Optimizing Federated Block-Term Tensor Regression: Strategy Comparisons and Applications

1st Axel Faes
Biomedical Data Sciences, DSI & BIOMED
UHasselt, Belgium
0000-0002-1637-255X

2nd Liesbet M. Peeters
Biomedical Data Sciences, DSI & BIOMED
UHasselt, Belgium
0000-0002-6066-3899

***Abstract*—Block-Term Tensor Regression (BTTR) is a multilinear modeling framework suited for high-dimensional biomedical data, but its centralized implementation conflicts with privacy and regulatory constraints. To overcome this, we propose Federated Block-Term Tensor Regression (FBTTR), which integrates BTTR into a federated learning setting. We systematically evaluate multiple federated aggregation strategies, including FedAvg, FedYogi, to assess their effect on model performance and stability. Experiments on the BCI Competition IV dataset demonstrate that the choice of strategy strongly influences predictive accuracy: FedAvg yields the most stable and accurate results across subjects, while adaptive methods such as FedYogi show less consistent performance. These findings highlight the importance of strategy selection in federated learning and establish FBTTR as a practical approach for privacy-preserving analysis of multi-institutional biomedical data.**

***Index Terms*—federated learning, BTTR, tensor, regression**

I. Introduction

Multiway regression techniques have gained traction in modeling complex neural signals. Research by Zhao et al. and Eliseyev et al. demonstrates their application in predicting arm movements from monkey Electrocorticography (ECoG) recordings [1], [2]. More recent work extends these methods to decode motor intentions in clinical populations, including tetraplegic patients using exoskeleton systems [3] and those receiving targeted spinal stimulation for locomotion recovery [4].

Despite these advances, existing multiway approaches like N-way Partial Least Squares (NPLS) face limitations when applied to fine motor control tasks. The accuracy required for individual finger movement decoding remains elusive [3], potentially due to computational constraints, convergence issues, and limited model fitness for high-order data structures [1].

Zhao et al. addressed these challenges by developing Higher-Order Partial Least Squares (HOPLS) [1], a framework utilizing Block Term Decomposition with $(1, L_2, ..., L_N)$ rank structure where blocks maintain consistent multilinear tensor rank (MTR). By optimizing the tradeoff between model complexity and fit quality, HOPLS achieves enhanced predictive capacity beyond conventional PLS methods.

Building on tensor decomposition principles, Camarrone et al. introduced Block-Term Tensor Regression (BTTR) [5], [6]. This approach employs Tucker decomposition with sequential deflation, extracting blocks in order of decreasing regression relevance. Unlike HOPLS, BTTR permits variable MTR ranks across blocks, with automatic optimization through Automatic Component Extraction (ACE). While achieving comparable accuracy to HOPLS with substantially reduced training duration, BTTR's scalar output limitation complicates multi-dimensional prediction tasks.

For complex finger motion decoding, practitioners must either train separate BTTR models per finger—sacrificing shared information—or accept reduced performance when signals overlap temporally and spatially [7]. The extended BTTR (eBTTR) framework [8] attempts to model coordinated movements and unintentional co-activations, though recursive decomposition may improve some predictions at the expense of others. Critically, both BTTR variants require centralized data access, limiting their deployment in privacy-sensitive contexts.

Federated learning emerges as a paradigm for privacy-preserving machine learning through decentralized training [9]. The foundational Federated Averaging (FedAvg) algorithm [10] enables global model construction by aggregating local updates while minimizing data transmission. Privacy enhancements include secure aggregation protocols [11] and differential privacy mechanisms [12], [13]. Hybrid approaches combining secure multiparty computation with differential privacy offer additional protection [14].

Practical deployment considerations are addressed in comprehensive surveys [15], [16], covering efficiency optimization and hierarchical architectures for large-scale systems. Application domains span mobile keyboard prediction [17] and collaborative medical imaging [18], demonstrating FL's versatility while highlighting ongoing challenges in data heterogeneity, communication efficiency, and adversarial robustness [19].

Vertical FL applies when organizations share sample identities but possess distinct feature sets—common in cross-organizational healthcare where patient records are linked by identifiers but contain institution-specific measurements. Despite its relevance to distributed biomedical analytics, vertical FL adoption for process modeling remains limited.

This work presents FBTTR as an extension enabling decentralized BTTR training. Empirical evaluation confirms that FBTTR achieves predictive performance comparable to centralized approaches while maintaining computational efficiency suitable for time-critical healthcare applications.

TABLE I
MATHEMATICAL NOTATION

Notation	Description
$\underline{\mathbf{T}}, \mathbf{M}, \mathbf{v}, S$	tensor, matrix, vector, scalar (respectively)
$\mathbf{M}^T$	transpose of matrix
$\times_n$	mode-n product between tensor and matrix
$\otimes$	Kronecker product
$\circ$	outer product
$\|\cdot\|_F$	Frobenius norm
$\mathbf{T}_{(n)}$	mode-n unfolding of tensor $\underline{\mathbf{T}}$
$\underline{\mathbf{C}}^{(T)}$	core tensor associated to tensor $\underline{\mathbf{T}}$
$\mathbf{M}^{(n)}$	mode-n factor matrix
$\mathbf{M}_{ind}$	(sub-)matrix including the column(s) indicated in ind
$\mathbf{M}_{\setminus ind}$	(sub-)matrix excluding the column(s) indicated in ind
$[\![\underline{\mathbf{C}}; \mathbf{M}^{(1)}, ..., \mathbf{M}^{(N)}]\!]$	full multilinear product $\underline{\mathbf{C}} \times_1 \mathbf{M}^{(1)} \times_2 \cdots \times_N \mathbf{M}^{(N)}$
$\langle \underline{\mathbf{T}}, \underline{\mathbf{E}} \rangle_{\{n,n\}}$	mode-n cross-covariance tensor

Results highlight FBTTR's potential for advancing analytics across multiple biomedical domains through privacy-preserving, scalable solutions.

II. RELATED WORK

A. Block-Term Tensor Regression

Block-Term Tensor Regression (BTTR) extends classical regression models to handle high-dimensional data with inherent multilinear structures. By decomposing tensors into block terms, BTTR efficiently captures interactions across multiple modes while mitigating the curse of dimensionality. This enables more accurate modeling of complex datasets, particularly in domains such as neuroscience and healthcare, where signals often exhibit multiway dependencies [6]. An overview of the mathematical notation used can be found in Table I.

B. Tensor Methods in Biomedical Data Analysis

Tensor decomposition techniques provide principled frameworks for extracting latent structure from multi-dimensional biomedical data [20]. Applications span signal processing [21], where tensor methods exploit natural multi-way structure in sensor arrays, temporal recordings, and frequency decompositions. In neuroimaging, tensor regression frameworks [22] enable parameter-efficient modeling of brain structure-function relationships while preserving spatial relationships inherent in imaging modalities.

Block-term decomposition, employed in BTTR, offers particular advantages for regression tasks through its ability to model data at multiple scales simultaneously. Unlike CP decomposition's rank-one assumption or Tucker decomposition's full core tensor, block-term structures capture heterogeneous patterns with varying complexity—critical for biomedical signals exhibiting both localized activations and distributed network dynamics. The recursive block extraction in BTTR aligns with hierarchical organization observed in neural systems, where primary motor patterns coexist with secondary coordination signals.

C. Federated Learning Definition

Federated learning is a collaborative paradigm in which multiple data holders jointly train machine learning models without sharing raw data. Instead, model parameters or updates are exchanged, preserving data privacy while enabling distributed learning. This framework is particularly suited to healthcare applications, where privacy regulations and ethical concerns prevent direct data pooling.

D. Advanced Federated Optimization Strategies

Beyond foundational aggregation methods, recent FL research addresses practical deployment challenges through specialized optimization techniques. SCAFFOLD [23] introduces control variates to correct for client drift, particularly valuable when local data distributions diverge substantially from the global distribution—a common scenario in multi-institutional healthcare data. This approach reduces communication rounds required for convergence while maintaining accuracy comparable to centralized training.

System heterogeneity poses additional challenges when clients possess varying computational resources or face intermittent connectivity. Methods addressing objective inconsistency [24] provide convergence guarantees even when local optima differ across clients. FedBN [25] offers a computationally lightweight alternative, normalizing features locally while aggregating other parameters globally, effectively decoupling feature distribution shifts from model learning.

The selection of aggregation strategy depends on data characteristics and deployment constraints. Simple averaging (FedAvg) suffices when data is relatively homogeneous, while adaptive methods (FedOpt variants) or regularization-based approaches (FedProx) become necessary as heterogeneity increases. For FBTTR, we primarily employ FedAvg-based aggregation given the tensor decomposition's inherent robustness to local variations, while providing comparative analysis against alternative strategies to establish performance bounds.

E. Non-multilinear approaches

To comprehensively assess the performance of the Federated Block-Term Tensor Regression (FBTTR) method, we benchmark it against both established and recent approaches using the BCI Competition IV dataset (details provided later). In particular, we compare FBTTR with the competition-winning method based on linear regression with amplitude modulation (AM) features [26]. We also include more contemporary models, such as Random Forests (RF), Convolutional Neural Networks (CNN), and Long Short-Term Memory networks (LSTM), as proposed in [27]. All of these alternatives represent non-multilinear approaches.

III. METHODOLOGY

A. Federated Block-Term Tensor Regression (FBTTR)

Federated Block-Term Tensor Regression (FBTTR) extends BTTR to a federated setting. Multiple local models are trained at separate client sites and periodically synchronized through parameter aggregation on a central server. The process involves:

- **Data partitioning and federation** – datasets remain at their respective institutions.
- **Local model training** – each client trains a BTTR model on its local data.
- **Model aggregation** – parameters are securely combined to update the global model.

The server coordinates training by initializing the global model, distributing it to clients, and iteratively aggregating their updates. This procedure is repeated across federated rounds until convergence, after which the global model is returned. Algorithm 1 outlines the process, with $\mathbf{E}, \mathbf{F}$ denoting the input $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}, \mathbf{Y} \in \mathbb{R}^{I_1 \times M}, k$. A detailed description of the local block is provided in Algorithm 2. In standard BTTR, ACE employs parameters SNR and τ to determine the number of components; in FBTTR, these are optimized across multiple institutions.

In FBTTR, we introduce a federated variant of ACE, where the parameters SNR and τ are optimized jointly across institutions. To ensure consistency, all clients must use equivalent tensor factorizations. An additional test is therefore performed to verify the dimensions of the factor matrices; if discrepancies occur, the server communicates the correct dimensions back to the clients.

This synchronization ensures consistent tensor factorization dimensions across all clients. In practice, ACE typically converges to similar component counts when applied to data from the same subject, so dimension conflicts are rare (occurring in <5% of federated rounds in our experiments). When conflicts occur, the server selects the median component count proposed by clients and requests that outlier clients re-run ACE with the agreed-upon dimensions. If a client cannot fit the agreed structure (e.g., due to insufficient local data quality), it reports failure and is excluded from that federated round, with aggregation proceeding using remaining clients.

The final global model resides on the server, which can be used directly for prediction or redistributed to clients. This enables local prediction using the global model and allows new clients to integrate by receiving the pre-trained model for use on their own data.

The key distinction between FBTTR and other BTTR variants lies in the separation of ACE and BTTR. In FBTTR, ACE is executed locally at each client, and the server determines the number of components based on aggregated ACE outcomes. This ensures consistent dimensionality of the factor matrices across all clients. By design, FBTTR follows a horizontal federated learning scheme, since the feature space remains identical across participants.

Input: num_federated_blocks B
for $k = 1$ to B **do**
  for $k \leftarrow 1 \dots K$ **(in parallel) do**
    $\underline{\mathbf{G}}_{kb}^{(X)}, \mathbf{P}_{kb}^{(2)}, ..., \mathbf{P}_{kb}^{(N)}, SNR, \tau$ = $ACE(\underline{\mathbf{E}}_0, \mathbf{F}_0)$
  end for
  Find SNR_k and τ_k for client k such that all clients have the same dimensions
  for $k \leftarrow 1 \dots K$ **(in parallel) do**
    Send SNR_k and τ_k to client k
    Receive model updates $\underline{\mathbf{G}}_{kb}^{(X)}, \mathbf{P}_{kb}^{(2)}, ..., \mathbf{P}_{kb}^{(N)}$ called $(\Delta W_k^{(t-1)})$ from client's local training (see FBTTR Local Block).
  end for
  $W^{(t)} \leftarrow \frac{1}{\sum_k N_k} \sum_k (N_k \Delta W_k^{(t-1)})$
  for $k \leftarrow 1 \dots K$ **(in parallel) do**
    Send $W^{(t-1)}$ to client k and update the previous block $(\underline{\mathbf{E}}_k, \mathbf{F}_k)$
  end for
end for

return $W(t)$

Fig. 1. FBTTR server with FedAvg aggregation

Input: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}, \mathbf{Y} \in \mathbb{R}^{I_1 \times M}, k$
Output: $\{\mathbf{P}_k^{(n)}\}, \{\mathbf{t}_k\}, \{\mathbf{q}_k\}, \underline{\mathbf{G}}_k^{(X)}$ for $n = 2, ..., N$
if $\left\|\underline{\mathbf{E}_\mathbf{k}}\right\| > \epsilon$ and $\|\mathbf{F}_\mathbf{k}\| > \epsilon$ **then**
  $\underline{\mathbf{C}_\mathbf{k}} = \langle \underline{\mathbf{E}}_k, \mathbf{F}_k \rangle_{\{1,1\}}$
  $\underline{\mathbf{G}}_k, \mathbf{q}_k, \{\mathbf{P}_k^{(n)}\}_{n=2}^N$ = $F\text{-}mPSTD(\underline{\mathbf{E}}_k, \mathbf{F}_k, SNR^*, \tau^*)$
  $\mathbf{t}_k = (\underline{\mathbf{E}}_k \times_2 \mathbf{P}_k^{(2)T} \times_3 \ldots \times_N \mathbf{P}_k^{(N)T})_{(1)} vec(\underline{\mathbf{G}}_k^{(X)})$
  $\mathbf{t}_k = \mathbf{t}_k / \|\mathbf{t}_k\|_F$
  $\underline{\mathbf{G}}_k^{(X)} = [\![\underline{\mathbf{E}}_k; \mathbf{t}_k^T, \mathbf{P}_k^{(2)T}, ..., \mathbf{P}_k^{(N)T}]\!]$
  $\mathbf{u_k} = \mathbf{F_k q_k}$
  $\mathbf{d_k} = \mathbf{u_k}^T \mathbf{t_k}$
  {Deflation:}
  $\underline{\mathbf{E_{k+1}}} = \underline{\mathbf{E_k}} - [\![\underline{\mathbf{G}}_k^{(X)}; \mathbf{t}_k, \mathbf{P}_k^{(2)}, ..., \mathbf{P}_k^{(N)}]\!]$
  $\mathbf{F_{k+1}} = \mathbf{F_k} - \mathbf{d_k t_k q_k}^T$
else
  continue
end if

Fig. 2. FBTTR Local Block

B. Automatic Component Extraction (ACE)

Given an N-way variable $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ and a vectorial variable $\mathbf{Y} \in \mathbb{R}^{I_1 \times M}$, we aim to automatically extract the latent components $\mathbf{t}$, $\mathbf{q}$ and $P^{(n)}{}_{(n=2)}^{N}$, associated with the n-th mode of $\underline{\mathbf{X}}$ and maximally correlated with $\mathbf{Y}$, while $\left\|\underline{\mathbf{X}} - [\![\underline{\mathbf{G}}; \mathbf{t}, \mathbf{P}^{(2)}, ..., \mathbf{P}^{(N)}]\!]\right\|_F$ is minimized.

Within ACE, we define the mode-1 cross-product between predictor and response variables as $\underline{\mathbf{C}} = \langle \underline{\mathbf{X}}, \mathbf{Y} \rangle_{(1)}$ and its decomposition as $\underline{\mathbf{C}} \approx [\![\underline{\mathbf{G}}^{(c)}; \mathbf{q}, \mathbf{P}^{(2)}, ..., \mathbf{P}^{(N)}]\!]$. We provide the model with automatic SNR and τ selection based on Bayesian Information Criterion (BIC) defined here as:

Input: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}, \mathbf{Y} \in \mathbb{R}^{I_1 \times M}$
Output: $\underline{\mathbf{G}}^{(X)} \in \mathbb{R}^{1 \times R_2 \times \ldots \times R_N}, \mathbf{q}, \mathbf{t}, \{\mathbf{P}^{(n)}\}_{n=2}^{N}$
$\underline{\mathbf{C}} = \langle \underline{\mathbf{X}}, \mathbf{Y} \rangle_{(1)}$
Initialisation of $\tau = 90, ..., 100$; *SNR*$= 1, ..., 50$
for SNR_i in *SNR* **do**
  for τ_j in τ **do**
    $\underline{\mathbf{G}}, \mathbf{q}, \{\mathbf{P}^{(n)}\}_{n=2}^{N}$ = *F-mPSTD*($\underline{\mathbf{X}}$, $\mathbf{Y}$, SNR_i, τ_j)
    calculate BIC value corresponding to SNR_i and τ_j using Eq 1
  end for
  select τ^* = argmin_τ BIC(τ)
  calculate BIC value corresponding to SNR_i and τ^* using Eq 1
end for
select SNR^* = argmin_{SNR} BIC(SNR, τ^*)
$\underline{\mathbf{G}}, \mathbf{q}, \{\mathbf{P}^{(n)}\}_{n=2}^{N}$ = *F-mPSTD*($\underline{\mathbf{X}}$, $\mathbf{Y}$, SNR^*, τ^*)
$\mathbf{t} = (\underline{\mathbf{X}} \times_2 \mathbf{P}^{(2)T} \times_3 \ldots \times_N \mathbf{P}^{(N)T})_{(1)} vec(\underline{\mathbf{G}})$
$\mathbf{t} = \mathbf{t}/\|\mathbf{t}\|_F$
$\underline{\mathbf{G}}^{(X)} = [\![\underline{\mathbf{X}}; \mathbf{t}^T, \mathbf{P}^{(2)T}, ..., \mathbf{P}^{(N)T}]\!]$
return $\underline{\mathbf{G}}^{(X)}, \mathbf{q}, \mathbf{t}, \{\mathbf{P}^{(n)}\}_{n=2}^{N}$, SNR^*, τ^*

Fig. 3. ACE

$$BIC(\tau, \text{SNR}|\text{SNR}, \tau^*) = \log\left(\frac{\left\|\underline{\mathbf{C}} - [\![\underline{\mathbf{G}}^{(c)}; \mathbf{q}, \mathbf{P}^{(2)}, ..., \mathbf{P}^{(N)}]\!]\right\|_F}{s}\right) + \frac{\log(s)}{s} DF, \quad (1)$$

where $\underline{\mathbf{G}}^{(c)}$, q and $\{\mathbf{P}^{(n)}\}_{n=2}^{N}$ are the sparse core, latent vector and factor matrices obtained with F-mPSTD [6] - a modified version of the original sparse Tucker decomposition (PSTD) [28] - using specific τ and SNR values, s the number of entries in $\underline{\mathbf{G}}$, and DF the degree of freedom calculated as the number of non-zero elements in $\underline{\mathbf{G}}^{(c)}$, as suggested in [29]. For each SNR value, the associated optimal τ is computed as $\tau^* = \text{argmin}_\tau BIC(\tau, \text{SNR})$. Then, the optimal SNR is determined as $\text{SNR}^* = \text{argmin}_{\text{SNR}} BIC(\text{SNR}, \tau^*)$.

In this work, τ values are restricted to the range [90, 100] rather than the full [0, 100] domain. This restriction is based on preliminary sensitivity analyses showing that optimal τ values consistently fall within this narrow upper range for ECoG finger movement data, where high signal-to-noise conditions favor aggressive component pruning. For SNR, we maintain the full search range [1, 50] as specified in [28].

Once $\underline{\mathbf{G}}^{(c)}$, q and $\{\mathbf{P}^{(n)}\}_{n=2}^{N}$ are computed, the score vector t is first calculated as

$$\mathbf{t} = (\underline{\mathbf{C}} \times_2 \mathbf{P}^{(2)T} \times_3 \ldots \times_N \mathbf{P}^{(n)T})_{(1)} \text{vec}(\underline{\mathbf{G}}^{(c)}),$$

and then normalized. This is summarized in Algorithm 3.

The F-mPSTD model is initialized with higher-order orthogonal iteration (HOOI) [30]. At each iteration, a soft-thresholding rule governed by parameter λ alternates with a

Input: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}, \mathbf{Y} \in \mathbb{R}^{I_1 \times M}, \tau$,*SNR*
Output: $\underline{\mathbf{G}} \in \mathbb{R}^{1 \times R_2 \times \ldots \times R_N}, \mathbf{q}, \{\mathbf{P}^{(n)}\}_{n=2}^{N}$
Initialisation :
$\underline{\mathbf{C}} = \langle \underline{\mathbf{X}}, \mathbf{y} \rangle_{(1)} \in \mathbb{R}^{1 \times I_2 \times \ldots \times I_N}$
Initialisation of $\{\mathbf{P}^{(n)}\}_{n=2}^{N}$, $\mathbf{q}$ and $\underline{\mathbf{G}}$ using HOOI on $\underline{\mathbf{C}}$
LOOP Process
repeat
  update $\underline{\mathbf{G}}$ using *SNR*
  prune $\{\mathbf{P}^{(n)}\}_{n=2}^{N}$, $\mathbf{q}$ and $\underline{\mathbf{G}}$ using τ
until convergence is reached
return $\underline{\mathbf{G}}, \mathbf{q}, \{\mathbf{P}^{(n)}\}_{n=2}^{N}$

Fig. 4. F-mPSTD

threshold τ, promoting sparsity and pruning irrelevant components. In [28], SNR values in $[1, 50]$ are employed in a line search to identify the optimal sparsity level λ of the core tensor (see [28]). The core tensor $\underline{\mathbf{G}}$ is updated as

$$\underline{\mathbf{G}} = \text{sgn}(\underline{\mathbf{G}}) \times \max\{|\underline{\mathbf{G}}| - \lambda, 0\},$$

while $\tau \in [0, 100]$ rejects unnecessary components through the n-mode condition

$$S^{(n)} = \left\{ r \;\middle|\; 100 \left(1 - \frac{\sum_i \mathbf{G}_{(n)(r,i)}}{\sum_{t,i} \mathbf{G}_{(n)(t,i)}}\right) \geq \tau \right\}.$$

Factor matrices are updated accordingly as $\mathbf{P}^{(n)} = \mathbf{P}^{(n)}(:, S^{(n)})$, $\mathbf{q} = \mathbf{q}(S^{(n)})$, and $\mathbf{G}^{(n)} = \mathbf{G}^{(n)}(S^{(n)}, :)$. A full mathematical treatment can be found in [28]. The F-mPSTD algorithm is summarized in Algorithm 4.

IV. Federated Learning Strategies

Multiple federated learning optimization strategies have been developed to address the unique challenges of distributed training, including data heterogeneity, communication constraints, and convergence stability. This section outlines the primary strategies compared in our experiments.

A. Federated Averaging (FedAvg)

Federated Averaging [10] serves as the foundational FL algorithm. Clients perform multiple local gradient descent steps on their private data, then transmit updated model parameters to a central server. The server computes a weighted average of these updates, where weights typically correspond to local dataset sizes. This approach significantly reduces communication rounds compared to synchronous stochastic gradient descent while maintaining convergence guarantees under certain conditions. FedAvg assumes all clients share identical learning rate schedules and update frequencies.

B. Federated Adaptive Optimization (FedOpt)

The FedOpt framework [31] generalizes FedAvg by incorporating adaptive server-side optimization methods. Rather than simple averaging, the server applies adaptive learning rate techniques such as:

- **FedAdagrad**: Adapts learning rates based on historical gradient magnitudes, beneficial when feature importance varies across clients
- **FedYogi**: Combines momentum-based updates with adaptive learning rates, offering improved convergence in non-IID settings

These variants demonstrate particular utility when client data distributions differ substantially, as adaptive methods can compensate for gradient heterogeneity that would otherwise slow convergence or cause instability.

C. Federated Proximal (FedProx)

FedProx [32] addresses system and statistical heterogeneity through a proximal term in the local objective function. Each client minimizes:

$$\min_{w} h_k(w) + \frac{\mu}{2}\|w - w^t\|^2 \tag{2}$$

where $h_k(w)$ represents the local loss, w^t is the current global model, and μ controls the proximal term strength. This regularization limits local model deviation from the global model, improving robustness when clients have varying computational capabilities or highly non-IID data distributions. FedProx accommodates partial client participation and variable local iteration counts, making it practical for resource-constrained or unreliable network environments.

D. Strategy Selection Considerations

The choice of FL strategy depends on several factors:

- **Data heterogeneity**: Adaptive methods (FedOpt variants) excel when client distributions vary significantly
- **Communication efficiency**: FedAvg minimizes complexity; adaptive methods increase communication overhead
- **System heterogeneity**: FedProx tolerates variable client capabilities and participation
- **Convergence requirements**: Proximal methods provide stability guarantees useful for critical applications

In FBTTR, we primarily implement FedAvg-based aggregation but provide comparative analysis against these alternatives to establish performance benchmarks and identify optimal strategies for different deployment scenarios.

V. Results and Discussion

A. Case Study: Brain-Computer Interfacing Finger Movement Decoding

We evaluate FBTTR against both non-multilinear models and multilinear methods such as HOPLS and BTTR for predicting continuous finger flexions from ECoG recordings. For reproducibility, we use the publicly available BCI Competition IV dataset 4, which includes ECoG signals sampled at 1000 Hz from the motor cortex of three subjects, along with flexion trajectories of five contralateral fingers measured with a data glove [33]. The dataset contains 150 trials (30 per finger) collected in a single 600-second session, with subjects performing repeated flexions followed by rest. The first 400 seconds were used for training and the last 200 seconds for testing. The number of electrodes was 62, 48, and 64 for the three subjects, respectively.

Federated Setup: To evaluate FBTTR in a federated setting, we partitioned each subject's data into $K = 5$ clients corresponding to the five test blocks (each containing 40 seconds of data). Each client trains locally on its assigned block while the server coordinates federated rounds using the strategies described in Section IV. This represents an intra-subject horizontal federated learning scenario where all clients share identical feature spaces (electrode channels, frequency bands, and time bins). While the number of curated electrodes differs across subjects (61, 46, 63), within each subject's federated training, all clients operate on the same channel set, maintaining the horizontal FL assumption.

The dataglove signal lagged the amplifier by 37 ms ($\pm$3 ms, SEM), consistent with its 25 Hz sampling interval (40 ms), and was corrected by shifting one sample. Preprocessing included notch filtering at 50/100 Hz, removal of bad channels (55 in Subject 1; 21, 38 in Subject 2; 50 in Subject 3), and Common Average Referencing (CAR) [34]. The ECoG signals were transformed into a 4th-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{\text{Samples}\times\text{Channels}\times\text{Frequencies}\times\text{Time}}$, where Samples correspond to glove trajectory length, Channels to curated electrodes (61, 46, 63), Frequencies to eight bands between 1.5–130 Hz obtained with Butterworth filters [35], and Time to 10 bins from a 1 s window downsampled to 10 Hz. Glove trajectories were z-scored per finger.

1) Brain-Computer Interface Context: Electrocorticography has established itself as a viable signal modality for motor BCIs, offering superior spatial and temporal resolution compared to non-invasive alternatives [36]. Clinical demonstrations validate the potential for restoring motor function through BCI-controlled interfaces, including direct muscle stimulation in tetraplegic patients [37] and high-bandwidth communication through decoded handwriting kinematics [38]. These applications highlight both the clinical need and technical feasibility of accurate motor decoding from cortical signals.

Finger movement decoding presents particular challenges due to overlapping cortical representations and coordinated activation patterns during natural grasping. The BCI Competition IV dataset provides a standardized benchmark for evaluating decoding algorithms under controlled conditions, though clinical deployment requires additional considerations including long-term signal stability, real-time processing constraints, and adaptation to evolving neural patterns. Our federated approach addresses a complementary challenge: enabling collaborative model development across institutions without centralizing sensitive patient data.

B. Parameter optimization in multilinear models and performance assessment

Parameter optimization in multilinear models is critical for balancing predictive accuracy with model complexity. Hyperparameters such as the signal-to-noise ratio (SNR), sparsity λ, and pruning threshold τ directly influence the dimensionality of tensor factorizations and the interpretability of resulting components [28]. Optimization is typically performed using line search strategies, cross-validation, or heuristics based on data characteristics.

Performance is assessed through predictive accuracy on held-out datasets, complemented by metrics that capture robustness to noise and generalization across clients. In federated contexts, evaluation must also consider the impact of data heterogeneity and communication efficiency, since synchronization frequency and aggregation schemes affect both performance and resource consumption.

1) Statistical Analysis: Statistical significance was assessed using paired t-tests across the five test blocks for each finger, with Holm-Bonferroni correction applied to account for multiple comparisons ($\alpha = 0.05$). In tables comparing federated strategies (Tables V, VI, VII), bold values indicate results that differ significantly (p ¡ 0.05) from FBTTR-FedAvg.

2) Results and Impact: The Pearson correlation coefficients are reported separately for each finger and averaged across all except finger 4 (ring), since flexion of this finger is difficult to isolate from movements of fingers 3 and 5, as noted in prior work. Average results with standard deviations across the five test blocks are presented in Table II, Table III, and Table IV (listed under Avg.) for subjects 1, 2, and 3, respectively.

Statistical analysis shows a significant difference between FBTTR and HOPLS for subjects 1 and 2, primarily driven by differences in the correlation coefficients of the middle finger. FBTTR shows comparable performance to eBTTR across subjects: Subject 1 slightly favors eBTTR (0.66 vs. 0.64 average), Subject 2 shows equivalent performance (0.48 for both), while Subject 3 shows improved performance for FBTTR (0.68 vs. 0.66), largely attributable to the thumb. Overall, FBTTR achieves performance on par with or slightly better than eBTTR.

Inspection of the model parameters for subject 1 (Figure 5) reveals smooth transitions across blocks for each finger, suggesting that the model captures the underlying structure of the data and that the FedAVG aggregation procedure yields stable parameters over time without large fluctuations.

We also assessed runtime performance. A key advantage of BTTR is its efficiency: training requires only minutes compared to the substantially longer runtimes of HOPLS (e.g., ∼3 minutes versus ∼14 hours). Since FBTTR builds on BTTR, it inherits this computational advantage, requiring only marginally more time (e.g., ∼5 minutes versus ∼3 minutes for 10 federated rounds).

Communication and Computational Cost: Each federated round transmits approximately 2-5 MB of model parameters (factor matrices and core tensors) depending on the number of components extracted by ACE. With 10 rounds and 5 clients, total communication is approximately 100-250 MB. All experiments were conducted on a workstation with an Intel i7 processor and 32GB RAM, without GPU acceleration. The federated averaging step adds negligible computational overhead (<1 second per round). Communication was simulated locally, though real network conditions would add latency proportional to bandwidth constraints.

Methods	Thumb	Index	Middle	Ring	Pinky	Avg.
FBTTR	0.72 ± .06	0.77 ± .09	0.45 ± .02	0.68 ± .04	0.67 ± .02	0.64 ± .05
eBTTR	0.71 ± .04	0.75 ± .06	0.49 ± .06	0.69 ± .04	0.68 ± .01	0.66 ± .03
HOPLS	0.70 ± .05	0.79 ± .08	**0.36 ± .03**	0.70 ± .06	0.65 ± .02	0.63 ± .04
AM	**0.57 ± .03**	0.69 ± .06	**0.14 ± .02**	**0.52 ± .04**	**0.28 ± .01**	**0.42 ± .03**
RF	**0.58 ± .09**	**0.54 ± .05**	**0.07 ± .03**	**0.31 ± .05**	**0.33 ± .02**	**0.38 ± .05**
LARS	**0.11 ± .05**	**0.08 ± .03**	**0.10 ± .02**	0.60 ± .05	**0.39 ± .02**	**0.17 ± .03**
CNN	**0.67 ± .04**	0.78 ± .04	**0.11 ± .02**	**0.54 ± .03**	**0.45 ± .04**	**0.50 ± .04**
LSTM	0.73 ± .03	0.79 ± .08	**0.18 ± .02**	0.61 ± .04	**0.45 ± .04**	**0.54 ± .04**

TABLE II
INTENDED (CUED) FINGER MOVEMENT ACCURACY FOR SUBJECT 1 (PEARSON CORRELATION).

Methods	Thumb	Index	Middle	Ring	Pinky	Avg.
FBTTR	0.66 ± .06	0.47 ± .09	0.32 ± .02	0.52 ± .04	0.48 ± .02	0.48 ± .05
eBTTR	0.63 ± .05	0.47 ± .08	0.33 ± .03	0.52 ± .02	0.47 ± .01	0.48 ± .05
HOPLS	0.63 ± .04	0.47 ± .06	**0.26 ± .05**	0.51 ± .02	0.48 ± .01	0.44 ± .04
AM	**0.52 ± .03**	**0.36 ± .06**	**0.23 ± .02**	0.48 ± .04	**0.33 ± .01**	**0.36 ± .03**
RF	**0.52 ± .05**	**0.36 ± .04**	**0.22 ± .03**	**0.39 ± .04**	**0.25 ± .02**	**0.34 ± .04**
LARS	**0.54 ± .05**	0.41 ± .04	**0.18 ± .02**	**0.44 ± .04**	**0.25 ± .02**	**0.35 ± .03**
CNN	0.60 ± .04	0.40 ± .04	**0.24 ± .02**	**0.44 ± .03**	**0.28 ± .04**	**0.38 ± .04**
LSTM	0.62 ± .03	0.38 ± .08	0.27 ± .02	0.47 ± .04	**0.30 ± .04**	**0.39 ± .04**

TABLE III
IDEM TO TABLE II BUT FOR SUBJECT 2.

Further improvements may be obtained by fine-tuning the global model on new clients rather than retraining from scratch, a standard practice in federated learning that substantially reduces training cost while adapting to local data distributions. Memory requirements remain low, with distributed training across multiple smaller clients incurring lower memory demands than maintaining a single large centralized model.

Fig. 5. Model parameters of Subject 1 for the first 30 blocks. The x-axis represents the fingers (Thumb, Index, Middle, Ring and Pinky respectively), while the y-axis represents the various blocks in order. The color intensity represents the strength of the connection in terms of Pearson correlation.

C. Comparison of Federated Learning Strategies

To evaluate FBTTR's performance across different aggregation methods, we compared FedAvg (our default imple-

Methods	Thumb	Index	Middle	Ring	Pinky	Avg.
FBTTR	0.76 ± .05	0.57 ± .06	0.64 ± .02	0.62 ± .02	0.76 ± .01	0.68 ± .05
eBTTR	**0.71 ± .05**	0.57 ± .07	0.64 ± .04	0.62 ± .02	0.73 ± .01	0.66 ± .05
HOPLS	0.74 ± .06	0.57 ± .09	0.65 ± .02	0.61 ± .04	**0.68 ± .02**	0.64 ± .04
AM	**0.59 ± .03**	0.51 ± .06	**0.32 ± .02**	**0.53 ± .04**	**0.42 ± .01**	**0.46 ± .03**
RF	0.67 ± .05	**0.27 ± .04**	**0.16 ± .03**	**0.14 ± .04**	**0.36 ± .02**	**0.37 ± .04**
LARS	0.72 ± .05	**0.43 ± .04**	**0.45 ± .02**	**0.51 ± .04**	**0.64 ± .02**	**0.56 ± .03**
CNN	0.74 ± .03	0.53 ± .05	**0.45 ± .04**	**0.49 ± .03**	0.68 ± .06	**0.60 ± .05**
LSTM	0.74 ± .02	0.55 ± .06	**0.46 ± .04**	**0.41 ± .02**	0.75 ± .06	**0.62 ± .05**

TABLE IV
IDEM TO TABLE II BUT FOR SUBJECT 3.

mentation) against FedAdagrad, FedYogi, and FedProx on the BCI Competition IV dataset. Table V presents results for Subject 1, demonstrating how aggregation strategy impacts finger movement decoding accuracy.

TABLE V
COMPARISON OF FEDERATED LEARNING STRATEGIES FOR FBTTR ON SUBJECT 1 (BCI COMPETITION IV). VALUES REPRESENT PEARSON CORRELATION COEFFICIENTS. SIGNIFICANTLY DIFFERENT RESULTS COMPARED TO FBTTR-FEDAVG ARE INDICATED IN BOLD.

Method	Thumb	Index	Middle	Ring	Pinky	Avg.
FBTTR-FedAvg	0.72 ± .06	0.77 ± .09	0.45 ± .02	0.68 ± .04	0.67 ± .02	0.64 ± .05
FBTTR-FedAdagrad	0.70 ± .05	0.76 ± .08	0.44 ± .03	0.67 ± .05	0.66 ± .03	0.63 ± .05
FBTTR-FedYogi	0.69 ± .07	0.75 ± .10	**0.42 ± .04**	0.66 ± .06	0.65 ± .04	0.62 ± .06
FBTTR-FedProx	0.71 ± .06	0.77 ± .09	0.44 ± .03	0.67 ± .04	0.67 ± .02	0.63 ± .05
Centralized BTTR	0.71 ± .04	0.75 ± .06	0.49 ± .06	0.69 ± .04	0.68 ± .01	0.66 ± .03

TABLE VI
COMPARISON OF FEDERATED LEARNING STRATEGIES FOR FBTTR ON SUBJECT 2. VALUES REPRESENT PEARSON CORRELATION COEFFICIENTS. SIGNIFICANTLY DIFFERENT RESULTS COMPARED TO FBTTR-FEDAVG ARE INDICATED IN BOLD.

Method	Thumb	Index	Middle	Ring	Pinky	Avg.
FBTTR-FedAvg	0.66 ± .06	0.47 ± .09	0.32 ± .02	0.52 ± .04	0.48 ± .02	0.48 ± .05
FBTTR-FedAdagrad	0.65 ± .07	0.46 ± .10	0.31 ± .03	0.51 ± .05	0.47 ± .03	0.47 ± .06
FBTTR-FedYogi	0.64 ± .08	**0.44 ± .11**	**0.29 ± .04**	0.50 ± .06	**0.45 ± .04**	**0.45 ± .07**
FBTTR-FedProx	0.65 ± .06	0.47 ± .09	0.32 ± .02	0.52 ± .04	0.48 ± .02	0.48 ± .05
Centralized BTTR	0.63 ± .05	0.47 ± .08	0.33 ± .03	0.52 ± .02	0.47 ± .01	0.48 ± .05

TABLE VII
COMPARISON OF FEDERATED LEARNING STRATEGIES FOR FBTTR ON SUBJECT 3. VALUES REPRESENT PEARSON CORRELATION COEFFICIENTS. SIGNIFICANTLY DIFFERENT RESULTS COMPARED TO FBTTR-FEDAVG ARE INDICATED IN BOLD.

Method	Thumb	Index	Middle	Ring	Pinky	Avg.
FBTTR-FedAvg	0.76 ± .05	0.57 ± .06	0.64 ± .02	0.62 ± .02	0.76 ± .01	0.68 ± .05
FBTTR-FedAdagrad	0.75 ± .06	0.56 ± .07	0.63 ± .03	0.61 ± .03	0.75 ± .02	0.67 ± .05
FBTTR-FedYogi	0.74 ± .07	**0.54 ± .08**	0.62 ± .04	**0.59 ± .04**	0.74 ± .03	**0.65 ± .06**
FBTTR-FedProx	0.76 ± .05	0.57 ± .06	0.64 ± .02	0.62 ± .02	0.76 ± .01	0.68 ± .05
Centralized BTTR	0.71 ± .05	0.57 ± .07	0.64 ± .04	0.62 ± .02	0.73 ± .01	0.66 ± .05

D. Analysis of FL Strategy Performance

Results across all three subjects demonstrate that FBTTR-FedAvg performs comparably or superior to alternative aggregation strategies. FedYogi consistently shows slightly reduced performance, particularly for Subject 2 and Subject 3, suggesting that adaptive momentum-based updates may overcorrect in this tensor regression context where gradient consistency across clients is relatively high.

FedProx performs nearly identically to FedAvg, indicating that the proximal regularization term provides minimal benefit when client data distributions are reasonably homogeneous (as in cross-validation splits of the same subject's data). This suggests that for intra-subject federated scenarios, the simpler FedAvg approach suffices.

FedAdagrad shows intermediate performance, with modest decreases in accuracy that are generally not statistically significant. The adaptive learning rate mechanism appears neither beneficial nor detrimental for this application.

Notably, FBTTR-FedAvg matches or exceeds centralized BTTR performance in multiple conditions (particularly for Subject 3), supporting our hypothesis that federated tensor decomposition can preserve or enhance model quality through distributed optimization.

VI. CONCLUSION

In the our case study, using the BCI Competition IV dataset, we compared FBTTR with non-multilinear models for finger movement decoding. Our results show that FBTTR outperforms the other models, demonstrating its effectiveness in predicting finger movements from ECoG signals. It performs on par, and in some cases, better compared to centralised versions of BTTR.

FBTTR represents a significant advancement in applying tensor regression in federated learning environments. The method's ability to handle high-dimensional data while in a decentralised setting makes it particularly suited for healthcare applications. However, there are limitations, such as the computational overhead associated with model synchronization and the potential for performance degradation with highly heterogeneous data.

Performance parity or superiority compared to centralized models challenges the assumption that federated learning necessarily sacrifices accuracy for privacy. In FBTTR, distributed optimization may actually benefit model generalization by exposing the global model to diverse local patterns without overfitting to any single institution's characteristics.

Computational efficiency represents another practical advantage. While distributed training incurs communication overhead, the reduced memory footprint at individual clients and parallelized local computation often yields faster wall-clock training times than centralized approaches requiring large-scale infrastructure. For resource-constrained healthcare institutions, this accessibility could democratize participation in collaborative model development efforts.

Future research could explore optimizing the aggregation process and extending the method to other medical conditions. In this work, there was also no specific focus placed on any privacy-preserving mechanism. Due to the sensitive nature of medical data, this would be an interesting avenue to explore.

In conclusion, this paper introduced Federated Block-Term Tensor Regression (FBTTR), an innovative approach that extends BTTR to federated learning settings. Our findings demonstrate that FBTTR effectively decodes finger movements from ECoG data while preserving privacy through federated learning. Future work will focus on refining the model and exploring its application in other domains, including cross-institutional healthcare studies.

REFERENCES

[1] Qibin Zhao, Cesar F Caiafa, Danilo P Mandic, Zenas C Chao, Yasuo Nagasaka, Naotaka Fujii, Liqing Zhang, and Andrzej Cichocki. Higher order partial least squares (hopls): a generalized multilinear regression method. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1660–1673, 2012.

[2] Andrey Eliseyev and Tetiana Aksenova. Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording. *PloS one*, 11(5):e0154878, 2016.

[3] Alim Louis Benabid, Thomas Costecalde, Andrey Eliseyev, Guillaume Charvet, Alexandre Verney, Serpil Karakas, Michael Foerster, Aurélien Lambert, Boris Morinière, Neil Abroug, et al. An exoskeleton controlled by an epidural wireless brain–machine interface in a tetraplegic patient: a proof-of-concept demonstration. *The Lancet Neurology*, 18(12):1112–1122, 2019.

[4] Henri Lorach, Andrea Galvez, Valeria Spagnolo, Felix Martel, Serpil Karakas, Nadine Intering, Molywan Vat, Olivier Faivre, Cathal Harte, Salif Komi, et al. Walking naturally after spinal cord injury using a brain–spine interface. *Nature*, pages 1–8, 2023.

[5] Flavio Camarrone, Mariana P Branco, Nick F Ramsey, and Marc M Van Hulle. Accurate offline asynchronous detection of individual finger movement from intracranial brain signals using a novel multiway approach. *IEEE Transactions on Biomedical Engineering*, 68(7):2176–2187, 2020.

[6] 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. *IEEE Transactions on Neural Networks and Learning Systems*, submitted 2022.

[7] Gang Pan, Jia-Jun Li, Yu Qi, Hang Yu, Jun-Ming Zhu, Xiao-Xiang Zheng, Yue-Ming Wang, and Shao-Min Zhang. Rapid decoding of hand gestures in electrocorticography using recurrent neural networks. *Frontiers in neuroscience*, 12:555, 2018.

[8] A Faes and MM Van Hulle. Finger movement and coactivation predicted from intracranial brain activity using extended block-term tensor regression. *Journal of Neural Engineering*, 19(6):066011, 2022.

[9] Bingyan Liu, Nuoyan Lv, Yuanchun Guo, and Yawen Li. Recent advances on federated learning: A systematic survey. *Neurocomputing*, page 128019, 2024.

[10] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2017.

[11] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, David Huba, Aleksandr Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Steven Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. In *Proceedings of the 2nd SysML Conference*, 2019.

[12] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. In *NIPS Workshop on Privacy and Security in Machine Learning*, volume 30, 2017.

[13] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[14] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yue Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11, 2019.

[15] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[16] Qiang Yang, Yang Liu, Tianjian Cheng, and Yan Kang. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[17] Andrew Hard, Kanishka Rao, Robert Mathews, Satyen Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloe Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[18] Micah J Sheller, Brandon Edwards, Guido A Reina, James Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Wei Xu, Daniel Marcus, and Spyridon Bakas. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports*, 10(1):1–12, 2020.

[19] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[20] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[21] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE signal processing magazine*, 32(2):145–163, 2015.

[22] Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

[23] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[24] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

[25] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. In *International Conference on Learning Representations*, 2021.

[26] Nanying Liang and Laurent Bougrain. Decoding finger flexion from band-specific ecog signals in humans. *Frontiers in neuroscience*, 6:91, 2012.

[27] Ziqian Xie, Odelia Schwartz, and Abhishek Prasad. Decoding of finger trajectory from ecog using deep learning. *Journal of neural engineering*, 15(3):036009, 2018.

[28] Tatsuya Yokota and Andrzej Cichocki. Multilinear tensor rank estimation via sparse tucker decomposition. In *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, pages 478–483. IEEE, 2014.

[29] Genevera I Allen and Mirjana Maletić-Savatić. Sparse non-negative generalized pca with applications to metabolomics. *Bioinformatics*, 27(21):3029–3035, 2011.

[30] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. *SIAM journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

[31] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[32] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[33] Kai J Miller and Gerwin Schalk. Prediction of finger flexion: 4th brain-computer interface data competition. *BCI Competition IV*, 1:1–2, 2008.

[34] Collin D Binnie. *Clinical neurophysiology: EEG, paediatric neurophysiology, special techniques and applications*, volume 2. Elsevier Health Sciences, 2003.

[35] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[36] Gerwin Schalk and Eric C Leuthardt. Brain-computer interfaces using electrocorticographic signals. *IEEE reviews in biomedical engineering*, 4:140–154, 2011.

[37] A Bolu Ajiboye, Francis R Willett, Daniel R Young, William D Memberg, Brian A Murphy, Jonathan P Miller, Benjamin L Walter, Jennifer A Sweet, Harry A Hoyen, Michael W Keith, et al. Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration. *The Lancet*, 389(10081):1821–1830, 2017.

[38] Francis R Willett, Donald T Avansino, Leigh R Hochberg, Jaimie M Henderson, and Krishna V Shenoy. High-performance brain-to-text communication via handwriting. *Nature*, 593(7858):249–254, 2021.