdn4)+MDM	71.44 ± 10.20	96.21 ± 6.50	96.61 ± 1.89	82.47 ± 12.56	80.64 ± 10.49
XDAWN+LDA	62.98 ± 5.38	97.74 ± 2.84	97.54 ± 1.58	96.45 ± 3.93	77.96 ± 7.19
XDAWNCov+MDM	83.20 ± 9.05	98.07 ± 2.09	97.78 ± 1.04	97.70 ± 2.68	88.98 ± 7.53
XDAWNCov+TS+SVM	84.29 ± 8.53	**98.69 ± 1.78**	**98.47 ± 0.97**	**98.41 ± 2.03**	90.82 ± 6.82
HODA+LDA	82.09 ± 8.11	97.70 ± 3.09	97.61 ± 1.57	97.00 ± 2.72	88.89 ± 7.04
PARAFACDA+LDA	85.13 ± 8.72	98.52 ± 1.86	98.36 ± 0.93	98.09 ± 1.84	90.94 ± 6.90
BTTDA+LDA	**85.95 ± 8.06**	98.54 ± 1.84	98.40 ± 0.90	98.13 ± 1.83	**91.25 ± 6.77**

Table 1: Area under the receiver operating characteristic curve for cross-validated within-session evaluation of HODA+LDA and our proposed decoders PARAFACDA+LDA and BTTDA+LDA evaluated on ERP datasets. Scores for other decoders were taken from Chevallier et al. [36]. BTTDA+LDA always outperforms HODA+LDA and PARAFACDA+LDA, except for datasets, and consistently is nearly on par with or outperforms the state-of-the-art XDAWNCov+TS+SVM decoder.

($p = 6.20 \times 10^{-5}$, SMD=0.75), BTTDA+LDA > PARAFAC+LDA ($p = 4.00 \times 10^{-6}$, SMD=1.48). BTTDA+LDA outperforms HODA+LDA except for datasets Zhou2016 and AlexandreMotorImagery. PARAFACDA+LDA outperforms HODA+LDA for dataset Schirrmeister2017. BTTDA+LDA outperforms PARAFACDA+LDA except for datasets Zhou2016 and AlexandreMotorImagery. Significance and effect sizes for all evaluated MI datasets are reported in table C3.

All of HODA+LDA (avg. accuracy 61.00 ± 11.11) and our proposed decoders PARAFACDA+LDA (58.89 ± 11.27) and BTTDA+LDA (64.52 ± 12.23) score substantially lower than state-of-the-art decoder ACM+TS+SVM (75.77 ± 11.12).

Full cross-validation results can be retrieved from additional file item 2.

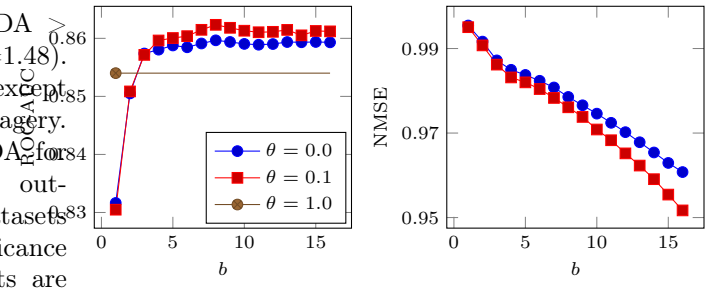

Figure 6: Cross-validated BTTDA+LDA ROC-AUC (left) and BTTDA normalized mean squared error (NMSE) (right) for dataset BNCI2014-008 as a function of the number of blocks b and the hyperparameter θ which controls the block dimensions. More effective class separation occurs as b increases while normalized mean squared error (NMSE) decreases. Eventually, overfitting occurs and class separation performance drops or plateaus depending on the effectiveness of feature selection as shown here.

3.4 Impact of block dimension and number of blocks

To analyze the contribution of extra feature blocks extracted by BTTDA over the first one found by HODA, we perform the following analyses on ERP dataset BNCI2014-008 chosen for its minimal computational requirements. We investigated cross-validated within-session ROC-AUC scores as function of the number of blocks (b) and hyperparameter θ, shown in fig. 6 (left) averaged over all subjects. b was varied from 1 to 16, while θ was chosen from $\{0.0, 0.1, 0.2, \ldots, 1.0\}$. Full results are presented in additional file item 3. Below, we report on selected of θ choices. When $\theta = 0$, the BTTDA model corresponds to the PARAFACDA decoder. $\theta = 0.1$ yielded the highest BTTDA+LDA ROC-AUC. $\theta = 1$ resulted in the highest overall HODA+LDA ($b = 1$) performance. At $\theta = 1$, no blocks other than the initial block can be modeled, since $\theta = 1$ by definition explains all data in the dataset and further forward modeling fails.

At $b = 1$, corresponding to the HODA model, sparse models with $\theta = 0$ (avg. ROC-AUC 83.16%) and $\theta = 0.1$ (avg. ROC-AUC 83.05%) are substantially lower than the optimal performance at $\theta = 1.0$ (avg. ROC-AUC 85.40). Moving from the HODA model ($b = 1$) to the BTTDA and PARAFACDA model allows the extraction of more blocks ($b \geq 1$). With this relaxation, PARAFACDA and BTTDA ($\theta = 0.1$) exceed HODA the at $b = 3$ (avg. ROC-AUC 85.74% and 85.71% respectively), while maintaining lower reduced dimensions than the (high) optimal dimensions for HODA ($\theta = 1.0$). Eventually, BTTDA its reaches the highest overall ROC-AUC at $b = 8$ (avg. ROC-AUC 86.23%). In general, when only a single block is used, a high θ is needed. When more blocks are used, higher θ deteriorates performance. Higher performance can be reached by choosing a low θ and $b > 1$, resulting in multiple blocks with low dimensions.

Additionally, fig. 6 (right) shows the effectiveness of the forward modeling step measured as the cross-validated normalized mean squared error (NMSE) when reconstructing the original data from the truncated BTTDA decomposition $\hat{\underline{\mathbf{X}}}^{(B)} = \sum_b^B \underline{\mathbf{G}}^{(b)} \times \{\mathbf{U}^{(b)}\}$, with NMSE is calculated as:

$$\text{NMSE}\left(\underline{\mathbf{X}}, \hat{\underline{\mathbf{X}}}^{(B)}\right) = \frac{\sum_n^N \left\|\underline{\mathbf{X}}(n) - \hat{\underline{\mathbf{X}}}^{(B)}(n)\right\|_{\text{F}}^2}{\sum_n^N \|\underline{\mathbf{X}}(n)\|_{\text{F}}^2} \quad (24)$$

NMSE decreases monotonically with b for both $\theta = 0.0$ and $\theta = 0.1$. In general, NMSE decreases faster

Pipelines	AlexandreMotorImagery	BNCI2014-001	Schirrmeister2017	Weibo2014	Zhou2016
ACM+TS+SVM	69.37 ± 15.07	$\mathbf{77.82 \pm 12.23}$	82.50 ± 10.20	$\mathbf{63.89 \pm 11.01}$	$\mathbf{85.25 \pm 4.06}$
EEGTCNet	34.17 ± 1.86	41.65 ± 13.73	71.11 ± 11.96	17.95 ± 3.88	37.19 ± 2.57
FgMDM	65.63 ± 15.63	70.14 ± 15.13	82.97 ± 10.08	56.94 ± 9.26	83.07 ± 4.96
ShallowConvNet	50.00 ± 12.94	72.47 ± 16.50	85.13 ± 9.57	48.94 ± 10.36	85.02 ± 3.78
TS+EL	$\mathbf{69.79 \pm 13.75}$	72.38 ± 14.85	$\mathbf{85.53 \pm 9.40}$	63.84 ± 8.77	84.54 ± 4.93
HODA+LDA	50.00 ± 15.17	53.84 ± 12.46	72.18 ± 9.02	54.69 ± 10.53	74.27 ± 6.27
PARAFACDA+LDA	46.67 ± 12.22	53.49 ± 12.28	76.11 ± 13.29	53.49 ± 11.04	64.68 ± 6.00
BTTDA+LDA	49.58 ± 15.68	57.42 ± 13.96	79.24 ± 12.51	59.36 ± 11.61	77.02 ± 4.04

Pipelines	Average
ACM+TS+SVM	$\mathbf{75.77 \pm 11.12}$
EEGTCNet	40.41 ± 8.45
FgMDM	71.75 ± 11.71
ShallowConvNet	68.31 ± 11.43
TS+EL	75.22 ± 10.95
HODA+LDA	61.00 ± 11.11
PARAFACDA+LDA	58.89 ± 11.27
BTTDA+LDA	64.52 ± 12.23

Table 2: Cross-validated classification accuracies for within-session evaluation to of HODA+LDA and our proposed decoders PARAFACDA+LDA and BTTDA+LDA, evaluated on three-class motor imagery datasets. Tensor-based methods generally score lower than Riemannian Geometry-based decoders. BTTDA outperforms Accuracies for other decoders were taken from Chevallier et al. [36].

as θ increases. For $\theta = 1$, reconstruction NMSE at $b = 0$ is near zero (1.32×10^{-30}) since no information is lost in the full-rank decomposition.

3.5 Interpretable decomposition

The following qualitative analysis reveals the model interpretability provided by the forward modeling step, by relating patterns in the reconstructed data to expected effects visible in the neural data at hand.

All three proposed models were trained on the combined subjects in BNCI2014-008 for ERP classification and AlexMI for MI. To allow proper visual inspection, the ERP epochs were extended with a pre-stimulus interval of 0.2 s for baseline correction and the original sample rate of 256 Hz was kept. The MI epochs were sampled at 250 Hz after time-frequency transformation. The number of blocks in this example was set to $B = 2$ and hyperparameters θ were tuned using 5-fold stratified cross-validation with entire-subject holdouts to determine the best hyperparameter for cross-subject decoding (ERP: $\theta = 0.3$, MI: $\theta = 0.7$). Each model was retrained with these hyperparameters on the full data combined over all subjects. Using these models, These models then generated reconstructed contrasts $\underline{\bar{\mathbf{C}}}^{(b)}_{c_2-c_2}$ between classes c_2 and c_1 for $b = 1$ and $b = 2$ as in

$$\underline{\bar{\mathbf{C}}}^{(b)}_{c_2-c_1} = \left[\underline{\bar{\mathbf{G}}}^{(b)}_{c_2} - \underline{\bar{\mathbf{G}}}^{(b)}_{c_1}\right] \times \left\{\mathbf{A}^{(b)}\right\} \tag{25}$$

with c_2 target and c_1 non-target trials for ERP, and c_2 right hand imagery and c_1 rest for MI. These contrasts, together with the grand-average contrast, are shown in fig. 7

The grand-average ERP contrast shows an entangled superposition of several different ERP components [37]. The activation patterns of the first two blocks disentangle this contrast in effects that can be related to ERP literature in the context of the classic visual P300 matrix speller task in BNCI2014-008 [38].

Block 1 exhibits positive and negative peaks in the lateral parieto-occipital regions corresponding to the visual cortex. The first positive peak and 2 negative peaks (P1, N1, N2) correspond to early components reflecting the task-related visual processing modulated by a mix of visual fixation and visual attention [39]. Block 2 has a more central scalp expression, and shows 2 positive peaks (P3a, P3b). Together with the residual positive activation between 0.4 s and 0.6 s in block 1, these constitute the processing of the attention-modulated detection of rare stimuli present in the P300 matrix speller task [40].

For motor imagery, results are displayed in the time-frequency domain. Positive values indicate event-related synchronization, negative values desynchronization. Upon visual inspection, the grand-average contrast shows no dominant pattern of synchronization or desynchronization, possibly due to the limited dataset size.

BTTDA decomposition extracts two distinct effects. Block 1 shows a persistent desynchronization between 9Hz and 13Hz most prominent in the left central area. For right-hand motor imagery, this corresponds to expected task-specific and localized high-μ band desynchronization in the contralateral motor cortex [41, 42]. Block 2 exhibits a synchronization between 8 and 12 Hz over the parieto-occipital region, from 1.2 to 2.2 s. This may be interpreted as the α band surround-ERS observed during hand movement [43, 44, 42].

4 Discussion

4.1 Contribution

The HODA model used for BCI decoding can be constrained by its Tucker structure. We introduced a more flexible generalization termed BTTDA with a block-term tensor structure. We also introduce PARAFACDA, a special case of BTTDA expressed as a sum of multilinear rank-1 terms. Our results show that BTTDA consistently scores on par or significantly higher than HODA as a supervised dimensionality reduction technique for BCI decoding. BTTDA managed to outperform HODA with 2.36%pt. on average for ERP datasets, and 2.75%pt. for MI datasets. PARAFACDA also scored 0.31%pt. higher than HODA in ERP datasets but was outperformed by BTTDA overall.

BTTDA yields state-of-the-art decoding perfor-

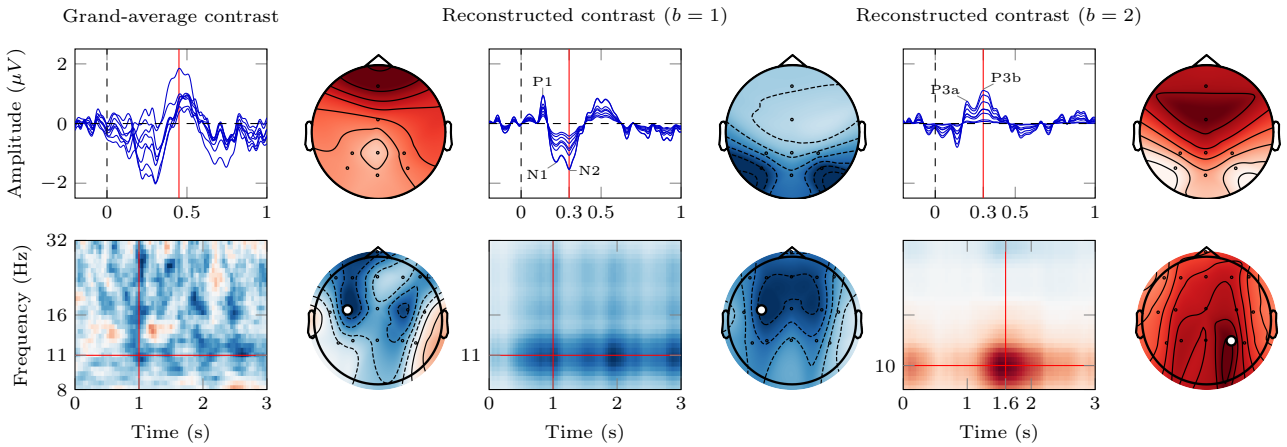

Figure 7: Per-block forward BTTDA model activation pattern contrasts and overall grand-average contrast for ERP dataset BNCI2014-008 (top) and MI dataset AlexMI (bottom). Red lines indicate the slices generating the scalp plot. In the bottom row, the white dot indicates for which channel the time-frequency spectrum was plotted. The ERP is decomposed in parieto-occipital components (P1, N1, N2) corresponding to visual processing and fronto-central components (P3a,P3b) related to task processing. The right-hand motor imagery (MI) ERS/D is decomposed in mostly contralateral high-μ band desynchronization, and parietal α-band synchronization.

mance for ERP datasets in the MOABB benchmark, but fails to do so for MI datasets. While this effect is not consistent over all ERP datasets, and the increase is often rather low (<2%pt.), it averages out over all datasets as a moderate increase of 0.43%pt. We note that performances of other decoders for these problems already achieve relatively high binary classification performance, which does not always leave room for improvement.

As mentioned above, results for MI were substantially lower than expected. Not only does BTTDA perform poorly, but the baseline HODA model as well. This is unexpected since it conflicts with literature which uses HODA to effectively classify MI from time-frequency transforms [15, 2, 19, 20]. If the issues hampering HODA performance can be identified, BTTDA could gain ground on the state of the art. We believe poor MI performance in our case could stem from the following issues. The time-frequency decomposition and data transformations or their parameters used in this study might not be suited to capture the relevant ERS/D information necessary for performant classification. For fair comparison with other MOABB decoders, the standard MOABB preprocessing pipeline was followed, which might interfere with our postprocessing. In this case, MI decoding might benefit from different preprocessing, transformation or tensorization techniques. On the other hand, hyperparameter selection could require more candidates or cross-validation folds due to the combination of $K = 3$ and larger data size following to the time-frequency transformation. Solutions for this problem can be computationally expensive.

Finally, due its the inherent forward modeling steps, BTTDA is intrinsically an explainable model which allows for interpretation of the signal components modeled by the tensor blocks. While the weights of the backward projections are hard to interpret [32], the activation patterns and contrasts after forward projection can reveal patterns in neural data. Qualitative analyses showed that block activation patterns correspond to task-related physiological processes for both ERP and MI classification problems. Given informed or correctly tuned hyperparameters, this method could be used to, e.g., separate and identify neural processes based on the task-related information in the class labels. More generally, BTTDA can achieve an effective unmixing of signal generators

relevant to the classification problem at hand, which might otherwise not be properly separated within the constraints of the HODA model. A point of care, however, arises from the deflation scheme: some processes might already be partially explained by previous blocks. In this case, information from a single physiologic process might not be modeled using only a single block. Hence, previous blocks might need to be taken into account to properly interpret a block activation pattern.

4.2 Modeling assumptions

We assume the main benefit of BTTDA stems from the following two aspects. Given fixed block dimensions, extra BTTDA blocks with proper feature selection can discover more discriminant information over HODA. While no proof is given here, we show that NMSE monotonically decreases. This suggests that all the variation in the signal will eventually be explained by the model while still extracting features that are maximally discriminant. Eventually, the number of blocks will reach a point of diminishing validation score returns. At this point, adding extra features to the decision classifier increases the risk of overfitting instead of adding extra useful discriminatory information. Hence, performance increases with the number of blocks until overfitting occurs.

On the other hand, a BTTDA solution is more parsimonious than a HODA solution can achieve due to its block-term structure compared to HODA's full Tucker structure, as illustrated by fig. 1. In other words, the same discriminative information captured by a relatively large Tucker-structured core tensor could be expressed more sparsely with a small number of block-terms, while avoiding redundant features. The PARAFAC structure employed in PARAFACDA is even more sparse, which could be a benefit or a drawback depending on the amount of regularization required, or on the true underlying structure of the data. BTTDA with a few, sparse blocks might perform worse then a dense HODA solution, adding extra BTTDA blocks eventually overpasses the HODA solution as indicated by fig. 6.

The enhanced performance could also partially stem from BTTDA's internal model of the data covariance. Since HODA estimates one within-class scatter matrix $\mathbf{S}_{-k,\mathrm{w}} \in \mathbb{R}^{D_k \times D_k}$ per mode during training, its overall model of the data scatter is determined by these per-mode scatter matrices as a Kronecker product $\mathbf{S}_{-1,\mathrm{w}} \otimes \mathbf{S}_{-2,\mathrm{w}} \otimes \cdots \otimes \mathbf{S}_{-K,\mathrm{w}}$. This corresponds to the assumption that the EEG data is drawn from a multilinear normal distribution [45]. Similar assumptions are made in ERP decoding algorithms such as Spatial-Temporal Discriminant-Analysis [46] and LCMV-beamforming with Kronecker covariance structure [47]. However, it is known that EEG covariance cannot fully be expressed as a single Kronecker product. Rather, it is more accurately modeled as a sum of multiple Kronecker products [48, 49]. Since BTTDA iteratively fits HODA models to the residual error, each with its own multilinear covariance model, it allows modeling multiple different multilinear covariance terms, refining the internal covariance model. This way, multiple effects with corresponding multilinear distributions can be extracted.

Finally, the drop in performance for PARAFACDA in MI datasets is attributed to the PARAFAC interaction of the model's rank-1 term structure with the multi-class nature of the MI problems. PARAFACDA only extracts a single feature per block, which cannot properly separate more than 2 classes. Further blocks are not properly adapted to take into account which classes have been separated by earlier blocks, hence extracting more PARAFAC blocks might not be helpful.

In summary, we conclude there is an effective added value in iteratively extracting multiple block terms. The flexibility of the BTTDA model is both expressed in its ability to capture more discriminant information with more parsimony, and in its ability to capture effects which cannot be expressed by the HODA model, such as the EEG covariance structure. This makes it specifically suited to tackle classification problems encountered in brain-computer interfacing.

4.3 Model selection

BTTDA trades the rigid HODA model for increased model complexity with more hyperparameters to

tune, which expands the solution space to settings where performance can be improved. Extracting more blocks and tuning the hyperparameters increases the time complexity of fitting BTTDA-based models compared to HODA-based models. HODA feature extraction can be solved with time complexity

$$\mathcal{O}\left(|\Theta| \, F I_{\max} N K^2 D^{K+1}\right) \tag{26}$$

whereas BTTDA training and model selection increases this to

$$\mathcal{O}\left(|\Theta| \, F B I_{\max} N K^2 D^{K+1}\right) \tag{27}$$

with Θ the set of θ candidates and F the number of cross-validation folds for hyperparameter tuning. Appendix A shows complexity derivation. Overall, the BTTDA approach shifts the focus of tensor discriminant analysis from finding optimal projections to model selection driven by computation.

The proposed θ-controlled selection procedure efficiently reduces the computational demand compared to tuning all hyperparameters $\left\{\left(R_1^{(b)}, R_2^{(b)}, \ldots, R_K^{(b)}\right)\right\}_b^B$. On the other hand, it also limits the chosen dimensions of each block to lie within a subset of all possible configurations. In a sense, this goes against the earlier proposition of increased model flexibility. Instances could occur where BTTDA offers little to no added value over the Tucker-structured HODA when both are given totally free choice of dimensions, but cases where BTTDA could achieve greater performance could equally be found. Finding these optimal-dimension configurations, can currently only be achieved through a costly, cross-validated hyperparameter search jointly over the dimensions of each block. Applications such as light-weight or mobile brain-computer interfaces should carefully weight potential performance gains against this computational demand. Future efforts should focus on more advanced automated hyperparameter selection methods relying on sparsity criteria, eigenvalue truncation or information criteria such as the ones used in BTTR [26], or other statistical measures depending on the application of the model.

Finally, we note that our proposed model selection procedure does not guarantee grouping coherent projections within the same block according to some desirable metrics. Features across blocks are heavily correlated, leading to a high degree of multicollinearity. Currently, this is corrected post-hoc by applying whitening and PCA. Solutions imposing some sense of subspace orthogonality between the extracted blocks could lead to a more effective feature extraction solution. Sparsity, pattern interpretability, minimal or maximal within-block feature correlation and ordering of blocks by decreasing discriminability are all examples of useful within-block grouping criteria.

As future work, The impact of higher-order tensors ($K > 3$) should be thoroughly investigated, since this could have a large impact on model behavior. We expect a dimensionality limit beyond which the forward modeling step cannot accurately regress from the low-dimensional latent tensors to the high-dimensional original tensors, introducing error in the input data for the next block which can stack up over blocks. The forward multilinear least squares problem is underdetermined hence prone to numerical instability, which calls for a suited regularization approach. Finally, other tensorization methods of the EEG data should be explored, such as time-lagged Hankel tensors [50] or tensors across subjects, conditions or sliding windows if they are appropriate given the available prior knowledge of the dataset.

5 Conclusion

We have introduced Block-Term Tensor Discriminant Analysis (BTTDA), a novel, tensor-based, supervised dimensionality reduction technique optimized for class discriminability, which adheres to the block-term tensor structure. BTTDA is a generalization of Higher Order Discriminant Analysis (HODA) and can also be applied as a special sum-of-rank-one tensors PARAFACDA model. The model is obtained by iteratively fitting HODA in a deflation scheme, leveraging a novel forward modeling step.

Via accompanying model selection hyperparameters, BCI decoders using BTTDA feature extraction can significantly outperform decoders based on HODA exceed state-of-the-art decoding performance on event-related potential problems (second-order

tensors) and outperform HODA in motor imagery problems (third-order tensors). The inherent forward model of BTTDA also allows interpreting the discriminative processes considered by the classifier.

Moving from the rigid Tucker tensor structure of HODA to the more flexible and sparse block-term structure shifts the focus from finding the best constrained multilinear projections to model and feature selection. This approach allows performance and generalization to be traded for computational cost, which is particularly relevant for BCI decoding problems. Because of its general implementation and minimal assumptions on data structure, BTTDA can equally be applied to classification for other neuroimaging modalities (MEG, ECoG, fNIRS, fMRI, EMG, etc.), or to tensor classification problems in other domains.

Code availability

The source code of the proposed BTTDA algorithm and the analyses performed in this work are available at `https://github.com/arnevdk/bttda`.

Additional data and materials

1. **Full ERP decoding cross-validation results**
   file: `erp_results.csv`
   format: *comma-separated values file*

2. **Full MI decoding cross-validation results**
   file: `mi_results.csv`
   format: *comma-separated values file*

3. **Full results of analysis in function of the number of blocks and block dimension**
   file: `block_theta_results.csv`
   format: *comma-separated values file*

Acknowledgments

We thank the Flemish Supercomputer Center (VSC) and the High-Performance Computing (HPC) center of KU Leuven for allowing us to execute our computational experiments on their systems. We also wish to acknowledge Dr. Axel Faes for his inspiration in conceptualizing this work.

AVDK has been is funded by the Belgian Fund for Scientific Research Flanders (G0A4321N) and received support of the special research fund of the KU Leuven (GPUDL/20/031) and the University of Lille under the Global PhD Scholarship Program. MMVH is supported by research grants received from the European Union's Horizon Europe Marie Sklodowska-Curie Action program (grant agreement No. 101118964), the European Union's Horizon 2020 research and innovation program (grant agreement No. 857375), the special research fund of the KU Leuven (C24/18/098), the Belgian Fund for Scientific Research – Flanders (G0A4118N, G0A4321N, G0C1522N), and the Hercules Foundation (AKUL 043).

The authors acknowledge the support of the RITMEA project co-financed by the European Union with the European Regional Development Fund, the French state, and the Hauts-de-France Region Council.

References

[1] Jonathan R Wolpaw, Jose Del R Millan, and Nick F Ramsey. "Brain-computer interfaces: Definitions and principles". In: *Handbook of clinical neurology* 168 (2020), pp. 15–23. ISSN: 0072-9752. DOI: 10.1016/b978-0-444-63934-9.00002-0.

[2] Fabien Lotte et al. "A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update". In: *Journal of neural engineering* 15.3 (Apr. 2018), p. 031005. ISSN: 1741-2552. DOI: 10.1088/1741-2552/aab2f2.

[3] Dean J Krusienski et al. "A comparison of classification techniques for the P300 Speller". In: *Journal of neural engineering* 3.4 (Oct. 2006), p. 299. ISSN: 1741-2552. DOI: 10.1088/1741-2560/3/4/007.

[4] Swati Aggarwal and Nupur Chugh. "Signal processing techniques for motor imagery brain computer interface: A review". In: *Array* 1 (Jan. 2019), p. 100003. ISSN: 2590-0056. DOI: 10.1016/j.array.2019.100003.

[5] Aybüke Erol and Borbála Hunyadi. "Tensors for neuroimaging: A review on applications of tensors to unravel the mysteries of the brain". In: *Tensors for Data Processing* (2022), pp. 427–482. DOI: https://doi.org/10.1016/B978-0-12-824447-0.00018-2.

[6] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. "A Multilinear Singular Value Decomposition". In: *SIAM Journal on Matrix Analysis and Applications* 21.4 (Jan. 2000), pp. 1253–1278. ISSN: 1095-7162. DOI: https://doi.org/10.1137/S0895479896305696.

[7] Jordi Sole-Casals et al. "Brain-computer interface with corrupted EEG data: a tensor completion approach". In: *Cognitive Computation* 10.6 (July 2018), pp. 1062–1074. ISSN: 1866-9964. DOI: 10.1007/s12559-018-9574-9.

[8] Frank L Hitchcock. "The expression of a tensor or a polyadic as a sum of products". In: *Journal of Mathematics and Physics* 6.1-4 (Apr. 1927), pp. 164–189. ISSN: 0097-1421. DOI: 10.1002/sapm192761164.

[9] Kianoush Nazarpour et al. "Parallel space-time-frequency decomposition of EEG signals for brain computer interfacing". In: *2006 14th European Signal Processing Conference*. IEEE. 2006, pp. 1–4.

[10] Lieven De Lathauwer. "Decompositions of a higher-order tensor in block terms—Part I: Lemmas for partitioned matrices". In: *SIAM Journal on Matrix Analysis and Applications* 30.3 (Jan. 2008), pp. 1022–1032. ISSN: 1095-7162. DOI: 10.1137/060661685.

[11] Lieven De Lathauwer. "Decompositions of a higher-order tensor in block terms—Part II: Definitions and uniqueness". In: *SIAM Journal on Matrix Analysis and Applications* 30.3 (Jan. 2008), pp. 1033–1066. ISSN: 1095-7162. DOI: 10.1137/070690729.

[12] Lieven De Lathauwer and Dimitri Nion. "Decompositions of a higher-order tensor in block terms—Part III: Alternating least squares algorithms". In: *SIAM journal on Matrix Analysis and Applications* 30.3 (Jan. 2008), pp. 1067–1083. ISSN: 1095-7162. DOI: 10.1137/070690730.

[13] Athanasios A Rontogiannis, Eleftherios Kofidis, and Paris V Giampouras. "Block-term tensor decomposition: Model selection and computation". In: *IEEE Journal of Selected Topics in Signal Processing* 15.3 (2021), pp. 464–475.

[14] Shuicheng Yan et al. "Discriminant analysis with tensor representation". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. ISSN: 1063-6919. June 2005, 526–532 vol. 1. DOI: 10.1109/CVPR.2005.131.

[15] Anh Huy Phan and Andrzej Cichocki. "Tensor decompositions for feature extraction and classification of high dimensional datasets". In: *Nonlinear Theory and Its Applications, IEICE* 1.1 (2010), pp. 37–68. ISSN: 2185-4106. DOI: https://doi.org/10.1587/nolta.1.37.

[16] Laura Frølich, Tobias S Andersen, and Morten Mørup. "Rigorous optimisation of multilinear discriminant analysis with Tucker and PARAFAC structures". In: *BMC Bioinformatics* 19.1 (May 2018), p. 197. ISSN: 1471-2105. DOI: 10.1186/s12859-018-2188-0. (Visited on 01/24/2022).

[17] Akinari Onishi et al. "Tensor classification for P300-based brain computer interface". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ISSN: 2379-190X. Mar. 2012, pp. 581–584. DOI: 10.1109/ICASSP.2012.6287946.

[18] Hiroshi Higashi et al. "Multilinear Discriminant Analysis With Subspace Constraints for Single-Trial Classification of Event-Related Potentials". In: *IEEE Journal of Selected Topics in Signal Processing* 10.7 (Oct. 2016), pp. 1295–1305. ISSN: 1941-0484. DOI: 10.1109/JSTSP.2016.2599297.

[19] Ye Liu, Qibin Zhao, and Liqing Zhang. "Uncorrelated Multiway Discriminant Analysis for Motor Imagery EEG Classification". In: *Int. J. Neural Syst.* (2015). DOI: 10.1142/S0129065715500136.

[20] Qian Cai et al. "Multilinear Discriminative Spatial Patterns for Movement-Related Cortical Potential Based on EEG Classification with Tensor Representation". In: *Computational Intelligence and Neuroscience* 2021 (May 2021). Ed. by Anastasios D. Doulamis, pp. 1–9. ISSN: 1687-5265. DOI: 10.1155/2021/6634672.

[21] Mina Jamshidi Idaji, Mohammad B. Shamsollahi, and Sepideh Hajipour Sardouie. "Higher order spectral regression discriminant analysis (HOSRDA): A tensor feature reduction method for ERP detection". In: *Pattern Recognition* 70 (Oct. 2017), pp. 152–162. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2017.05.004.

[22] Tania Jorajuría et al. "Oscillatory Source Tensor Discriminant Analysis (OSTDA): A regularized tensor pipeline for SSVEP-based BCI systems". In: *Neurocomputing* 492 (July 2022), pp. 664–675. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.07.103.

[23] Seyedeh N Aghili et al. "A spatial-temporal linear feature learning algorithm for P300-based brain-computer interfaces". In: *Heliyon* 9.4 (Apr. 2023), e15380. ISSN: 2405-8440. DOI: https://doi.org/10.1016/j.heliyon.2023.e15380.

[24] Qibin Zhao et al. "Higher order partial least squares (HOPLS): A generalized multilinear regression method". In: *IEEE transactions on pattern analysis and machine intelligence* 35.7 (July 2012), pp. 1660–1673. ISSN: 2160-9292. DOI: 10.1109/tpami.2012.254.

[25] Flavio Camarrone and Marc M Van Hulle. "Fast multiway partial least squares regression". In: *IEEE Transactions on Biomedical Engineering* 66.2 (2018), pp. 433–443.

[26] Axel Faes, Flavio Camarrone, and Marc M Van Hulle. "Single Finger Trajectory Prediction From Intracranial Brain Activity Using Block-Term Tensor Regression With Fast and Automatic Component Extraction". In: *IEEE Transactions on Neural Networks and Learning Systems* (2022), pp. 1–12. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2022.3216589.

[27] A Faes and Marc M Van Hulle. "Finger movement and coactivation predicted from intracranial brain activity using extended block-term tensor regression". In: *Journal of Neural Engineering* 19.6 (Nov. 2022), p. 066011. ISSN: 1741-2552. DOI: 10.1088/1741-2552/ac9a75.

[28] Flavio Camarrone et al. "Accurate Offline Asynchronous Detection of Individual Finger Movement From Intracranial Brain Signals Using a Novel Multiway Approach". In: *IEEE Transactions on Biomedical Engineering* 68.7 (July 2021), pp. 2176–2187. ISSN: 1558-2531. DOI: 10.1109/TBME.2020.3037934.

[29] Shoulin Huang et al. "Spectrum-weighted tensor discriminant analysis for motor imagery-based BCI". In: *IEEE Access* 8 (2020), pp. 93749–93759. DOI: 10.1109/access.2020.2995302.

[30] Huan Wang et al. "Trace Ratio vs. Ratio Trace for Dimensionality Reduction". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2007, pp. 1–8. DOI: 10.1109/cvpr.2007.382983.

[31] Olivier Ledoit and Michael NMI Wolf. "Honey, I Shrunk the Sample Covariance Matrix". In: *SSRN Electronic Journal* (2003). ISSN: 1556-5068. DOI: https://dx.doi.org/10.2139/ssrn.433840.

[32] Stefan Haufe et al. "On the interpretation of weight vectors of linear models in multivariate neuroimaging". In: *NeuroImage* 87 (Feb. 2014), pp. 96–110. ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2013.10.067.

[33] Lucas C. Parra et al. "Recipes for the linear analysis of EEG". In: *NeuroImage* 28.2 (Nov. 2005), pp. 326–341. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2005.05.032. URL: https://www.sciencedirect.com/science/article/pii/S1053811905003381.

[34] Abdeslem H Bentbib, Asmaa Khouia, and Hassane Sadok. "The LSQR method for solving tensor least-squares problems". In: *Electron. Trans. Numer. Anal* 55 (2022), pp. 92–111.

[35] Bruno Aristimunha et al. *Mother of all BCI Benchmarks*. Version 1.0.0. 2023. DOI: 10.5281/zenodo.10034223. URL: https://github.com/NeuroTechX/moabb.

[36] Sylvain Chevallier et al. "The largest EEG-based BCI reproducibility study for open science: the MOABB benchmark". In: *arXiv preprint arXiv:2404.15319* (2024).

[37] Steven J Luck and Emily S Kappenman. *The Oxford handbook of event-related potential components*. Oxford university press, 2011.

[38] Angela Riccio et al. "Attention and P300-based BCI performance in people with amyotrophic lateral sclerosis". In: *Frontiers in human neuroscience* 7 (2013), p. 732. ISSN: 1662-5161. DOI: 10.3389/fnhum.2013.00732.

[39] Matthias S Treder and Benjamin Blankertz. "(C) overt attention and visual speller design in an ERP-based brain-computer interface". In: *Behavioral and brain functions* 6.1 (2010), p. 28. ISSN: 1744-9081. DOI: 10.1186/1744-9081-6-28.

[40] Siri-Maria Kamp, Anthony R Murphy, and Emanuel Donchin. "The component structure of event-related potentials in the P300 speller paradigm". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 21.6 (Nov. 2013), pp. 897–907. ISSN: 1558-0210. DOI: 10.1109/tnsre.2013.2285398.

[41] Gert Pfurtscheller, Christa Neuper, and Gunther Krausz. "Functional dissociation of lower and upper frequency mu rhythms in relation to voluntary limb movement". In: *Clinical neurophysiology* 111.10 (Oct. 2000), pp. 1873–1879. ISSN: 1388-2457. DOI: 10.1016/s1388-2457(00)00428-4.

[42] Jonathan Wolpaw and Elizabeth Winter Wolpaw. "Brain–Computer Interfaces: Principles and Practice". In: (Jan. 2012). DOI: https://doi.org/10.1093/acprof:oso/9780195388855.001.0001.

[43] P Suffczynski et al. "Event-related dynamics of alpha band rhythms: a neuronal network model of focal ERD-surround ERS". In: (1999).

[44] Christian Gerloff et al. "Functional coupling and regional activation of human cortical motor areas during simple, internally paced and externally paced finger movements." In: *Brain: a journal of neurology* 121.8 (Aug. 1998), pp. 1513–1531. ISSN: 1460-2156. DOI: 10.1093/brain/121.8.1513.

[45] Martin Ohlson, M Rauf Ahmad, and Dietrich Von Rosen. "The multilinear normal distribution: Introduction and some basic properties". In: *Journal of Multivariate Analysis* 113 (Jan. 2013), pp. 37–47. ISSN: 0047-259X. DOI: 10.1016/j.jmva.2011.05.015.

[46] Yu Zhang et al. "Spatial-Temporal Discriminant Analysis for ERP-Based Brain-Computer Interface". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 21.2 (Mar. 2013), pp. 233–243. ISSN: 1558-0210. DOI: 10.1109/TNSRE.2013.2243471.

[47] Arne Van Den Kerchove et al. "Classification of Event-Related Potentials with Regularized Spatiotemporal LCMV Beamforming". In: *Applied Sciences* 12.6 (Mar. 2022), p. 2918. ISSN: 2076-3417. DOI: https://doi.org/10.3390/app12062918.

[48] Fetsje Bijma, Jan C de Munck, and Rob M Heethaar. "The spatiotemporal MEG covariance matrix modeled as a sum of Kronecker products". In: *NeuroImage* 27.2 (Aug. 2005), pp. 402–415. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2005.04.015.

[49] Jan Sosulski and Michael Tangermann. "Introducing block-Toeplitz covariance matrices to remaster linear discriminant analysis for event-related potential brain–computer interfaces". In: *Journal of neural engineering* 19.6 (Nov. 2022), p. 066001. ISSN: 1741-2552. DOI: 10.1088/1741-2552/ac9c98.

[50] Jean-Michel Papy, Lieven De Lathauwer, and Sabine Van Huffel. "Exponential data fitting using multilinear algebra: the single-channel and multi-channel case". In: *Numerical linear algebra with applications* 12.8 (2005), pp. 809–826.

[51] Pietro Aricò et al. "Influence of P300 latency jitter on event related potential-based brain–computer interface performance". In: *Journal of neural engineering* 11.3 (2014), p. 035008.

[52] Christoph Guger et al. "How many people are able to control a P300-based brain–computer interface (BCI)?" In: *Neuroscience letters* 462.1 (Sept. 2009), pp. 94–98. ISSN: 0304-3940. DOI: 10.1016/j.neulet.2009.06.045.

[53] Gijsbrecht Van Veen et al. "Building brain invaders: EEG data of an experimental validation". In: *arXiv preprint arXiv:1905.05182* (2019).

[54] Erwan Vaineau et al. "Brain invaders adaptive versus non-adaptive P300 brain-computer interface dataset". In: *arXiv preprint arXiv:1904.09111* (2019).

[55] Louis Korczowski et al. "Brain Invaders calibration-less P300-based BCI using dry EEG electrodes Dataset (bi2014a)". PhD thesis. GIPSA-lab, 2019.

[56] Louis Korczowski et al. "Brain invaders solo versus collaboration: Multi-user P300-based brain-computer interface dataset (bi2014b)". PhD thesis. GIPSA-lab, 2019.

[57] Louis Korczowski et al. "Brain Invaders calibration-less P300-based BCI with modulation of flash duration Dataset (bi2015a)". PhD thesis. GIPSA-lab, 2019.

[58] Louis Korczowski et al. "Brain invaders cooperative versus competitive: Multi-user p300-based brain-computer interface dataset (bi2015b)". PhD thesis. GIPSA-lab, 2019.

[59] Grégoire Cattan et al. "Dataset of an EEG-based BCI experiment in Virtual Reality and on a Personal Computer". In: *arXiv preprint arXiv:1903.11297* (2019).

[60] Ulrich Hoffmann et al. "An efficient P300-based brain–computer interface for disabled subjects". In: *Journal of Neuroscience methods* 167.1 (2008), pp. 115–125.

[61] David Hübner et al. "Learning from label proportions in brain-computer interfaces: Online unsupervised learning with guarantees". In: *PloS one* 12.4 (2017), e0175856.

[62] David Hübner et al. "Unsupervised learning for brain-computer interfaces based on event-related potentials: Review and online comparison [research frontier]". In: *IEEE Computational Intelligence Magazine* 13.2 (May 2018), pp. 66–77. ISSN: 1556-6048. DOI: 10.1109/mci.2018.2807039.

[63] Min-Ho Lee et al. "EEG dataset and OpenBMI toolbox for three BCI paradigms: An investigation into BCI illiteracy". In: *GigaScience* 8.5 (2019), giz002.

[64] Jan Sosulski and Michael Tangermann. "Electroencephalogram signals recorded from 13 healthy subjects during an auditory oddball paradigm under different stimulus onset asynchrony conditions". en. In: (2019). DOI: https://doi.org/10.6094/unifr/154576.

[65] Alexandre Barachant. "Commande robuste d'un effecteur par une interface cerveau machine EEG asynchrone". PhD thesis. Université de Grenoble, 2012.

[66] Michael Tangermann et al. "Review of the BCI competition IV". In: *Frontiers in neuroscience* 6 (2012), p. 55. ISSN: 1662-4548. DOI: 10.3389/fnins.2012.00055.

[67] Robin T Schirrmeister et al. "Deep learning with convolutional neural networks for EEG decoding and visualization". In: *Human brain mapping* 38.11 (2017), pp. 5391–5420. eprint: 1703.05051.

[68] Weibo Yi et al. "Evaluation of EEG oscillatory patterns and cognitive process during simple and compound limb motor imagery". In: *PloS one* 9.12 (2014), e114853.

[69] Bangyan Zhou et al. "A Fully Automated Trial Selection Method for Optimization of Motor Imagery Based Brain-Computer Interface". In: *PLOS ONE* 11.9 (Sept. 2016). Ed. by Bin He, e0162657. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0162657.

A Time complexity of HODA and BTTDA

A.1 Backward HODA algorithm

To obtain time complexity of the HODA backward model algorithm, we start by determining the number of operations within a single inner loop for given i and k. Since we are interested in worst-case complexity, assume all input dimensions D_k are equal to $\max_k(D_k) = D$ and all reduced dimensions R_k take their maximum value $R_k = D_k = D$.

The multi-mode products $\underline{\mathbf{X}}(n) \times_{-k} \{U\}$ require

$$(K-1)\left[DD\left(ND^{K-1}\right)\right] = N\left(K-1\right)D^{K+1} \quad (28)$$

operations. Calculating class means and centering the data requires

$$\begin{aligned} &C\left(DD^{K-1} + D + DD^{K-1}\right) \\ &= ND^K + D + ND^K \\ &= 2ND^K + D \end{aligned} \quad (29)$$

From this, the number of operations for the within-subject scatter matrix $\mathbf{S}_{-k,\mathrm{w}}$

$$D(ND^{K-1})D = ND^{K+1} \quad (30)$$

and the between-class scatter matrix $\mathbf{S}_{-k,\mathrm{b}}$

$$C\left(D1D\right) + CD^2 = 2CD^2 \quad (31)$$

can be obtained. φ_k can be calculated as $\mathrm{Tr}\left(\mathbf{U}_k^\intercal \mathbf{S}_{-k,\mathrm{b}} \mathbf{U}_k\right) / \mathrm{Tr}\left(\mathbf{U}_k^\intercal \mathbf{S}_{-k\mathrm{w}} \mathbf{U}_k\right)$ in

$$\begin{aligned} &(DDD + DDD + D) + (DDD + DDD + D) + 1 \\ &= 2\left(2D^3 D\right) + 1 = 4D^3 + 2D + 1 \end{aligned} \quad (32)$$

operations. The difference of the scatter matrices $\mathbf{S}_{-k,\mathrm{w}} - \varphi - \mathbf{S}_{-k,\mathrm{b}}$ then yields

$$D^2 + D^2 + D^2 = 3D^2 \quad (33)$$

operations, and its eigendecomposition

$$D^3 \quad (34)$$

Finally, the projection for orthogonalization $\mathbf{V}_k \mathbf{V}_k^\intercal \mathbf{S}_{k,\mathrm{t}} \mathbf{V}_k \mathbf{V}_k^\intercal$ adds

$$DDD + DDD + DDD + DDD = 4D^3 \quad (35)$$

and its eigendecomposition

$$D^3 \quad (36)$$

Together, this forms

$$\begin{aligned} &N(K-1)D^{K+1} + 2ND^K + D + 2CD^2 + 4D^3 \\ &\quad + 2D + 1 + 3D^2 + D^3 + 4D^3 + D^3 \\ &= NKD^{K+1} - ND^{K+1} + 2ND^K + 10D^3 \\ &\quad + (3 + 2C)D^2 + 3D + 1 \end{aligned} \quad (37)$$

From the number of operations, the time complexity can be derived as

$$\begin{aligned} &\mathcal{O}\left[NKD^{K+1} - ND^{K+1} + 2ND^K + 10D^3\right. \\ &\quad \left.+(3 + 2C)D^2 + 3D + 1\right] \\ &= \mathcal{O}\left(NKD^{K+1}\right) \end{aligned} \quad (38)$$

assuming $C < N$. The procedure in the inner loop over k and the outer loop over i is executed $I_{\max}K$ times, yielding

$$\mathcal{O}\left(I_{\max}KNKD^{K+1}\right) = \mathcal{O}\left(I_{\max}K^2ND^{K+1}\right) \quad (39)$$

A.2 Forward HODA algorithm

Similar to the previous derivation, we start by determining the operations within a single iteration of a nested over the i and k.

The first step is again a multi-mode product, $\underline{\mathbf{G}}(n) \times_{-k} \{\mathbf{A}\}$:

$$(K-1)\left[DD\left(ND^{K-1}\right)\right] = N\left(K-1\right)D^{K+1} \quad (40)$$

The second step requires least squares regression which can be solved in

$$\begin{aligned} &D\left(ND^{K-1}\right)D + D\left(ND^{K-1}\right)D + D^3 \\ &= 2ND^{K+1} + D^3 \end{aligned} \quad (41)$$

operations.

Together, this forms

$$\begin{aligned} &N(K-1)D^{K+1} + 2ND^{K+1} + D^3 \\ &= NKD^{K+1} - ND^{K+1} + 2ND^{K+1} + D^3 \end{aligned} \tag{42}$$

The time complexity to fit one iteration of the algorithm for the forward model is then

$$\begin{aligned} &\mathcal{O}\left(NKD^{K+1} - ND^{K+1} + 2ND^{K+1} + D^3\right) \\ &= \mathcal{O}\left(NKD^{K+1}\right) \end{aligned} \tag{43}$$

and, when integrating it in the inner and outer loops over i and k,

$$\mathcal{O}\left(I_{\max}KNKD^{K+1}\right) = \mathcal{O}\left(I_{\max}NK^2D^{K+1}\right) \tag{44}$$

This is the same asymptotic time complexity as the backwards modeling algorithm, since they are both dominated by the multi-mode product.

A.3 Backward BTTDA algorithm

Fitting the BTTDA model involves a loop over blocks b. At each iteration, a backward model is fit with complexity as in eq. (39) The core tensors $\underline{\mathbf{G}}^{(\mathrm{b})}(n)$ are extracted with the multi-mode product using

$$K\left[DD\left(ND^{K-1}\right)\right] = KND^{K+1} \tag{45}$$

operations

Next, the forward model is fit on these core tensors, with complexity as in eq. (44). The number of steps for the reconstructed tensors can similarly be obtained using eq. (45), and calculating the residual requires

$$ND^K \tag{46}$$

operations.

A single block b can thus be fit with complexity

$$\begin{aligned} &\mathcal{O}\left(I_{\max}NK^2D^{K+1} + KND^{K+1}\right. \\ &\quad \left. + I_{\max}NK^2D^{K+1} + KND^{K+1} + ND^K\right) \\ &= \mathcal{O}\left(I_{\max}NK^2D^{K+1}\right) \end{aligned} \tag{47}$$

The complexity when calculating all blocks is

$$\mathcal{O}\left(BI_{\max}NK^2D^{K+1}\right) \tag{48}$$

A.4 Hyperparameter tuning

Finally, proper decoding training relies heavily on tuning the hyperparameters θ and B through cross-validation. Let F be the number of cross-validation folds and Θ a set of θ candidates. We can take advantage of the fact that the BTTDA model can be fit on a fixed amount of blocks B, but intermediary blocks $1, 2, \cdots B$ can easily be extracted. This way, no second iteration over candidates for B is necessary and complexity can be kept at

$$\mathcal{O}\left(|\Theta|\, FBI_{\max}NK^2D^{K+1}\right) \tag{49}$$

B Datasets

Dataset	# Sub.	# Chan.	# Trials/class	Epoch len. (s)	S. freq. (Hz)	# Sess.	Ref.
ERP datasets							
BNCI2014-008	8	8	3500/700	1.0	256	1	[38]
BNCI2014-009	10	16	1440/288	0.8	256	3	[51]
BNCI2015-003	10	8	1500/300	0.8	256	1	[52]
BrainInvaders2012	25	16	640/128	1.0	128	2	[53]
BrainInvaders2013a	24	16	3200/640	1.0	512	8 (sub. 1-7) or 1	[54]
BrainInvaders2014a	64	16	990/198	1.0	512	up to 3	[55]
BrainInvaders2014b	38	32	200/40	1.0	512	3	[56]
BrainInvaders2015a	43	32	4131/825	1.0	512	3	[57]
BrainInvaders2015b	44	32	2160/480	1.0	512	1	[58]
Cattan2019-VR	21	16	600/120	1.0	512	2	[59]
EPFLP300	8	32	2753/551	1.0	2048	4	[60]
Huebner2017	13	31	364/112	0.9	1000	3	[61]
Huebner2018	12	31	364/112	0.9	1000	3	[62]
Lee2019-ERP	54	62	6900/1380	1.0	1000	2	[63]
Sosulski2019	13	31	7500/1500	1.2	1000	1	[64]
MI datasets							
AlexandreMotorImagery	8	16	20.0	3.0	512	1	[65]
BNCI2014-001	9	22	144.0	4.0	250	2	[66]
Schirrmeister2017	14	128	120.0	4.0	500	1	[67]
Weibo2014	10	60	80.0	4.0	200	1	[68]
Zhou2016	4	14	160.0	5.0	250	3	[69]

Table B1: MOABB datasets used for evaluation, with the number of subjects (# Sub.), the number of EEG channels (# Chan.), the number of trials or trials per class for ERP datasets (# Trials), the epoch length (Epoch len.), the sampling frequency (S. freq.), the number of sessions per subject (# Sess.) and the number of runs (# Runs). ERP datasets contain 2 classes, for MI datasets the first 3 classes were retained. ERP dataset Sosulski2019 was omitted due to technical problems. MI dataset PhysionetMI was omitted due to its high computational and storage demands. Adapted from [35] and [36].

C Pairwise statistics

Decoder 1	BTTDA				PARAFACDA	
Decoder 2	HODA		PARAFACDA		HODA	
	p	SMD	p	SMD	p	SMD
BNCI2014-008	3.91×10^{-3}	1.57	1.95×10^{-2}	0.90	7.81×10^{-3}	1.31
BNCI2014-009	5.86×10^{-3}	0.94	1.76×10^{-2}	0.80	5.47×10^{-2}	0.58
BNCI2015-003	2.93×10^{-3}	1.31	7.81×10^{-3}	0.96	6.84×10^{-3}	1.11
BrainInvaders2012	6.14×10^{-6}	1.74	1.98×10^{-2}	0.49	7.85×10^{-6}	1.45
BrainInvaders2013a	3.28×10^{-6}	1.05	8.00×10^{-2}	0.22	1.51×10^{-5}	0.98
BrainInvaders2014a	2.56×10^{-11}	1.18	2.81×10^{-3}	0.40	5.71×10^{-11}	1.14
BrainInvaders2014b	3.02×10^{-4}	0.62	1.34×10^{-1}	0.15	3.02×10^{-4}	0.59
BrainInvaders2015a	1.59×10^{-12}	1.17	3.22×10^{-7}	0.84	8.62×10^{-10}	1.00
BrainInvaders2015b	2.11×10^{-11}	1.33	4.37×10^{-2}	0.27	6.47×10^{-10}	1.19
Cattan2019-VR	6.68×10^{-6}	1.28	3.97×10^{-1}	0.21	2.38×10^{-6}	1.38
EPFLP300	3.91×10^{-3}	1.72	3.52×10^{-2}	0.81	3.91×10^{-3}	1.36
Huebner2017	3.66×10^{-4}	0.62	1.37×10^{-1}	0.32	4.88×10^{-4}	0.61
Huebner2018	1.22×10^{-3}	1.15	3.91×10^{-3}	0.88	3.17×10^{-3}	1.10
Lee2019-ERP	8.13×10^{-11}	1.06	3.30×10^{-3}	0.38	1.08×10^{-10}	1.02

Table C2: Results of one-sided Wilcoxon rank-sum tests comparing the per-subject cross-validated classification scores of the evaluated ERP decoders. Significance is reported as p, the effect size as the standardized mean difference (SMD).

Decoder 1	BTTDA				PARAFACDA	
Decoder 2	HODA		PARAFACDA		HODA	
	p	SMD	p	SMD	p	SMD
BNCI2014-008	3.91×10^{-3}	1.57	1.95×10^{-2}	0.90	7.81×10^{-3}	1.31
BNCI2014-009	5.86×10^{-3}	0.94	1.76×10^{-2}	0.80	5.47×10^{-2}	0.58
BNCI2015-003	2.93×10^{-3}	1.31	7.81×10^{-3}	0.96	6.84×10^{-3}	1.11
BrainInvaders2012	6.14×10^{-6}	1.74	1.98×10^{-2}	0.49	7.85×10^{-6}	1.45
BrainInvaders2013a	3.28×10^{-6}	1.05	8.00×10^{-2}	0.22	1.51×10^{-5}	0.98
BrainInvaders2014a	2.56×10^{-11}	1.18	2.81×10^{-3}	0.40	5.71×10^{-11}	1.14
BrainInvaders2014b	3.02×10^{-4}	0.62	1.34×10^{-1}	0.15	3.02×10^{-4}	0.59
BrainInvaders2015a	1.59×10^{-12}	1.17	3.22×10^{-7}	0.84	8.62×10^{-10}	1.00
BrainInvaders2015b	2.11×10^{-11}	1.33	4.37×10^{-2}	0.27	6.47×10^{-10}	1.19
Cattan2019-VR	6.68×10^{-6}	1.28	3.97×10^{-1}	0.21	2.38×10^{-6}	1.38
EPFLP300	3.91×10^{-3}	1.72	3.52×10^{-2}	0.81	3.91×10^{-3}	1.36
Huebner2017	3.66×10^{-4}	0.62	1.37×10^{-1}	0.32	4.88×10^{-4}	0.61
Huebner2018	1.22×10^{-3}	1.15	3.91×10^{-3}	0.88	3.17×10^{-3}	1.10
Lee2019-ERP	8.13×10^{-11}	1.06	3.30×10^{-3}	0.38	1.08×10^{-10}	1.02

Table C3: Results of one-sided Wilcoxon rank-sum tests comparing the per-subject cross-validated classification scores of the evaluated MI decoders. Significance is reported as p, the effect size as the standardized mean difference (SMD).