Higher-Order Partial Least Squares (HOPLS): A Generalized Multi-Linear Regression Method

Qibin Zhao, Cesar F. Caiafa, Danilo P. Mandic, Zenas C. Chao, Yasuo Nagasaka, Naotaka Fujii, Liqing Zhang and Andrzej Cichocki

Abstract—A new generalized multilinear regression model, termed the Higher-Order Partial Least Squares (HOPLS), is introduced with the aim to predict a tensor (multiway array) \( Y \) from a tensor \( X \) through projecting the data onto the latent space and performing regression on the corresponding latent variables. HOPLS differs substantially from other regression models in that it explains the data by a sum of orthogonal Tucker tensors, while the number of orthogonal loadings serves as a parameter to control model complexity and prevent overfitting. The low dimensional latent space is optimized sequentially via a deflation operation, yielding the best joint subspace approximation for both \( X \) and \( Y \). Instead of decomposing \( X \) and \( Y \) individually, higher order singular value decomposition on a newly defined generalized cross-covariance tensor is employed to optimize the orthogonal loadings. A systematic comparison on both synthetic data and real-world decoding of 3D movement trajectories from electrocorticogram (ECoG) signals demonstrate the advantages of HOPLS over the existing methods in terms of better predictive ability, suitability to handle small sample sizes, and robustness to noise.

Index Terms—Multilinear regression, Partial least squares (PLS), Higher-order singular value decomposition (HOSVD), Constrained block Tucker decomposition, Electrocorticogram (ECoG), Fusion of behavioral and neural data.

1 INTRODUCTION

The Partial Least Squares (PLS) is a well-established framework for estimation, regression and classification, whose objective is to predict a set of dependent variables (responses) from a set of independent variables (predictors) through the extraction of a small number of latent variables. One member of the PLS family is Partial Least Squares Regression (PLSR) - a multivariate method which, in contrast to Multiple Linear Regression (MLR) and Principal Component Regression (PCR), is proven to be particularly suited to highly collinear data [1], [2]. In order to predict response variables \( Y \) from independent variables \( X \), PLS finds a set of latent variables (also called latent vectors, score vectors or components) by projecting both \( X \) and \( Y \) onto a new subspace, while at the same time maximizing the pairwise covariance between the latent variables of \( X \) and \( Y \). A standard way to optimize the model parameters is the Non-linear Iterative Partial Least Squares (NIPALS) [3], for an overview of PLS and its applications in neuroimaging see [4], [5], [6]. There are many variations of the PLS model including the orthogonal projection on latent structures (O-PLS) [7], Biorthogonal PLS (BPLS) [8], recursive partial least squares (RPLS) [9], nonlinear PLS [10], [11]. The PLS regression is known to exhibit high sensitivity to noise, a problem that can be attributed to redundant latent variables [12], whose selection still remains an open problem [13]. Penalized regression methods are also popular for simultaneous variable selection and coefficient estimation, which impose e.g., L2 or L1 constraints on the regression coefficients. Algorithms of this kind are Ridge regression and Lasso [14]. The recent progress in sensor technology, biomedicine, and biochemistry has highlighted the necessity to consider multiple data streams as multi-way data structures [15], for which the corresponding analysis methods are very naturally based on tensor decompositions [16], [17], [18]. Although matricization of a tensor is an alternative way to express such data, this would result in the “Large \( p \) Small \( n \)” problem and also make it difficult to interpret the results, as the physical meaning and multi-way data structures would be lost due to the unfolding operation.

The \( N \)-way PLS (N-PLS) decomposes the independent and dependent data into rank-one tensors, subject to maximum pairwise covariance of the latent vectors. This promises enhanced stability, resilience to noise, and intuitive interpretation of the results [19], [20]. Owing to these desirable properties N-PLS has found applications in areas ranging from chemometrics [21], [22], [23] to neuroscience [24], [25]. A modification of the N-PLS and the multi-way covariates regression were studied in [26]....
where the weight vectors yielding the latent variables are optimized by the same strategy as in N-PLS, resulting in better fitness to independent data $X$ while maintaining no difference in predictive performance. The tensor decomposition utilized within N-PLS is Canonical Decomposition /Parallel Factor Analysis (CANDECOMP/PARAFAC or CP) \[29\], which makes N-PLS inherit both the advantages and limitations of CP \[30\]. These limitations are related to poor fitness ability, computational complexity and slow convergence when handling multivariate dependent data and higher order ($N > 3$) independent data, causing N-PLS not to be guaranteed to outperform standard PLS \[23\], \[31\].

In this paper, we propose a new generalized multilinear regression model, called Higher-Order Partial Least Squares (HOPLS), which makes it possible to predict an $M$th-order tensor $Y$ ($M \geq 3$) (or a particular case of two-way matrix $Y$) from an $N$th-order tensor $X(N \geq 3)$ by projecting tensor $X$ onto a low-dimensional common latent subspace. The latent subspaces are optimized sequentially through simultaneous rank-$\{1, L_2, \ldots, L_N\}$ approximation of $X$ and rank-$\{1, K_2, \ldots, K_M\}$ approximation of $Y$ (or rank-one approximation in particular case of two-way matrix $Y$). Owing to the better fitness ability of the orthogonal Tucker model as compared to CP \[16\], the analysis and simulations show that HOPLS proves to be a promising multilinear subspace regression framework that provides not only an optimal tradeoff between fitness and model complexity but also enhanced predictive ability in general. In addition, we develop a new strategy to find a closed-form solution by employing higher-order singular value decomposition (HOSVD) \[33\], which makes the computation more efficient than the classical iterative procedure.

The article is structured as follows. In Section 2 an overview of two-way PLS is presented, and the notation and notions related to multi-way data analysis are introduced. In Section 3 the new multilinear regression model is proposed, together with the corresponding solutions and algorithms. Extensive simulations on synthetic data and a real world case study on the fusion of behavioral and neural data are presented in Section 4 followed by conclusions in Section 5.

## 2 Background and Notation

### 2.1 Notation and definitions

$N$th-order tensors (multi-way arrays) are denoted by underlined boldface capital letters, matrices (two-way arrays) by boldface capital letters, and vectors by boldface lower-case letters. The $i$th entry of a vector $x$ is denoted by $x_i$, and element $(i_1, i_2, \ldots, i_N)$ of an $N$th-order tensor $X \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by $x_{i_1i_2\ldots i_N}$ or $(X)_{i_1i_2\ldots i_N}$. Indices typically range from 1 to their capital version, e.g., $i_N = 1, \ldots, I_N$. The mode-$n$ matricization of a tensor is denoted by $X_{(n)} \in \mathbb{R}^{I_n \times I_{n-1} \cdots I_{n-1} I_{n+1} \cdots I_N}$. The $n$th factor matrix in a sequence is denoted by $A^{(n)}$.

The $n$-mode product of a tensor $X \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and matrix $A \in \mathbb{R}^{I_n \times I_{n'}}$ is denoted by $Y = X \circ A \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} I_{n} I_{n+1} \times \cdots \times I_N}$ and is defined as:

$$y_{i_1i_2\ldots i_{n-1}i_{n+1}\ldots i_N} = \sum_{i_n} x_{i_1i_2\ldots i_n \cdots i_N} a_{j_{n}i_n}, \quad (1)$$

The rank-$\{R_1, R_2, \ldots, R_N\}$ Tucker model \[34\] is a tensor decomposition defined and denoted as follows:

$$Y \approx g \circ \sum_{n=1}^{R} \lambda_n a_n^{(1)} \circ a_n^{(2)} \circ \cdots \circ a_n^{(N)}, \quad (3)$$

where the symbol ‘$\circ$’ denotes the outer product of vectors, $a_n^{(n)}$ is the column-$n$ vector of matrix $A^{(n)}$, and $\lambda_n$ are the singular values. The CP model \[16\], \[29\], \[36\], \[37\], \[38\] became prominent in Chemistry \[28\] and is defined as a sum of rank-one tensors:

$$Y \approx \sum_{r=1}^{R} \lambda_r a^{(1)}_r \circ a^{(2)}_r \cdots \circ a^{(N)}_r, \quad (3)$$

where $a^{(n)}_r$ denotes the outer product of column vectors of $A^{(n)}$, and $\lambda_r$ are scalars. The CP model can also be represented by CP under the condition that the core tensor is super-diagonal, i.e., $R_1 = \cdots = R_N$ and $g_{i_1i_2\cdots i_N} = 0$ if $i_n \neq i_m$ for all $n \neq m$.

The 1-mode product between $G \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $t \in \mathbb{R}^{I_1 \times 1}$ is of size $I_1 \times I_2 \times \cdots \times I_N$, and is defined as:

$$(G \times t)_{i_1i_2\ldots i_N} = g_{i_1i_2\ldots i_N} t_{i_1}, \quad (4)$$

The inner product of two tensors $A, B \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is defined by $\langle A, B \rangle = \sum_{i_1, \ldots, i_N} a_{i_1\ldots i_N} b_{i_1\ldots i_N}$, and the squared Frobenius norm by $\|A\|_F^2 = \langle A, A \rangle$.

The $n$-mode cross-covariance between an $N$th-order tensor $X \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and an $M$th-order tensor $Y \in \mathbb{R}^{J_1 \times \cdots \times J_M}$ with the same size $I_n$ on the $n$th-mode, denoted by $\text{COV}_{(n;n)}(X, Y) \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} I_{n+1} \cdots \times I_N J_1 \cdots J_{n-1} I_{n+1} \cdots \times J_M}$, is defined as:

$$\text{C} = \text{COV}_{(n;n)}(X, Y) = \langle X, Y \rangle_{(n;n)} \quad (5)$$

where the symbol $\langle \cdot, \cdot \rangle_{(n;n)}$ represents an $n$-mode multiplication between two tensors, and is defined as:

$$c_{i_1\ldots i_{n-1}i_{n+1}\ldots i_{n-1}j_1\ldots j_{n-1}j_n\ldots j_M} = \sum_{i_n=1}^{I_n} x_{i_1i_2\ldots i_n \cdots i_N} y_{j_1i_2\ldots j_n, i_{n+1}\ldots i_M}, \quad (6)$$

As a special case, for a matrix $Y \in \mathbb{R}^{I_n \times M}$, the $n$-mode cross-covariance between $X$ and $Y$ simplifies as:

$$\text{COV}_{(n;1)}(X, Y) = X \times_n Y^T, \quad (7)$$
under the assumption that \( n \)-mode column vectors of \( X \) and columns of \( Y \) are mean-centered.

### 2.2 Standard PLS (two-way PLS)

\[
X = T P^T + E = \sum_{r=1}^{R} t_r p_r^T + E, \quad \text{and} \quad Y = U Q^T + F = \sum_{r=1}^{R} u_r q_r^T + F,
\]

where \( T \) and \( U \) are \( R \times J \) matrices of \( t_r \) and \( u_r \) respectively. The matrices \( P \) and \( Q \) represent loadings and \( E, F \) are respectively the residuals for \( X \) and \( Y \). In order to find the first set of components, the classical PLS algorithm is to optimize the two sets of weights \( w, q \) so as to satisfy

\[
\max_{\{w, q\}} [w^T X^T Y q]^2, \quad \text{s.t.} \quad w^T w = 1, q^T q = 1.
\]

The latent variables are then given by \( t = Xw / \|Xw\| \) and \( u = Yq \). Here, two assumptions are made: i) the latent variables \( \{t_r\}_{r=1}^R \) are good predictors of \( Y \); ii) a linear inner relation between the latent variables \( t \) and \( u \) exists; i.e., \( U = TD + Z \) where \( D \) is a linear relation diagonal matrix and \( Z \) denotes the matrix of Gaussian i.i.d. residuals. The maximum likelihood estimate of parameters \( D \) is \( D = (T^T T)^{-1} T^T U \). Upon combining it with the decomposition of \( Y \) matrix, \( Y = TDQ^T + (ZQ^T + F) \) can be written as

\[
Y = TDQ^T + (ZQ^T + F) \approx TDQ^T,
\]

where \( ZQ^T + F \) is the residual matrix. Thus \( Y = TDQ^T \) indicates that the problem boils down to finding common latent variables \( T \) that explain the variance of both \( X \) and \( Y \), as illustrated in Fig. 1.

### 3 Higher-Order PLS (HOPLS)

In the case of a two-way matrix, column rank and row rank are equal, and correspond to the minimal number of rank-one terms into which the matrix can be decomposed, and thus the low-rank approximation is equivalent to subspace approximation. However, for a higher-order tensor, these criteria lead to two typical tensor decompositions (i.e., CP and Tucker model). The CP decomposes a tensor as a sum of rank-one tensors approximating a higher-order tensor, these criteria lead to two typical tensor decompositions (i.e., CP and Tucker model). The CP decomposes a tensor as a sum of rank-one tensors, while the Tucker model decomposes a tensor into a core tensor multiplied by matrices along each mode \( \{\text{mode}\} \), which can be considered as a low-rank approximation, while the Tucker model decomposes a tensor into a core tensor multiplied by matrices along each mode \( \{\text{mode}\} \), which is a rank-(\( R_1, R_2, R_3 \)) subspace approximation. Both the CP and Tucker decompositions can be regarded as higher-order generalizations of the matrix singular value decomposition (SVD) and principal component analysis (PCA). In fact, CP can be viewed as a special case of Tucker where the core tensor is superdiagonal. Block term decompositions unify the Tucker and CP decompositions into one framework by decomposing a tensor into a sum of rank-(\( R_1, R_2, R_3 \)) terms \( \{\text{mode}\} \).

Consider an \( N \)-order independent tensor \( X \in \mathbb{R}^{I_1 \times \cdots \times I_N} \) and an \( M \)-order dependent tensor \( Y \in \mathbb{R}^{J_1 \times \cdots \times J_M} \), having the same size on the first mode, i.e., \( I_1 = J_1 \). Our objective is to find an optimal subspace approximation of \( X \) and \( Y \), in which the latent vectors from \( X \) and \( Y \) have maximum pairwise covariance. Considering a linear relation between the latent vectors, the problem boils down to finding the common latent subspace which can approximate both \( X \) and \( Y \) simultaneously. We firstly address the general case of a tensor.
$\mathbf{X}(N \geq 3)$ and a tensor $\mathbf{Y}(M \geq 3)$. A particular case with a tensor $\mathbf{X}(N \geq 3)$ and a matrix $\mathbf{Y}(M = 2)$ is presented separately in Sec 3.3 using a slightly different approach.

### 3.1 Proposed model

Applying Tucker decomposition within a PLS framework is not straightforward, and to that end we propose a novel block-wise orthogonal Tucker approach to model the data. More specifically, we assume $\mathbf{X}$ is decomposed as a sum of rank-$(1, L_2, \ldots, L_N)$ Tucker blocks, while $\mathbf{Y}$ is decomposed as a sum of rank-$(1, K_2, \ldots, K_M)$ Tucker blocks (see Fig. 2), which can be expressed as

$$
\mathbf{X} = \sum_{r=1}^{R} \mathbf{G}_r \times_1 t_r \times_2 P^{(1)}_r \times_3 \cdots \times_N P^{(N-1)}_r + \mathbf{E}_R,
$$

$$
\mathbf{Y} = \sum_{r=1}^{D} \mathbf{D}_r \times_1 t_r \times_2 Q^{(1)}_r \times_3 \cdots \times_M Q^{(M-1)}_r + \mathbf{E}_R, \quad (12)
$$

where $R$ is the number of latent vectors, $t_r \in \mathbb{R}^{I_1}$ is the $r$-th latent vector, $\{P^{(n)}_r\}_{r=1}^{N-1} \in \mathbb{R}^{I_{n+1} \times K_{n+1}}$, and $\{Q^{(m)}_r\}_{r=1}^{M-1} \in \mathbb{R}^{J_{m+1} \times K_{m+1}}$ are loading matrices on mode-$n$ and mode-$m$ respectively, and $\mathbf{G}_r \in \mathbb{R}^{I_1 \times L_2 \times \cdots \times L_N}$ and $\mathbf{D}_r \in \mathbb{R}^{K_1 \times K_2 \times \cdots \times K_M}$ are core tensors.

However, the Tucker decompositions in (12) are not unique due to the permutation, rotation, and scaling issues. To alleviate this problem, additional constraints should be imposed such that the core tensors $\mathbf{G}_r$ and $\mathbf{D}_r$ are all-orthogonal, a sequence of loading matrices are column-wise orthonormal, i.e., $P^{(n)}_r \mathbf{P}^{(n)T}_r = \mathbf{I}$ and $Q^{(m)}_r \mathbf{Q}^{(m)T}_r = \mathbf{I}$, the latent vector is of length one, i.e. $\|t_r\|_F = 1$. Thus, each term in (12) is represented as an orthogonal Tucker model, implying essentially uniqueness as it is subject only to trivial indeterminacies [2].

By defining a latent matrix $\mathbf{T} = [t_1, \ldots, t_R]$ mode-$n$ loading matrix $\mathbf{P}^{(n)} = [P^{(n)}_1, \ldots, P^{(n)}_R]$, mode-$m$ loading matrix $\mathbf{Q}^{(m)} = [Q^{(m)}_1, \ldots, Q^{(m)}_M]$ and core tensor $\mathbf{G} = \text{blockdiag}(\mathbf{G}_1, \ldots, \mathbf{G}_R) \in \mathbb{R}^{R1 \times L_2 \times \cdots \times L_N}$ and core tensor $\mathbf{D} = \text{blockdiag}(\mathbf{D}_1, \ldots, \mathbf{D}_R) \in \mathbb{R}^{K_1 \times K_2 \times \cdots \times K_M}$, the HOPLS model in (12) can be rewritten as

$$
\mathbf{X} = \mathbf{G} \times_1 \mathbf{T} \times_2 \mathbf{P}^{(1)} \times_3 \cdots \times_N \mathbf{P}^{(N-1)} + \mathbf{E}_R,
$$

$$
\mathbf{Y} = \mathbf{D} \times_1 \mathbf{T} \times_2 \mathbf{Q}^{(1)} \times_3 \cdots \times_M \mathbf{Q}^{(M-1)} + \mathbf{E}_R, \quad (13)
$$

where $\mathbf{E}_R$ and $\mathbf{E}_R$ are residuals after extracting $R$ components. The core tensors $\mathbf{G}$ and $\mathbf{D}$ have a special block-diagonal structure (see Fig. 2) and their elements indicate the level of local interactions between the corresponding latent vectors and loading matrices. Note that the tensor decomposition in (13) is similar to the block term decomposition discussed in [32], which aims to the decomposition of only one tensor. However, HOPLS attempts to find the block Tucker decompositions of two tensors with block-wise orthogonal constraints, which at the same time satisfies a certain criteria of sharing the common latent components on a specific mode.

Benefiting from the advantages of Tucker decomposition over the CP model [16], HOPLS promises to approximate data better than N-PLS. Specifically, HOPLS differs substantially from the N-PLS model in the sense that extraction of latent components in HOPLS is based on subspace approximation rather than on low-rank approximation and the size of loading matrices is controlled by a hyperparameter, providing a tradeoff between fitness and model complexity. Note that HOPLS simplifies into N-PLS if we define $\forall n : \{L_n\} = 1$ and $\forall m : \{K_m\} = 1$.

### 3.2 Optimization criteria and algorithm

There are two different approaches for extracting the latent components: sequential and simultaneous methods. A sequential method extracts one latent component at a time, deflates the proper tensors and calculates the next component from the residuals. In a simultaneous method, all components are calculated simultaneously by minimizing a certain criterion. In the following, we employ a sequential method.

The tensor decompositions in (12) can be represented as an optimization problem of approximating $\mathbf{X}$ and $\mathbf{Y}$ by orthogonal Tucker model, while having a common latent component on a specific mode. If we apply HOSVD individually on $\mathbf{X}$ and $\mathbf{Y}$, the best rank-(1, $L_2, \ldots, L_N$) approximation for $\mathbf{X}$ and the best rank-(1, $K_2, \ldots, K_M$) approximation for $\mathbf{Y}$ can be obtained while the common latent vector $t_r$ cannot be ensured. Another way is to find the best approximation of $\mathbf{X}$ by HOSVD first, subsequently, $\mathbf{Y}$ can be approximated by a fixed $t_r$. However, this procedure, which resembles multi-way principal component regression [28], has the drawback that the common latent components are not necessarily predictive for $\mathbf{Y}$.

The optimization of subspace transformation according to (12) will be formulated as a problem of determining a set of orthogonal loadings $P^{(n)}$, $Q^{(m)}$, $r = 1, 2, \ldots, R$ and latent vectors $t_r$ that satisfies a certain criterion. Since each term can be optimized sequentially with the same criteria based on deflation, in the following, we shall simplify the problem to that of finding the first latent vector $t$ and two sequences of loading matrices $P^{(n)}$ and $Q^{(m)}$.

In order to develop a strategy for the simultaneous minimization of the Frobenius norm of residuals $\mathbf{E}$ and $\mathbf{F}$, while keeping a common latent vector $t$, we first need to introduce the following basic results:

**Proposition 3.1.** Given a tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and column orthornormal matrices $P^{(n)} \in \mathbb{R}^{I_{n+1} \times F_{n+1}}$, $n = 1, \ldots, N - 1$, $t \in \mathbb{R}^{I_1}$ with $\|t\|_F = 1$, the least-squares (LS) solution to min $\|\mathbf{X} - \mathbf{G} \times_1 t \times_2 P^{(1)} \times_3 \cdots \times_N P^{(N-1)}\|_F^2$ is given by $\mathbf{G} = \mathbf{X} \times_1 t^T \times_2 P^{(1)T} \times_3 \cdots \times_N P^{(N-1)T}$.

**Proof:** This result is very well known and is widely used in the literature [16], [33]. A simple proof is based on writing the mode-1 matricization of tensor
X as
\[ X_{(1)} = tG_{(1)}(P^{(N-1)} \otimes \cdots \otimes P^{(1)})^T + E_{(1)}, \] (14)
where tensor \( E_{(1)} \) is the residual and the symbol ‘\( \otimes \)’ denotes the Kronecker product. Since \( t^T t = 1 \) and \((P^{(N-1)} \otimes \cdots \otimes P^{(1)})\) is column orthonormal, the LS solution of \( G_{(1)} \) with fixed matrices \( t \) and \( D^{(n)} \) is given by \( G_{(1)} = t^T X_{(1)}^2 \otimes \cdots \otimes \otimes P^{(1)}) \); writing it in a tensor form we obtain the desired result.

**Proposition 3.2.** Given a tensor \( X \in R^{I_1 \times \cdots \times I_N} \), the following two constrained optimization problems are equivalent:

1) \( \text{min}_{\{P^{(n)}, t \in G \}} \| X - G \times_1 t \times_2 P^{(1)} \times_3 \cdots \times_N P^{(N-1)} \| F \),
   s. t. matrices \( P^{(n)} \) are column orthonormal and \( \| t \|_F = 1 \).

2) \( \max_{\{P^{(n)}, t \}} \| X \times_1 t \times_2 P^{(1)} \times_3 \cdots \times_N P^{(N-1)} \| F \),
   s. t. matrices \( P^{(n)} \) are column orthonormal and \( \| t \|_F = 1 \).

The proof is available in [16] (see pp. 477-478).

Assume that the orthonormal matrices \( P^{(n)}, Q^{(m)} \), \( t \) are given, then from Proposition 3.1 the core tensors in (12) can be computed as
\[
G = X \times_1 t^T \times_2 P^{(1)} \times_3 \cdots \times_N P^{(N-1)}T \\
D = Y \times_1 t^T \times_2 Q^{(1)} \times_3 \cdots \times_M Q^{(M-1)}T. \tag{15}
\]
According to Proposition 3.2, the minimization of \( \| E \|_F \) and \( \| F \|_F \) under the orthonormality constraint is equivalent to maximization of \( \| G \|_F \) and \( \| D \|_F \).

However, there is no straightforward tensor decomposition method to maximize \( \| G \|_F \) and \( \| D \|_F \) simultaneously over the factor matrices \( \{P^{(n)}\}_{n=1}^{N-1} \) and \( \{Q^{(m)}\}_{m=1}^{M-1} \) and a common latent vector \( t \). This leads to the proposed method to maximize a product of norms of two core tensors, i.e., \( \max \|G\|_F^2, \|D\|_F^2 \). Since the latent vector \( t \) is determined by \( P^{(n)}, Q^{(m)} \), the first step is to optimize the orthonormal loadings, then the common latent vectors can be computed by the fixed loadings.

**Proposition 3.3.** Let \( G \in R^{I_1 \times I_2 \times \cdots \times I_N} \) and \( D \in R^{I_1 \times K_2 \times \cdots \times K_M} \), then \( \| G \times_1 t \times_2 P^{(1)} \times_3 \cdots \times_N P^{(N-1)} \|_F^2 = \| G \|_F^2 \times \| D \|_F^2 \).

Proof: \( \| G \times_1 t \times_2 P^{(1)} \times_3 \cdots \times_N P^{(N-1)} \|_F = \| \text{vec}(G) \text{vec}(t)^T \|_F \)
\[
= \text{trace}(\text{vec}(D) \text{vec}(G) \text{vec}(G)^T \text{vec}(D)^T) \\
= \| \text{vec}(G) \|_F^2 \times \| \text{vec}(D) \|_F^2. \tag{16}
\]
where \( \text{vec}(G) \in R^{I_1 \times K_2 \times \cdots \times L_N} \) is the vectorization of the tensor \( G \).

From Proposition 3.3 observe that to maximize \( \| G \|_F^2 \times \| D \|_F^2 \) is equivalent to minimizing \( \| G \times_1 t \times_2 P^{(1)} \times_3 \cdots \times_N P^{(N-1)} \|_F^2 \). According to (15) and \( t^T t = 1 \), \( \| G \times_1 t \times_2 P^{(1)} \times_3 \cdots \times_N P^{(N-1)} \|_F^2 \) can be expressed as
\[
\| \langle X, Y \rangle_{(1:1)} : P^{(1)} \times_3 \cdots \times_2 P^{(N-1)} \times_1 P^{(N)} \|_F^2. \tag{17}
\]
Note that this form is quite similar to the optimization problem for two-way PLS in (10), where the cross-covariance matrix \( X^T Y \) is replaced by \( \langle X, Y \rangle_{(1:1)} \). In addition, the optimization item becomes the norm of a small tensor in contrast to a scalar in (10). Thus, if we define \( \langle X, Y \rangle_{(1:1)} \) as a mode-1 cross-covariance tensor \( C = \text{COV}_{(1:1)}(X, Y) \in R^{I_2 \times \cdots \times I_N \times I_2 \times \cdots \times I_M} \), the optimization problem can be finally formulated as
\[
\max \\{P^{(n)}, Q^{(m)}\} \quad \| [C : P^{(1)} : \cdots : P^{(N-1)} : Q^{(1)} : \cdots : Q^{(M-1)}] \|_F^2
\]
s. t. \( P^{(n)}^T P^{(n)} = I_{K_n+1}, Q^{(m)}^T Q^{(m)} = I_{K_m+1} \), (18)
where \( P^{(n)}, n = 1, \ldots, N-1 \) and \( Q^{(m)}, m = 1, \ldots, M-1 \) are the parameters to optimize.

Based on Proposition 3.2 and orthogonality of \( P^{(n)}, Q^{(m)} \), the optimization problem in (18) is equivalent to find the best subspace approximation of \( C \) as
\[
C \approx [C^{(C)} : P^{(1)} : \cdots : P^{(N-1)} : Q^{(1)} : \cdots : Q^{(M-1)}], \tag{19}
\]
which can be obtained by rank-(\(L_2, \ldots, L_N, K_2, \ldots, K_M\)) HOSVD on tensor \( C \). Based on Proposition 3.3, the optimization term in (18) is equivalent to the norm of core tensor \( C^{(C)} \). To achieve this goal, the higher-order orthogonal iteration (HOOI) algorithm [16], [37], which is known to converge fast, is employed to find the parameters \( P^{(n)} \) and \( Q^{(m)} \) by orthogonal Tucker decomposition of \( C^{(C)} \).

Subsequently, based on the estimate of the loadings \( P^{(n)} \) and \( Q^{(m)} \), we can now compute the common latent vector \( t \). Since our goal is to predict \( Y \) from \( X \), similarly to the PLS method, we need to estimate \( t \) from predictors \( X \) and also to estimate the regression coefficient \( D \) in order to predict responses \( Y \). For a given set of loading matrices \( \{P^{(n)}\} \), the latent vector \( t \) should explain variance of \( X \) as much as possible, that is
\[
t = \arg \min \| X - [G ; t, P^{(1)} : \cdots : P^{(N-1)}] \|_F^2 , \tag{20}
\]
which can be easily achieved by choosing \( t \) as the first leading left singular vector of the matrix \( X \times_1 P^{(1)} \times_2 \cdots \times_1 P^{(N-1)} \times_1 (1) \) in the HOOI algorithm (see [16], [35]). Thus, the core tensors \( G \) and \( D \) are computed by (15).

The above procedure should be carried out repeatedly using the deflation operation, until an appropriate number of components (i.e., \( R \)) are obtained, or the norms of residuals are smaller than a certain threshold. The deflation is performed by subtracting from \( X \) and \( Y \) the information explained by a rank-(1, \( L_2, \ldots, L_N \)) tensor \( \tilde{X} \) and a rank-(1, \( K_2, \ldots, K_M \)) tensor \( \tilde{Y} \), respectively. The HOPLS algorithm is outlined in Algorithm 4.

### 3.3 The case of the tensor X and matrix Y
Consider an \( N \)-th order independent tensor \( X \) \( \in R^{I_1 \times \cdots \times I_N} \) \( (N \geq 3) \) and a two-way dependent data \( Y \) \( \in R^{I_1 \times M} \), with the same sample size \( I_1 \). The HOPLS models independent data \( X \) as a sum of rank-(1, \( L_2, \ldots, L_N \)) and also to estimate the regression coefficient
Algorithm 1 The Higher-order Partial Least Squares (HOPLS) Algorithm for a Tensor \( \mathbf{X} \) and a Tensor \( \mathbf{Y} \)

**Input:** \( \mathbf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N} \), \( \mathbf{Y} \in \mathbb{R}^{J_1 \times \cdots \times J_M} \), \( N \geq 3, M \geq 3 \) and \( I_1 = J_1 \).

Number of latent vectors is \( R \) and number of loading vectors are \( \{L_r\}_{r=1}^R \) and \( \{K_m\}_{m=1}^M \).

**Output:** \( \{\mathbf{P}_r^{(o)}\}; \{\mathbf{Q}_r^{(m)}\}; \{\mathbf{D}_r\}; \{\mathbf{F}_r\}; \mathbf{T} \)

\( r = 1, \ldots, R; \) \( n = 1, \ldots, N - 1; \) \( m = 1, \ldots, M - 1. \)

**Initialization:** \( \mathbf{E}_0 \leftarrow \mathbf{X} \), \( \mathbf{F}_1 \leftarrow \mathbf{Y} \)

for \( r = 1 \) to \( R \) do

if \( \|\mathbf{E}_r\|_F > \varepsilon \) and \( \|\mathbf{F}_r\|_F > \varepsilon \) then

\( \mathbf{C}_r \leftarrow \mathbf{E}_r \) \( (r = 1,\ldots,1) \)

Rank-\( (L_2,\ldots,L_N,K_2,\ldots,K_M) \) orthogonal Tucker decomposition of \( \mathbf{C}_r \) by HOOT \( [16] \) as
\( \mathbf{C}_r \approx \left[ \mathbf{G}_r; \mathbf{P}_r^{(1)}; \mathbf{P}_r^{(2)}; \mathbf{Q}_r^{(3)} \right] \)

\( t_r \leftarrow \) the first leading left singular vector by SVD (13)

\( \mathbf{G}_r \leftarrow \left[ \mathbf{E}_r \times_{1} \mathbf{P}_r^{(1)} \times_{2} \mathbf{P}_r^{(2)} \times_{3} \mathbf{P}_r^{(3)} \right] \)

\( \mathbf{D}_r \leftarrow \left[ \mathbf{F}_r \times_{1} \mathbf{Q}_r^{(1)} \times_{2} \mathbf{Q}_r^{(2)} \times_{3} \mathbf{Q}_r^{(3)} \right] \)

Deflation:

\( \mathbf{E}_{r+1} \leftarrow \mathbf{E}_r - \mathbf{G}_r \mathbf{t}_r \mathbf{P}_r^{(1)} \mathbf{P}_r^{(2)} \mathbf{P}_r^{(3)} \)

\( \mathbf{F}_{r+1} \leftarrow \mathbf{F}_r - \mathbf{D}_r \mathbf{t}_r \mathbf{Q}_r^{(1)} \mathbf{Q}_r^{(2)} \mathbf{Q}_r^{(3)} \)

else Break;
end if

end for

The problem is \( \mathbf{u} = \mathbf{Yq}(q^Tq)^{-1} \), and hence, \( \mathbf{u} = \mathbf{Yq} \)
gives the best fit of \( \mathbf{Y} \) for that \( q \).

The sequential deflation type optimization is employed for the model parameters. Observe that according to Proposition 3.4, the optimal fit is achieved for \( \mathbf{u} = \mathbf{Yq} \). Thus, combining the linear relation between \( \mathbf{t} \) and \( \mathbf{u} \) with the expression for the core tensor \( \mathbf{G} \) in (15), we can optimize the parameters of X-loading matrices \( \mathbf{P}^{(o)} \) and Y-loading vector \( q \) via

\[
\max \| \mathbf{X} \times_1 \mathbf{Y}^T \times_1 \mathbf{q}^T \times_2 \mathbf{P}_1^{(1)t} \times_3 \cdots \times_N \mathbf{P}_N^{(N-1)t} \|^2_F
\]

subject to \( \mathbf{P}_r^{(s)}q_r = I, \|q_r\|_F = 1 \).

This form is similar to (15), but has a different cross-covariance tensor \( \mathbf{C} = \mathbf{X} \times_1 \mathbf{Y}^T \) defined between a tensor and a matrix, implying that the problem can be solved by performing a rank-\( (1,L_2,\ldots,L_N) \) HOSVD on \( \mathbf{C} \), which also makes it possible to compute the core tensor \( \mathbf{G}^{(C)} \) corresponding to \( \mathbf{C} \).

Subsequently, the loading vector \( \mathbf{t} \) is estimated from \( \mathbf{X} \) using the given loading matrices \( \mathbf{P}^{(o)} \). The mode-1 matricization of the model for \( \mathbf{X} \) allows us to write

\[
\mathbf{X}_{(1)} = \mathbf{tG}^{(1)}(\mathbf{P}^{(N-1)t} \otimes \cdots \otimes \mathbf{P}^{(1)t}) + \mathbf{E}^{(1)}
\]

where \( \mathbf{G}^{(1)} \in \mathbb{R}^{I_1 \times 1 \times \cdots \times \mathbb{R}^{I_N} \times L_2 \times \cdots \times L_N} \), \( \mathbf{P}^{(o)} \) is still remains unknown. However, owing to the linear relationship between the core tensor \( \mathbf{G}^{(o)} \) of [\( (\mathbf{X}; \mathbf{t}; \mathbf{P}^{(1)t}; \mathbf{P}^{(2)t}; \mathbf{P}^{(3)t}) \] and the core tensor \( \mathbf{G}^{(C)} \) of [\( (\mathbf{C}; \mathbf{q}; \mathbf{P}^{(1)t}; \mathbf{P}^{(2)t}; \mathbf{P}^{(3)t}) \] so that \( \mathbf{G}^{(C)} = d\mathbf{G} \), a least square solution for the normalized \( \mathbf{t} \), which minimizes the squared norm of the residual \( \|\mathbf{E}^{(1)}\|_F^2 \), can be obtained from

\[
\mathbf{t} \leftarrow (\mathbf{X} \times_2 \mathbf{P}^{(1)t} \times_3 \cdots \times_N \mathbf{P}^{(N-1)t}) \mathbf{G}^{(C)+}, \mathbf{t} \leftarrow \mathbf{t}/\|\mathbf{t}\|_F
\]

where we used the property that \( \mathbf{P}^{(o)} \) is columnwise orthonormal and the symbol '*' denotes Moore-Penrose pseudoinverse. Based on the estimated latent vector \( \mathbf{t} \) of length one and loadings \( \mathbf{q} \), the regression coefficient used to predict \( \mathbf{Y} \) is computed as

\[
d = \mathbf{t}^T \mathbf{u} = \mathbf{t}^T \mathbf{Yq}
\]

The procedure for a two-way response matrix is summarized in Algorithm 2. In this case, the HOPLS model is shown to unify both the standard PLS and N-PLS, when the appropriate parameters \( L_n \) are selected.

### 3.4 Prediction of the Response Variables

Predictions from the new observations \( \mathbf{X}_{new} \) are performed in two steps: projecting the data to the low-dimensional latent space based on model parameters \( \mathbf{G}_r; \mathbf{P}_r^{(o)} \), and predicting the response data based on latent vectors \( \mathbf{T}_{new} \) and model parameters \( \mathbf{Q}_r^{(m)}; \mathbf{D}_r \). For simplicity, we use a matricized form to express the prediction procedure as

\[
\mathbf{Y}_{new} \approx \mathbf{T}_{new} \mathbf{Q}^T = \mathbf{X}_{new}^{(1)} \mathbf{WQ}^T
\]

2. Explanation and proof are given in the supplement material.
Algorithm 2 Higher-order Partial Least Squares (HOPLS)
for a Tensor $\mathbf{X}$ and a Matrix $\mathbf{Y}$

**Input:** $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, $N \geq 3$ and $\mathbf{Y} \in \mathbb{R}^{I_1 \times M}$

The number of latent vectors is $R$ and the number of loadings are $L_n$, $n=2,...,N$.

**Output:** $\{\mathbf{P}_r^{(n)}\}$; $\mathbf{Q}$: $\{\mathbf{G}_r\}$: $\mathbf{D}$; $\mathbf{T}$; $r=1,...,R$, $n=2,...,N$.

**Initialization:** $\mathbf{E}_0 \leftarrow \mathbf{X}$, $\mathbf{F}_1 \leftarrow \mathbf{Y}$.

for $r=1$ to $R$ do
  if $\|\mathbf{E}_r\|_F > \varepsilon$ and $\|\mathbf{F}_r\|_F > \varepsilon$ then
    $\mathbf{C}_r \leftarrow \mathbf{E}_r \times_1 \mathbf{F}_r^T$;
    Perform rank-$1$ ($L_1, L_2, \ldots, L_N$) HOOI on $\mathbf{C}_r$ as
    $\mathbf{C}_r \approx \mathbf{G}_r^{(C)}_r \times_1 q_r \times_2 \mathbf{P}_r^{(1)} \times_3 \cdots \times_N \mathbf{P}_r^{(N-1)}$;
    $\mathbf{t}_r \leftarrow \frac{\mathbf{E}_r \times_2 \mathbf{P}_r^{(1)} \times_3 \cdots \times_N \mathbf{P}_r^{(N-1)}}{(\mathbf{G}_r^{(C)})_r^{(1)}}$;
    $\mathbf{G}_r \leftarrow \mathbf{G}_r$. $\mathbf{t}_r$;
    $\mathbf{D}_r \leftarrow \mathbf{Q}_r^{(M-1)} \times_1 \cdots \times_N \mathbf{Q}_r^{(N)}$;
    $\mathbf{u}_r \leftarrow \mathbf{F}_r \mathbf{q}_r$;
    $d_r \leftarrow u_r^T \mathbf{t}_r$;
  else
    Deflation:
    $\mathbf{E}_{r+1} \leftarrow \mathbf{E}_r - [\mathbf{G}_r \mathbf{t}_r, \mathbf{P}_r^{(1)}, \ldots, \mathbf{P}_r^{(N-1)}]$;
    $\mathbf{F}_{r+1} \leftarrow \mathbf{X} - d_r \mathbf{t}_r \mathbf{q}_r^T$
  end if
end for

where $\mathbf{W}$ and $\mathbf{Q}^*$ have $R$ columns, represented by

$\mathbf{w}_r = \left( \mathbf{P}_r^{(N-1)} \otimes \cdots \mathbf{P}_r^{(1)} \right) \mathbf{G}_r^{(r)}$, $r=1,\ldots,R$.

$\mathbf{q}_r^* = \mathbf{D}_r^{(1)} \left( \mathbf{Q}_r^{(M-1)} \otimes \cdots \mathbf{Q}_r^{(N)} \right)^T$.

In the particular case of a two-way matrix $\mathbf{Y}$, the prediction is performed by

$\hat{\mathbf{Y}}_{\text{new}} \approx \mathbf{X}_{\text{new}} \mathbf{W} \mathbf{D}_r \mathbf{Q}_r^T$,

where $\mathbf{D}$ is a diagonal matrix whose entries are $d_r$ and $r$th column of $\mathbf{Q}$ is $\mathbf{q}_r$, $r=1,\ldots,R$.

3.5 Properties of HOPLS

Robustness to noise. An additional constraint of keeping the largest $L_n$, $n=2,...,N$ loading vectors on each mode is imposed in HOPLS, resulting in a flexible model that balances the two objectives of fitness and the significance of associated latent variables. In the case of a single large $L_n$, one may fit $\mathbf{X}$ better but introduces more noise to each latent vector. In contrast, N-PLS is more robust due to the strong constraint of rank-one tensor structure, while lacking good fit to the data. The flexibility of HOPLS allows us to adapt the model complexity based on the dataset in hands, providing considerable prediction ability (see Fig. 3).

"Large $p$, Small $n$" problem. This is particularly important when the dimension of independent variables is high. In contrast to PLS, the relative low dimension of model parameters that need to be optimized in HOPLS. For instance, assume that a 3th-order tensor $\mathbf{X}$ has the dimension of $5 \times 10 \times 100$, i.e., there are 5 samples and 1000 features. If we apply PLS on $\mathbf{X}_{(1)}$ with size of $5 \times 1000$, there are only five samples available to optimize a 1000-dimensional loading vector $\mathbf{p}$, resulting in an unreliable estimate of model parameters. In contrast, HOPLS allows us to optimize loading vectors, having relatively low-dimension, on each mode alternately; thus the number of samples is significantly elevated. For instance, to optimize 10-dimensional loading vectors on the second mode, 500 samples are available, and to optimize the 100-dimensional loading vectors on the third mode there are 50 samples. Thus, a more robust estimate of low-dimensional loading vectors can be obtained, which is also less prone to overfitting and more suitable for "Large $p$, Small $n$" problem (see Fig. 4).

Ease of interpretation. The loading vectors in $\mathbf{P}^{(n)}$ reveal new subspace patterns corresponding to the $n$-mode features. However, the loadings from Unfold-PLS are difficult to interpret since the data structure is destroyed by the unfolding operation and the dimension of loadings is relatively high.

Computation. N-PLS is implemented by combining a NIPALS-like algorithm with the CP decomposition. Instead of using an iterative algorithm, HOPLS can find the model parameters using a closed-form solution, i.e., applying HOSVD on the cross-covariance tensor, resulting in enhanced computational efficiency.

Due to the flexibility of HOPLS, the tuning parameters of $L_n$ and $K_m$, controlling the model complexity, need to be selected based on calibration data. Similarly to the parameter $R$, the tuning parameters can be chosen by cross-validation. For simplicity, two alternative assumptions will be utilized: a) $\forall n, \forall m, L_n = K_m = \lambda_1$; b) $L_n = \eta R_n, K_m = \eta R_m$, $0 < \eta \leq 1$, i.e., explaining the same percentage of the $n$-mode variance.

4 Experimental Results

In the simulations, HOPLS and N-PLS were used to model the data in a tensor form whereas PLS was performed on a mode-1 matricization of the same tensors. To quantify the predictability, the index $Q^2$ was defined as $Q^2 = 1 - \|\hat{\mathbf{Y}} - \mathbf{Y}\|^2 / \|\mathbf{Y}\|^2$, where $\hat{\mathbf{Y}}$ denotes the prediction of $\mathbf{Y}$ using a model created from a calibration dataset. Root mean square errors of prediction (RMSEP) were also used for evaluation [48].

4.1 Synthetic data

In order to quantitatively benchmark our algorithm against the state of the art, an extensive comparative exploration has been performed on synthetic datasets to evaluate the prediction performance under varying conditions with respect to data structure, noise levels and ratio of variable dimension to sample size. For parameter selection, the number of latent vectors ($R$) and number of loadings $(L_n = K_m = \lambda_1)$ were chosen based on five-fold cross-validation on the calibration dataset. To reduce random fluctuations, evaluations were performed over 50 validation datasets generated repeatedly according to the same criteria.
4.1.1 Datasets with matrix structure

The independent data \( X \) and dependent data \( Y \) were generated as:

\[
X = TP^T + \xi E, \quad Y = TQ^T + \xi F, \tag{30}
\]

where latent variables \( \{t, p, q\} \sim \mathcal{N}(0, 1) \), \( E, F \) are Gaussian noises whose level is controlled by the parameter \( \xi \). Both the calibration and the validation datasets were generated according to \( \text{(30)} \), with the same loadings \( P, Q \), but a different latent \( T \) which follows the same distribution \( \mathcal{N}(0, 1) \). Subsequently, the datasets were re-organized as \( N \)-th order tensors.

To investigate how the prediction performance is affected by noise levels and small sample size, \( \{X, Y\} \in \mathbb{R}^{20 \times 10 \times 10} \) (Case 1) and \( \{X, Y\} \in \mathbb{R}^{10 \times 10 \times 10} \) (Case 2) were generated under varying noise levels of 10dB, 5dB, 0dB and -5dB. In the case 3, \( \{X, Y\} \in \mathbb{R}^{10 \times 10 \times 10} \) were generated with the loadings \( P, Q \) drawn from a uniform distribution \( U(0, 1) \). The datasets were generated from five latent variables (i.e., \( T \) has five columns) for all the three cases.

![Fig. 3: Five-fold cross-validation performance of HOPLS at different noise levels versus the number of latent variables (R) and loadings (λ). The optimal values for these two parameters are marked by green squares.](image)

![Fig. 4: The prediction performance comparison among HOPLS, N-PLS and PLS at different noise levels for three cases. Case 1: \( \{X, Y\} \in \mathbb{R}^{20 \times 10 \times 10} \) and \( \{P, Q\} \sim \mathcal{N}(0, 1) \); Case 2: \( \{X, Y\} \in \mathbb{R}^{10 \times 10 \times 10} \) and \( \{P, Q\} \sim \mathcal{N}(0, 1) \); Case 3: \( \{X, Y\} \in \mathbb{R}^{10 \times 10 \times 10} \) and \( \{P, Q\} \sim U(0, 1) \).](image)

<table>
<thead>
<tr>
<th>SNR</th>
<th>PLS</th>
<th>N-PLS</th>
<th>HOPLS</th>
<th>SNR</th>
<th>PLS</th>
<th>N-PLS</th>
<th>HOPLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10dB</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>0dB</td>
<td>3</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>5dB</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>-5dB</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

There are two tuning parameters, i.e., number of latent variables \( R \) and number of loadings \( \lambda \) for HOPLS and only one parameter \( R \) for PLS and N-PLS, that need to be selected appropriately. The number of latent variables \( R \) is crucial to prediction performance, resulting in under-modelling when \( R \) was too small while overfitting easily when \( R \) was too large. The cross-validations were performed when \( R \) and \( \lambda \) were varying from 1 to 10 with the step length of 1. In order to alleviate the computation burden, the procedure was stopped when the performance starts to decrease with increasing \( \lambda \). Fig. 3 shows the grid of cross-validation performance of HOPLS in Case 2 with the optimal parameters marked by green squares. Observe that the optimal \( \lambda \) for HOPLS is related to the noise levels, and for increasing noise levels, the best performance is obtained by smaller \( \lambda \), implying that only few significant loadings on each mode are kept in the latent space. This is expected, due to the fact that the model complexity is controlled by \( \lambda \) to suppress noise. The optimal \( R \) and \( \lambda \) for all three methods at different noise levels are shown in Table 1.

After the selection the parameters, HOPLS, N-PLS and PLS are re-trained on the whole calibration dataset using the optimal \( R \) and \( \lambda \), and were applied to the validation datasets for evaluation. Fig. 4 illustrates the predictive performance over 50 validation datasets for the three cases at four different noise levels. In Case 1, a relatively larger sample size was available, when SNR=10dB, HOPLS achieved a similar prediction performance to PLS while outperforming N-PLS. With increasing the noise level in both the calibration and validation datasets, HOPLS showed a relatively stable performance whereas...
the performance of PLS decreased significantly. The superiority of HOPLS was shown clearly with increasing the noise level. In Case 2 where a smaller sample size was available, HOPLS exhibited better performance than the other two models and the superiority of HOPLS was more pronounced at high noise levels, especially for SNR<5dB. These results demonstrated that HOPLS is more robust to noise in comparison with N-PLS and PLS. If we compare Case 1 with Case 2 at different noise levels, the results revealed that the superiority of HOPLS over the other two methods was enhanced in Case 2, illustrating the advantage of HOPLS in modeling datasets with small sample size. Note that N-PLS also showed better performance than PLS when SNR≤0dB in Case 2, demonstrating the advantages of the model in dataset in a tensor form for small sample sizes. In Case 3, N-PLS showed much better performance as compared to its performance in Case 1 and Case 2, implying sensitivity of N-PLS to data distribution. With the increasing noise level, both HOPLS and N-PLS showed enhanced predictive abilities over PLS.

4.1.2 Datasets with tensor structure

Note that the datasets generated by (30) do not originally possess multi-way data structures although they were organized in a tensor form, thus the structure information of data was not important for prediction. We here assume that HOPLS is more suitable for the datasets which originally have multi-way structure, i.e., information carried by interaction among each mode are useful for our regression problem. In order to verify our assumption, the independent data \( \mathbf{X} \) and dependent data \( \mathbf{Y} \) were generated according to the Tucker model that is regarded as a general model for tensors. The latent variables \( \mathbf{t} \) were generated in the same way as described in Section 4.1.1. A sequence of loadings \( \mathbf{B}^{(n)} \) and \( \mathbf{Q}^{(m)} \) and the core tensors were drawn from \( \mathcal{N}(0, 1) \). For the validation dataset, the latent matrix \( \mathbf{T} \) was generated from the same distribution as the calibration dataset, while the core tensors and loadings were fixed. Similarly to the study in Section 4.1.1, to investigate how the prediction performance is affected by noise levels and sample size, \( \{\mathbf{X}, \mathbf{Y}\} \in \mathbb{R}^{20 \times 10 \times 10} \) (Case 1) and \( \{\mathbf{X}, \mathbf{Y}\} \in \mathbb{R}^{10 \times 10 \times 10} \) (Case 2) were generated under noise levels of 10dB, 5dB, 0dB and -5dB. The datasets for both cases were generated from five latent variables.

### TABLE 2: The selection of parameters \( R \) and \( \lambda \) in Case 2.

<table>
<thead>
<tr>
<th>SNR</th>
<th>PLS</th>
<th>N-PLS</th>
<th>HOPLS</th>
<th>SNR</th>
<th>PLS</th>
<th>N-PLS</th>
<th>HOPLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10dB</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>0dB</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5dB</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>-5dB</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

The optimal parameters of \( R \) and \( \lambda \) were shown in Table 2. Observe that the optimal \( R \) is smaller with the increasing noise level for all the three methods. The parameter \( \lambda \) in HOPLS was also shown to have a similar behavior. For more detail, Fig. 5 exhibits the cross-validation performance grid of HOPLS with respect to \( R \) and \( \lambda \). When SNR was 10dB, the optimal \( \lambda \) was 4, while it were 2, 2 and 1 for 5dB, 0dB and -5dB respectively. This indicates that the model complexity can be adapted to provide a better model when a specific dataset was given, demonstrating the flexibility of HOPLS model.

Fig. 5: Five-fold cross-validation performance of HOPLS at different noise levels versus the number of latent variables (\( R \)) and loadings (\( \lambda \)). The optimal values for these two parameters are marked by green squares.

Fig. 6: The prediction performance comparison among HOPLS, N-PLS and PLS at different noise levels for the two cases (i.e., Case1: \( \{\mathbf{X}, \mathbf{Y}\} \in \mathbb{R}^{20 \times 10 \times 10} \) and Case 2: \( \{\mathbf{X}, \mathbf{Y}\} \in \mathbb{R}^{10 \times 10 \times 10} \) with different sample size.

The prediction performance evaluated over 50 validation datasets using HOPLS, N-PLS and PLS with individually selected parameters were compared for different noise levels and different sample sizes (i.e., two cases). As shown in Fig. 6 for both the cases, the prediction performance of HOPLS was better than both N-PLS and PLS at 10dB, and the discrepancy among them...
was enhanced when SNR changed from 10dB to -5dB. The performance of PLS decreased significantly with the increasing noise levels while HOPLS and N-PLS showed relative robustness to noise and outperformed PLS when SNR ≤ 5dB, illustrating the advantages of tensor-based methods with respect to noisy data. Regarding the small sample size problem, we found the performances of all the three methods were decreased when comparing Case 1 with Case 2. Observe that the superiority of HOPLS over N-PLS and PLS were enhanced in Case 2 as compared to Case 1 at all noise levels. A comparison of Fig. 6 and Fig. 4 shows that the performances are significantly improved when handling the datasets having tensor structure by tensor-based methods (e.g., HOPLS and N-PLS). As for N-PLS, it outperformed PLS when the datasets have tensor structure and in the presence of high noise, but it may not perform well when the datasets have no tensor structure. By contrast, HOPLS performed well in both cases, in particular, it outperformed both N-PLS and PLS in critical cases with high noise and small sample size.

### 4.1.3 Comparison on matrix response data

In this simulation, the response data was a two-way matrix, thus HOPLS2 algorithm was applied to evaluate the performance. \( \mathbf{X} \in \mathbb{R}^{3 \times 5 \times 5 \times 5} \) and \( \mathbf{Y} \in \mathbb{R}^{3 \times 2} \) were generated from a full-rank normal distribution \( \mathcal{N}(0, 1) \), which satisfies \( \mathbf{Y} = \mathbf{X}(1) \mathbf{W} \) where \( \mathbf{W} \) was also generated from \( \mathcal{N}(0, 1) \). Fig. 7(A) visualizes the predicted and original data with the red line indicating the ideal prediction. Observe that HOPLS was able to predict the validation dataset with smaller error than PLS and N-PLS. The independent data and dependent data are visualized in the latent space as shown in Fig. 7(B).

### 4.2 Decoding of ECoG signals

In [46], ECoG-based decoding of 3D hand trajectories was demonstrated by means of classical PLS regression [49]. The movement of monkeys was captured by an optical motion capture system (Vicon Motion Systems, USA). In all experiments, each monkey wore a custom-made jacket with reflective markers for motion capture affixed to the left shoulder, elbows, wrists and hand, thus the response data was naturally represented as a 3th-order tensor (i.e., time × 3D positions × markers). Although PLS can be applied to predict the trajectories corresponding to each marker individually, the structure information among four markers would be unused. For every point in time of the movement trajectory, a corresponding ECoG epoch (one second before the current time point) was extracted and transformed to the time-frequency domain in order to construct the predictors. Hence, the independent data is also naturally represented as a higher-order tensor (i.e., epoch × channel × time × frequency). In this study, the proposed HOPLS regression model was applied for decoding movement trajectories based on ECoG signals to verify its effectiveness in real-world applications.

The overall scheme of ECoG decoding is illustrated in Fig. 8. Specifically, ECoG signals were preprocessed by a band-pass filter with cutoff frequencies at 0.1 and 600Hz and a spatial filter with a common average reference. Motion marker positions were down-sampled to 20Hz. In order to represent features related to the movement trajectory from ECoG signals, the Morlet wavelet transformation at 10 different center frequencies (10-150Hz, arranged in a logarithmic scale) was used to obtain the time-frequency representation. The epochs of one-second ECoG signals before every time point of the movement data were extracted and downsampled to 10Hz in order to construct the predictors, which can be represented by channel × time × frequency \((32 \times 10 \times 10)\).

We first applied the HOPLS2 algorithm to predict only the hand movement trajectory, represented as a matrix \( \mathbf{Y} \in \mathbb{R}^{10 \times 3} \) (time points × 3D positions) from a three-order tensor of ECoG predictors \( \mathbf{X} \in \mathbb{R}^{32 \times 3 \times 100} \) (epoch × channel × time-frequency) that was constructed by combining the time and frequency modes into one mode. The ECoG data was divided into a calibration dataset

---

4. The datasets and more detailed description are freely available from [http://neurotycho.org](http://neurotycho.org).
(10 minutes) and a validation dataset (5 minutes). To select the optimal parameters of \( L_n \) and \( R \), the cross-validation was applied on the calibration dataset. Finally, \( L_n = 10 \) and \( R = 23 \) were selected for the HOPLS model. Likewise, the best values of \( R \) for PLS and N-PLS were 19 and 60, respectively. The X-latent space is visualized in Fig. 9(A), where each point represents one sample of independent variables, while the Y-latent space is presented in Fig. 9(B), with each point representing one dependent sample. Observe that the distributions of these two latent variable spaces were quite similar, and the two dominant clusters are clearly distinguished. The joint distributions between each \( t_c \) and \( u_r \) are depicted in Fig. 9(C). Two clusters can be observed from the first component which might be related to the ‘movement’ and ‘non-movement’ behaviors.

Another advantage of HOPLS was better physical interpretation of the model. To investigate how the spatial, spectral, and temporal structure of ECoG data were used to create the regression model, loading vectors can be regarded as a subspace basis in spatial and time-frequency domains, as shown in Fig. 10. With regard to time-frequency loadings, the \( \beta \) and \( \gamma \) bands were most significant implying the importance of \( \beta \), \( \gamma \)-band activities for encoding of movements; the duration of \( \beta \)-band was longer than that of \( \gamma \)-band, which indicates that hand movements were related to long history oscillations of \( \beta \)-band and short history oscillations of \( \gamma \)-band. These findings also demonstrated that a high gamma band activity in the premotor cortex is associated with movement preparation, initiation and maintenance.

From Table 3, observe that the improved prediction performances were achieved by HOPLS, for all the performance metrics. In particular, the results from dataset 1 demonstrated that the improvements by HOPLS over N-PLS were 0.03 for the correlation coefficient of X-position, 0.02 for averaged RMSEP, 0.04 for averaged \( Q^2 \), whereas the improvements by HOPLS over PLS were 0.03 for the correlation coefficient of X-position, 0.02 for averaged RMSEP, and 0.03 for averaged \( Q^2 \).

Since HOPLS enables us to create a regression model between two higher-order tensors, all trajectories recorded from shoulder, elbow, wrist and hand were contracted as a tensor \( \mathbf{Y} \in \mathbb{R}^{T \times 3 \times 4} \) (samples in 3D positions by markers). In order to verify the superiority of HOPLS for small sample sizes, we used 100 second data for calibration and 100 second data for validation. The resolution of time-frequency representations was improved to provide more detailed features, thus we have a 4th-order tensor \( \mathbf{X} \in \mathbb{R}^{I_1 \times 32 \times 20 \times 20} \) (samples by channels by time by frequency). The prediction performances from HOPLS, N-PLS and PLS are shown in Fig. 11 illustrating the effectiveness of HOPLS when the response data originally has tensor structure.

Time-frequency features of ECoG epochs for each sample are extremely overlapped, resulting in a lot of information redundancy and high computational burden. In addition, it is generally not necessary to predict behaviors with a high time-resolution. Hence, an additional analysis has been performed by down-sampling motion marker positions at 1Hz, to ensure that non-overlapped features were used in any adjacent samples. The cross-validation performance was evaluated for all the markers from the ten minute calibration dataset and the best performance for PLS of \( Q^2 = 0.19 \) was obtained using \( R = 2 \), for N-PLS it was \( Q^2 = 0.22 \) obtained by \( R = 5 \), and for HOPLS it was \( Q^2 = 0.28 \) obtained by \( R = 24, \lambda = 5 \). The prediction performances on the five minute validation dataset are shown in Fig. 12, implying the significant improvements obtained by HOPLS over N-PLS and PLS for all the four markers. For visualization, Fig. 13 exhibits the observed and predicted 3D hand trajectories in the 150s time window. The video

![Fig. 9: Panels (A) and (B) depict data distributions in the X-latent space T and Y-latent space U, respectively. (C) presents a joint distribution between X- and Y-latent vectors.](image1)

![Fig. 10: (A) Spatial loadings \( \mathbf{P}^{(1)} \) corresponding to the first two latent components. Each row shows 5 significant loading vectors. Likewise, (B) depicts time-frequency loadings \( \mathbf{P}^{(2)} \), with \( \beta \) and \( \gamma \)-band exhibiting significant contribution.](image2)
TABLE 3: Comprehensive comparison of the HOPLS, N-PLS and PLS for the prediction of 3D hand trajectories on the four datasets recorded from the same monkey brain. The numbers of latent vector for HOPLS, N-PLS and Unfold-PLS were 23, 60, and 19, respectively.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Model</th>
<th>$Q^2$(ECoG)</th>
<th>$Q^2$(3D hand positions)</th>
<th>RMSEP (3D hand positions)</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$X$ $Y$ $Z$</td>
<td>Mean</td>
<td>$X$ $Y$ $Z$</td>
<td>Mean</td>
</tr>
<tr>
<td>Dataset 1</td>
<td>HOPLS</td>
<td>0.25</td>
<td>0.43 0.48 0.61 0.51</td>
<td>0.82 0.70 0.66 0.73</td>
<td>0.67 0.72 0.78</td>
</tr>
<tr>
<td></td>
<td>N-PLS</td>
<td>0.33</td>
<td>0.39 0.44 0.59 0.47</td>
<td>0.85 0.73 0.68 0.75</td>
<td>0.64 0.71 0.77</td>
</tr>
<tr>
<td></td>
<td>Unfold-PLS</td>
<td>0.23</td>
<td>0.39 0.45 0.59 0.48</td>
<td>0.85 0.72 0.68 0.75</td>
<td>0.64 0.72 0.77</td>
</tr>
<tr>
<td>Dataset 2</td>
<td>HOPLS</td>
<td>0.25</td>
<td>0.12 0.42 0.50 0.35</td>
<td>0.99 0.77 0.72 0.83</td>
<td>0.35 0.64 0.71</td>
</tr>
<tr>
<td></td>
<td>N-PLS</td>
<td>0.33</td>
<td>0.03 0.40 0.51 0.32</td>
<td>1.04 0.78 0.71 0.84</td>
<td>0.32 0.64 0.71</td>
</tr>
<tr>
<td></td>
<td>Unfold-PLS</td>
<td>0.22</td>
<td>0.05 0.40 0.53 0.32</td>
<td>1.04 0.78 0.70 0.84</td>
<td>0.34 0.63 0.73</td>
</tr>
<tr>
<td>Dataset 3</td>
<td>HOPLS</td>
<td>0.22</td>
<td>0.36 0.39 0.48 0.41</td>
<td>0.74 0.77 0.66 0.73</td>
<td>0.62 0.62 0.69</td>
</tr>
<tr>
<td></td>
<td>N-PLS</td>
<td>0.30</td>
<td>0.31 0.37 0.46 0.38</td>
<td>0.77 0.78 0.68 0.74</td>
<td>0.61 0.62 0.68</td>
</tr>
<tr>
<td></td>
<td>Unfold-PLS</td>
<td>0.21</td>
<td>0.30 0.37 0.46 0.38</td>
<td>0.77 0.79 0.67 0.74</td>
<td>0.61 0.62 0.68</td>
</tr>
<tr>
<td>Dataset 4</td>
<td>HOPLS</td>
<td>0.16</td>
<td>0.16 0.50 0.57 0.41</td>
<td>1.04 0.66 0.62 0.77</td>
<td>0.43 0.71 0.76</td>
</tr>
<tr>
<td></td>
<td>N-PLS</td>
<td>0.23</td>
<td>0.12 0.45 0.55 0.37</td>
<td>1.06 0.69 0.67 0.80</td>
<td>0.41 0.70 0.76</td>
</tr>
<tr>
<td></td>
<td>Unfold-PLS</td>
<td>0.15</td>
<td>0.11 0.46 0.57 0.38</td>
<td>1.07 0.69 0.62 0.79</td>
<td>0.42 0.70 0.76</td>
</tr>
</tbody>
</table>

Fig. 11: The prediction performance of 3D trajectories recorded from shoulder, elbow, wrist and hand. The optimal $R$ are 16, 28, 49 for PLS, N-PLS and HOPLS, respectively, and $\lambda = 5$ for HOPLS.

is also available for visualization of the results\footnote{http://www.bsp.brain.riken.jp/%7Eqibin/homepage/HOPLS.html}

5 CONCLUSIONS

The higher-order partial least squares (HOPLS) has been proposed as a generalized multilinear regression model. The analysis and simulations have shown that the advantages of the proposed model include its robustness to noise and enhanced performance for small sample sizes. In addition, HOPLS provides an optimal tradeoff between fitness and overfitting due to the fact that model complexity can be adapted by a hyperparameter. The proposed strategy to find a closed-form solution for HOPLS makes computation more efficient than the existing algorithms. The results for a real-world application in decoding 3D movement trajectories from ECoG signals have also demonstrated that HOPLS would be a promising multilinear subspace regression method.

Fig. 12: The prediction performance of 3D trajectories for shoulder, elbow, wrist and hand using non-overlapped ECoG features.

Fig. 13: Visualization of observed trajectories (150s time window) and the trajectories predicted by HOPLS, N-PLS and PLS.
Yasu Nagasaka received the PhD degree in psychology from Rikkyo University, Japan in 2000. He was a postdoctoral fellow at the University of Iowa in 2004-2006. He is currently a postdoctoral fellow at RIKEN Brain Science Institute, Japan. His research interests broadly span the areas of animal cognition, including illusory perception, motor control behavior, and social brain function.

Naotaka Fujii received MD and PhD from Tohoku University School of Medicine, Sendai, Japan in 1991 and 1997 respectively. After PhD was granted, he worked in BCS, MIT as postdoctoral fellow. Currently he is a head of Laboratory for Adaptive Intelligence in Brain Science Institute, RIKEN, Japan. His research interests span the wide areas, social neuroscience in primate and human and development of brain machine interface and substitute reality system.

Qibin Zhao was born in 1979. He received the Ph.D. degree in engineering from Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2009. He is currently a research scientist at Laboratory for Advanced Brain Signal Processing in RIKEN Brain Science Institute, Japan and a visiting researcher at Non-Invasive BMI Unit of BSI-TOYOTA Collaboration Center, RIKEN. His research interests include multi-way data analysis, brain computer interface and machine learning.

Andrzej Cichocki received the Ph.D. and Dr.Sc. (Habilitation) degrees, all in electrical engineering, from Warsaw University of Technology (Poland). He is currently the senior team leader of the laboratory for Advanced Brain Signal Processing, at RIKEN Brain Science Institute (JAPAN). He is co-author of more than 250 technical papers and 4 monographs (two of them translated to Chinese). According to a 2011 analysis, he is an author and coauthor of one of the top 1% most highly cited papers in his field worldwide.

Cesar F. Caiafa was born in Buenos Aires, Argentina, in 1970. He received the Ph.D. degree in engineering from Faculty of Engineering, University of Buenos Aires, in 2007. He is currently Adjunct Researcher with the Argentinian Radioastronomy Institute (IAR) - CONICET and Assistant Professor with Faculty of Engineering, University of Buenos Aires. He is also Visiting Scientist at Lab. for Advanced Brain Signal Processing, BSI - RIKEN, Japan.

Yonas Chao received the PhD degree in biomedical engineering from the Georgia Institute of Technology in 2007. He is currently a research scientist in the Laboratory for Adaptive Intelligence at RIKEN Brain Science Institute (BSI), Japan, where the research focus is on primate neurophysiological recording and analysis.

Liqing Zhang received the Ph.D. degree from Zhongshan University, Guangzhou, China, in 1988. He was promoted to full professor position in 1995 at South China University of Technology. He worked as a research scientist in RIKEN Brain Science Institute, Japan from 1997 to 2002. He is now a Professor with Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests cover computational theory for cortical networks, brain-computer interface, perception and cognition computing model, statistical learning and inference. He has published more than 170 papers in international journals and conferences.

Danilo P. Mandic (M’99-SM’03) is a Professor of signal processing at Imperial College London, London, U.K., where he has been working in the area of nonlinear and multidimensional adaptive signal processing and time-frequency analysis. His publication record includes two research monographs titled Recurrent Neural Networks for Prediction (West Sussex, U.K.: Wiley, August 2001) and Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models (West Sussex, U.K.: Wiley, April 2009), an edited book titled Signal Processing for Information Fusion (New York: Springer, 2008), and more than 200 publications on signal and image processing. He has been a member of the IEEE Technical Committee on Machine Learning for Signal Processing, an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS, and the International Journal of Mathematical Modelling and Algorithms.

