

# Federated Low-Rank Tensor Estimation for Multimodal Image Reconstruction

Anh Van Nguyen  
Northwestern University  
Evanston, USA  
janguyen@u.northwestern.edu

Diego Klabjan  
Northwestern University  
Evanston, USA  
d-klabjan@northwestern.edu

Minseok Ryu  
Arizona State University  
Tempe, USA  
minseok.ryu@asu.edu

Kibaek Kim  
Argonne National Laboratory  
Lemont, USA  
kimk@anl.gov

Zichao Di  
Argonne National Laboratory  
Lemont, USA  
wendydi@mcs.anl.gov

**Abstract**—Low-rank tensor estimation offers a powerful approach to addressing high-dimensional data challenges and can substantially improve solutions to ill-posed inverse problems, such as image reconstruction under noisy or undersampled conditions. Meanwhile, tensor decomposition has gained prominence in federated learning (FL) due to its effectiveness in exploiting latent space structure and its capacity to enhance communication efficiency. In this paper, we present a federated image reconstruction method that applies Tucker decomposition, incorporating joint factorization and randomized sketching to manage large-scale, multimodal data. Our approach avoids reconstructing full-size tensors and supports heterogeneous ranks, allowing clients to select personalized decomposition ranks based on prior knowledge or communication capacity. Numerical results demonstrate that our method achieves superior reconstruction quality and communication compression compared to existing approaches, thereby highlighting its potential for multimodal inverse problems in the FL setting.

**Index Terms**—federated learning, image reconstruction, tensor decomposition, communication compression, multimodal data

## I. INTRODUCTION

In recent years, tensors (or high-order arrays) have gained considerable traction in data analysis. A rich body of work demonstrates the effectiveness of tensor decompositions in tasks such as predictive modeling [1], [15], [28], image processing [35], feature extraction [10], [26] and data fusion [19], [36], which is the integration of data from different modalities or sources. In particular, the underlying idea of data fusion via tensor decomposition is the joint factorization concept of different datasets into a common latent space.

Meanwhile, motivated by the substantial overhead and burden of sharing large-scale data across multiple entities (such as privacy concerns), Federated Learning (FL) has emerged as an effective decentralized framework. By allowing clients to collaboratively solve a training problem without transmitting raw data, FL significantly reduces data transfer costs. One of the earliest methods, Federated Averaging (FedAvg) [18], exemplifies this approach by aggregating clients' locally trained model weights at the server and distributing the resulting global model back to each client for the next training round. In the past

years, much efforts have been made to extend FL to scientific applications such as tomographic reconstruction [3], [14], [30]. A recent approach, called FIRM [2], is proposed to enable federated tomographic reconstruction from multimodal data by harnessing the complementary relationship between modalities. Furthermore, an increasing number of studies explore the use of tensor decomposition in FL to leverage the benefits of low-rank representations in a decentralized environment. By factorizing model parameters, clients transmit smaller, structured components rather than full-scale weights, thereby mitigating the communication overhead while taking advantage of the low-rank regularization.

In this work, we expand on the federated tomographic problem in [2] by incorporating the Tucker decomposition. Notably, inverse problems, such as tomographic reconstruction, tend to be ill-posed (i.e., many local minima) due to limited data [5], [35]. Low-rank regularization via tensor decomposition helps constrain the solution space by leveraging the underlying low-dimensional structure in the parameter and data space, thereby suppressing noise and artifacts, ultimately improving reconstruction quality [11], [24]. Thus, to enforce low-rank regularization, our method employs a projected gradient descent update for local clients, inspired by [1]. We further exploit the power of tensor decomposition in addressing multimodal data by utilizing joint factorization and randomized sketching in the server aggregation scheme. Our Tucker-based framework is able to significantly improve reconstruction quality and reduce communication overhead. We summarize our contributions as follows.

- We propose a joint factorization approach that streamlines the server aggregation process, eliminating the need to reconstruct full-size tensors while preserving orthogonality of Tucker factor matrices.
- We leverage randomized sketching to reduce the server's computational burden, facilitating large-scale deployment in federated environments.
- Finally, we validate our method with numerical

simulations and achieve superior reconstruction quality under noisy and undersampled settings. Moreover, our method outperforms popular compression techniques such as Top-k and sparse encoding. We also show that with heterogeneous ranks, our method is still able to maintain high quality reconstruction.

## II. BACKGROUND AND RELATED WORKS

### A. Tensor decomposition

Given a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ , its matricization (i.e., mode- $k$  unfolding) is the rearrangement of the  $k$ th dimension into the first dimension of a matrix, and denoted by  $\mathcal{X}_{(k)} \in \mathbb{R}^{n_k \times \prod_{i=1, i \neq k}^d n_i}$ . A tensor can be characterized by its rank and the (multilinear) rank of a tensor consists of the column rank of its mode- $k$  unfolding. Tensor-times-matrix (TTM) in mode- $k$  for tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$  with matrix  $\mathbf{S} \in \mathbb{R}^{n_k \times r}$  is defined as  $\mathcal{Y} = \mathcal{X} \times_k \mathbf{S}$ . The result  $\mathcal{Y}$  is a tensor of shape  $n_1 \times \dots \times n_{k-1} \times r \times n_{k+1} \times \dots \times n_d$  such that  $\mathcal{Y}(i_1, \dots, i_{k-1}, j, i_{k+1}, \dots, i_d) = \sum_{i_k=1}^{n_k} \mathcal{X}(i_1, \dots, i_k, \dots, i_d) \mathbf{S}(i_k, j)$ ,  $j \in [r]$ . Furthermore, given  $\mathbf{S}_k \in \mathbb{R}^{n_k \times r_k}$ ,  $k \in [d]$ , if  $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{S}_1 \times_2 \dots \times_d \mathbf{S}_d$ , the mode- $k$  unfolding  $\mathcal{Y}_{(k)} = \mathbf{S}_k \mathcal{X}_{(k)} (\mathbf{S}_d \otimes \dots \otimes \mathbf{S}_{k+1} \otimes \mathbf{S}_{k-1} \otimes \dots \otimes \mathbf{S}_1)^T$  where  $\otimes$  denotes the Kronecker product of two matrices.

Tensor representation is a powerful tool for handling complex high-dimensional datasets and models. Among the many tensor decompositions, Tucker decomposition is particularly popular because it can be considered an extension of the Singular Value Decomposition (SVD) to higher dimensions [12]. The Tucker rank- $(r_1, \dots, r_d)$  decomposition of tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$  is defined as  $\mathcal{X} \approx \llbracket \mathcal{G}; \mathbf{S}_1, \dots, \mathbf{S}_d \rrbracket \triangleq \mathcal{G} \times_1 \mathbf{S}_1 \times_2 \dots \times_d \mathbf{S}_d$ , where  $\mathcal{G} \in \mathbb{R}^{r_1 \times \dots \times r_d}$  is called the core tensor and  $\mathbf{S}_k \in \mathbb{R}^{n_k \times r_k}$ ,  $k \in [d]$  is a set of factor matrices as the  $r_k$  leading left singular vectors of the mode- $k$  unfolding of  $\mathcal{X}$ . Note that this decomposition is not necessarily unique. More detailed discussions of tensors can be found in [12].

Tucker decomposition can be achieved via the higher-order SVD (HOSVD) algorithm [12]. The sequentially truncated HOSVD (ST-HOSVD) is proposed to improve the computational efficiency of its predecessor [27]. Unlike the original algorithm, which treats each dimension independently, ST-HOSVD finds a factor matrix and immediately compresses the tensor in the corresponding dimension. The truncated tensor is then used to find the factor matrix in the next dimension.

Recently, randomized sketching is used to further reduce the cost of SVD. The key idea is to reduce the dimensions of a matrix by generating a randomized sketch, which is smaller but is a good approximation of the original matrix [8]. In particular, the Rand-Tucker algorithm [34] obtains a sketch  $\mathbf{Z}$  from the mode- $k$  unfolding of tensor  $\mathcal{X}$  and sets the factor matrix  $\mathbf{S}_k$  as the orthonormal basis of  $\mathbf{Z}$  (via QR decomposition). In this work, we adopt the core idea of Rand-Tucker and modify the randomized procedure to the federated setting.

The framework that underpins most tensor-based data analysis tasks is the generalized low-rank tensor estimation problem [9]. Given a dataset  $D$ , this problem is described as

$$\min_{\mathcal{X}} f(\mathcal{X}; D) \quad \text{s.t.} \quad \text{rank}(\mathcal{X}_{(k)}) \leq r_k, k \in [d] \quad (1)$$

where  $f(\mathcal{X}; D)$  is a loss function, and tuple  $(r_1, \dots, r_d)$  is the expected (multilinear) rank of the tensor of interest. In general, this problem can be solved by the projected gradient descent method (e.g., [1]) where the gradient descent iterate is projected onto the feasible set via tensor decomposition such as HOSVD to satisfy the multilinear rank constraint.

### B. Tensor decomposition in FL

Tensor decomposition in the FL framework has previously been studied [20], [21], [31]. A key aspect in the integration of tensor decomposition in FL is how the server aggregates the factorized components. Mainly, there are two aggregation schemes: the server either reconstructs the full weights before taking the average [4], [13], [16], or adapts FedAVG and directly averages the components [22], [32], [33]. A drawback of the former scheme is the significant increase in computational efforts. The latter combines latent spaces that are independently calculated from separate datasets and can suffer from compounding errors. As noted in [29], a potential remedy is joint factorization, whereby these tensors are decomposed into a shared latent factor space. Indeed, joint factorization is a popular approach in data fusion [19], [36] as it provides a systemic framework for merging data obtained from different sources. However, it is still underutilized as an aggregation scheme in FL. In this work, we benchmark against the current aggregation schemes and address their shortcomings using joint factorization.

Mathematically, given  $N$  tensors  $\mathcal{X}^i \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , let their concatenation be denoted by the tensor  $\mathcal{X}$  such that  $\mathcal{X} = [\mathcal{X}^1 \mid \mathcal{X}^2 \mid \dots \mid \mathcal{X}^N]$  and  $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_d \times N}$ . The joint factorization of  $N$  tensors  $\mathcal{X}^i$  is equivalent to finding the factor matrices of the concatenated tensor  $\mathcal{X}$  in the first  $d$  dimensions. Gao et al. [7] propose the first joint factorization-based scheme for the federated feature extraction task. The authors leverage the result in Lemma II.1 to approximate the left singular vectors of the mode- $k$  unfolding  $\mathcal{X}_{(k)}$  as those of  $\mathbf{Y} = [\mathbf{S}_k^1 \mathcal{G}_{(k)}^1 \dots \mathbf{S}_k^N \mathcal{G}_{(k)}^N]$ , which effectively bypasses the reconstruction of full-size tensors.

**Lemma II.1.** (From Gao et al. [7]). For  $i = 1, \dots, N$  and  $k = 1, \dots, d$ , let  $\mathcal{G}^i \in \mathbb{R}^{r_1 \times \dots \times r_d}$  and  $\mathbf{S}_k^i \in \mathbb{R}^{n_k \times r_k}$  such that the columns of  $\mathbf{S}_k^i$  are orthogonal. Let  $\mathcal{X}^i = \llbracket \mathcal{G}^i; \mathbf{S}_1^i, \dots, \mathbf{S}_d^i \rrbracket$ ,  $\mathbf{W} = [\mathcal{X}_{(k)}^1 \dots \mathcal{X}_{(k)}^N]$  and  $\mathbf{Y} = [\mathbf{S}_k^1 \mathcal{G}_{(k)}^1 \dots \mathbf{S}_k^N \mathcal{G}_{(k)}^N]$ . Then  $\mathbf{W}$  and  $\mathbf{Y}$  have the same set of singular values and if  $\mathbf{U}$  and  $\bar{\mathbf{U}}$  are the left singular matrix of  $\mathbf{W}$  and  $\mathbf{Y}$  respectively, then  $\mathbf{U} = \bar{\mathbf{U}} \mathbf{P}$  where  $\mathbf{P}$  is a unitary block diagonal matrix.

Similarly to [7], we exploit the orthogonality of the factor matrices in the aggregation process and extend it by incorporating randomized sketching to improve computational efficiency. Note that [7] focuses on feature extraction while we focus on a multimodal inverse problem, which requires new concepts, ideas, and tricks.

### C. Federated tomographic reconstruction

In this paper, we extend the multimodal image reconstruction problem in [2], which is given by

$$\min_{\mathbf{x}^1, \dots, \mathbf{x}^N} \sum_{i=1}^N f(\mathbf{X}^i; D^i) \quad \text{s.t.} \quad \mathbf{X}_N = \sum_j^{N-1} c_j \mathbf{X}_j^j \quad (2)$$

where  $\mathbf{X}^N$  denotes the client with X-ray transmission (XRT) data and  $\mathbf{X}^j$ ,  $j \in [N-1]$  are clients with X-ray fluorescence (XRF) data. The coefficient  $c_j$  denotes the mass attenuation coefficient of the element of interest and is well characterized.

FIRM, the approach developed in [2], enables clients with data obtained from different modalities to collaborate and improve their respective solution quality with convergence guarantee [2]. Specifically, at epoch  $t$ , clients use gradient descent to locally update the previous solution  $\mathbf{X}^i(t-1)$  and obtain  $\tilde{\mathbf{X}}^i$ ,  $i \in [N]$ . After receiving the full-size tensors  $\tilde{\mathbf{X}}^i$ , the server performs a series of arithmetic operations to enforce the multimodality constraint as follows.

$$\begin{aligned} \Sigma &\leftarrow \sum_{i=1}^{N-1} c_i \tilde{\mathbf{X}}^i \\ \mathbf{X}^i(t) &\leftarrow \tilde{\mathbf{X}}^i + \frac{c_i}{2} (\tilde{\mathbf{X}}^N - \Sigma), i \in [N-1] \\ \mathbf{X}^N(t) &\leftarrow \frac{\tilde{\mathbf{X}}^N + \Sigma}{2} \end{aligned} \quad (3)$$

Previous works have shown that the low-rank regularization via tensor decomposition improves standard tomographic reconstruction quality [11]. In this paper, we introduce low-rank regularization and incorporate tensor decomposition to the FIRM framework to further improve the solution of Problem 2.

### III. METHODOLOGY

In this section, we build on the projected gradient descent approach from [1] and develop a joint factorization scheme for the server; we also adapt the FIRM update (3) to enforce the multimodality constraint in the Tucker component space.

#### A. Algorithm

We consider a constrained low-rank tensor estimation problem in the federated setting as follows,

$$\begin{aligned} \min_{\mathbf{x}^1, \dots, \mathbf{x}^N} \sum_i^N f(\mathbf{X}^i; D^i) \\ \text{s.t.} \quad \text{rank}(\mathbf{X}_{(k)}^i) \leq r_k \quad k \in [d], i \in [N] \\ \mathbf{X}^i = \llbracket \mathcal{G}^i; \mathbf{S}_1, \dots, \mathbf{S}_d \rrbracket \quad i \in [N] \\ \mathbf{S}_k^T \mathbf{S}_k = \mathbf{I}_{r_k} \quad k \in [d] \\ \mathcal{G}^N = \sum_j^{N-1} c_j \mathcal{G}^j \end{aligned} \quad (4)$$

where the core tensors  $\mathcal{G}^i \in \mathbb{R}^{r_1 \times \dots \times r_d} \forall i \in [N]$  are client-specific, the factor matrices  $\mathbf{S}_k \in \mathbb{R}^{n_k \times r_k} \forall k \in [d]$  are shared across local clients, and  $\mathbf{I}_{r_k} \in \mathbb{R}^{r_k \times r_k}$  is the identity matrix.

This structure ensures that while clients retain their data locally, global knowledge can still be learned collaboratively. Note that the linear constraint in Problem 2 naturally applies to  $\mathcal{G}^i$ . The algorithm is provided in Algorithm 1.

At local clients, we use the projected gradient descent method to optimize the low-rank tensor estimation problem (1), where ST-HOSVD is used for projection due to its robustness comparing to HOSVD. With this local update, clients obtain their independent set of factor matrices and the corresponding cores. For the global stage on the server, we develop an aggregation scheme based on joint factorization to combine the latent spaces across clients while maintaining the orthogonality condition  $\mathbf{S}_k^T \mathbf{S}_k = \mathbf{I}_{r_k}$ ,  $k \in [d]$ . We also utilize randomized sketching in the factorization procedure for better computational efficiency. To enforce the last constraint in (4), we leverage the FIRM update to aggregate the cores. Our aggregation scheme avoids reconstructing full-size tensors and facilitates heterogeneous ranks, compared to other works [4], [16], [22], [32], [33]. Since both the server and the clients send low-rank components, our method promotes compression for both upstream and downstream communication.

---

#### Algorithm 1: Federated Low-rank Tensor Estimation with Joint Factorization

---

**Server Input :** Tucker rank  $(r_1, \dots, r_d)$ , number of epochs  $T$ , learning rate  $\eta$

**Client Input :** Dataset  $D^i$ ,  $i \in [N]$

- 1 **Server** initializes and broadcasts to clients  $\mathcal{G}^i(0)$ ,  $i \in [N]$  and  $\mathbf{S}_k(0)$ ,  $k \in [d]$ ;
- 2 **for**  $t = 1, \dots, T$  **do**
- 3     **CLIENT**  $i = 1, \dots, N$ :
- 4      $\mathbf{X}^i(t) \leftarrow \mathcal{G}^i(t-1) \times_1 \mathbf{S}_1(t-1) \times_2 \dots \times_d \mathbf{S}_d(t-1)$ ;
- 5      $\tilde{\mathbf{X}}^i \leftarrow \mathbf{X}^i(t) - \eta \nabla f(\mathbf{X}^i(t); D^i)$ ;
- 6      $\hat{\mathcal{G}}^i, \hat{\mathbf{S}}_1^i, \dots, \hat{\mathbf{S}}_d^i \leftarrow \text{ST-HOSVD}(\tilde{\mathbf{X}}^i)$ ;
- 7     Send  $\hat{\mathcal{G}}^i, \hat{\mathbf{S}}_1^i, \dots, \hat{\mathbf{S}}_d^i$  to the Server;
- 8     **SERVER:**
- 9      $\mathbf{S}_1(t), \dots, \mathbf{S}_d(t) \leftarrow \text{Joint-Factorization}(\hat{\mathcal{G}}^i, \hat{\mathbf{S}}_1^i, \dots, \hat{\mathbf{S}}_d^i, i \in [N])$ ;
- 10     Broadcasts  $\mathbf{S}_k(t)$ ,  $k \in [d]$  to all clients;
- 11     **for**  $i = 1, \dots, N$  **do**
- 12     |  $\tilde{\mathcal{G}}^i \leftarrow \llbracket \hat{\mathcal{G}}^i; \mathbf{S}_1(t)^T \hat{\mathbf{S}}_1^i, \dots, \mathbf{S}_d(t)^T \hat{\mathbf{S}}_d^i \rrbracket$ ;
- 13     **end**
- 14      $\Sigma \leftarrow \sum_{i=1}^{N-1} c_i \tilde{\mathcal{G}}^i$ ;
- 15      $\mathcal{G}^i(t) \leftarrow \tilde{\mathcal{G}}^i + \frac{c_i}{2} (\tilde{\mathcal{G}}^N - \Sigma)$ ,  $i \in [N-1]$ ;
- 16      $\mathcal{G}^N(t) \leftarrow \frac{\tilde{\mathcal{G}}^N + \Sigma}{2}$ ;
- 17     Sends  $\mathcal{G}^i(t)$  to client  $i$ ;
- 18 **end**

---

Let  $\mathcal{G}^i(t)$ ,  $i \in [N]$  and  $\mathbf{S}_k(t)$ ,  $k \in [d]$  denote the cores and factor matrices obtained at the end of epoch  $t$ . In lines 4-6 of Algorithm 1, a client  $i$  constructs the full tensor  $\mathbf{X}^i(t)$  from the core  $\mathcal{G}^i(t-1)$  and factor matrices  $\mathbf{S}_k(t-1)$ ,  $k \in [d]$

received from the server. The client solves the local low-rank optimization problem by updating its respective tensor  $\mathcal{X}^i(t)$  using gradient descent to obtain  $\tilde{\mathcal{X}}^i$  and applying ST-HOSVD to this new tensor. The client then sends the factorized components  $\hat{\mathcal{G}}^i, \hat{\mathbf{S}}_1^i, \dots, \hat{\mathbf{S}}_d^i$  to the server.

To enforce the first two constraints in (4), the server performs joint factorization. One option is to directly apply Lemma II.1 from [7] so that the server can perform joint factorization without full-size tensors (as summarized in Algorithm 2). However, it involves the SVD calculation of large matrices  $\mathbf{Y}_k$

---

**Algorithm 2:** Joint factorization on the server at epoch  $t$  (Direct application of [7])

---

**Input:** Tucker rank  $(r_1, \dots, r_d)$   
**Input:**  $\hat{\mathcal{G}}^i, \hat{\mathbf{S}}_1^i, \dots, \hat{\mathbf{S}}_d^i, i \in [N], t$   
**Output:**  $\mathbf{S}_k(t), k \in [d]$   
1 **for**  $k = 1, \dots, d$  **do**  
2      $\mathbf{Y}_k \leftarrow \begin{bmatrix} \hat{\mathbf{S}}_k^1 \hat{\mathcal{G}}_{(k)}^1 & \dots & \hat{\mathbf{S}}_k^N \hat{\mathcal{G}}_{(k)}^N \end{bmatrix};$   
3      $\mathbf{S}_k(t) \leftarrow r_k$  left singular vectors of  $\mathbf{Y}_k$ ;  
4 **end**

---

of size  $n_k \times N \prod_{j \neq k}^d r_j$ , which can be very expensive as the number of clients and dimensions grow.

Therefore, to further improve efficiency, we consider QR decomposition-based randomized Tucker decomposition (as proposed in [34]) and adapt it to the federated setting as follows.

**Lemma III.1.** *Suppose that  $\mathcal{X}^i = [\mathcal{G}^i; \mathbf{S}_1^i, \dots, \mathbf{S}_d^i]$  where  $\mathbf{S}_k^i \mathbf{S}_k^i{}^T = \mathbf{I}_{r_k} \forall i \in [N], \forall k \in [d]$ . Let  $\mathbf{W}_k = [\mathcal{X}_{(k)}^1 \dots \mathcal{X}_{(k)}^N]$ . If  $\mathbf{Z}_k = \mathbf{W}_k \mathbf{\Omega}_k$  for a Gaussian matrix  $\mathbf{\Omega}_k$  of size  $N \prod_{j \neq k}^d n_j \times r_k$ , then  $\mathbf{Z}_k = \sum_{i=1}^N \mathbf{S}_k^i \mathcal{G}_{(k)}^i \mathbf{\Omega}_k^i$  for some  $\mathbf{\Omega}_k^i$  Gaussian matrices of size  $\prod_{j \neq k}^d r_j \times r_k$ .*

*Proof.* We rewrite  $\mathbf{\Omega}_k = \begin{bmatrix} \bar{\mathbf{\Omega}}_k^1 \\ \vdots \\ \bar{\mathbf{\Omega}}_k^N \end{bmatrix}$  where  $\bar{\mathbf{\Omega}}_k^i \in \mathbb{R}^{\prod_{j \neq k}^d n_j \times r_k}$ .

We have  $\mathbf{W}_k \mathbf{\Omega}_k = \sum_{i=1}^N \mathbf{W}_{(k)}^i \bar{\mathbf{\Omega}}_k^i$ . In addition, since  $\mathcal{X}^i = [\mathcal{G}^i; \mathbf{S}_1^i, \dots, \mathbf{S}_d^i]$ , the mode- $k$  unfolding is  $\mathbf{W}_{(k)}^i = \mathbf{S}_k^i \mathcal{G}_{(k)}^i \mathbf{V}_k^i{}^T$  where  $\mathbf{V}_k^i = (\mathbf{S}_d^i \otimes \dots \otimes \mathbf{S}_{k+1}^i \otimes \mathbf{S}_{k-1}^i \otimes \dots \otimes \mathbf{S}_1^i)$ . Because  $\mathbf{S}_k^i \mathbf{S}_k^i{}^T = \mathbf{I}_{r_k}$ , it is easy to show that  $\mathbf{V}_k^i$  also has orthogonal columns.

We now have  $\mathbf{W}_k \mathbf{\Omega}_k = \sum_{i=1}^N \mathbf{S}_k^i \mathcal{G}_{(k)}^i \mathbf{V}_k^i{}^T \bar{\mathbf{\Omega}}_k^i$ . Let  $\mathbf{\Omega}_k^i = \mathbf{V}_k^i{}^T \bar{\mathbf{\Omega}}_k^i$ . Using the property of the Gaussian distribution, it is readily observed that  $\mathbf{\Omega}_k^i$  is also a Gaussian matrix with each element independently drawn from the normal distribution.  $\square$

Finally, we propose Algorithm 3 that combines the result in Lemma III.1 (shown in lines 2-3) and QR decomposition. Compared to Algorithm 2, we now decompose a smaller matrix of size  $n_k \times r_k$ , which is independent of the number of clients and dimensions. Note that with randomized factorization, we essentially sample from the column space of a matrix, so as  $r_k$  decreases, the randomized sketch becomes a less accurate approximation of the original tensor [8]. This in turn

can introduce more errors in subsequent QR decomposition. Therefore, if we choose small values for the Tucker ranks, the loss in accuracy due to randomization may outweigh the gain in computational efficiency. In this case, it is advisable to use Algorithm 2 instead. We name our method Component Randomized Joint Factorization (i.e., CompRandJF) with the use of Algorithm 3 for joint factorization in Algorithm 1. Alternatively, we term the method using Algorithm 2 for joint factorization in Algorithm 1 as Component Joint Factorization (i.e., CompJF).

---

**Algorithm 3:** Randomized joint factorization on the server at epoch  $t$

---

**Input:** Tucker rank  $(r_1, \dots, r_d)$   
**Input:**  $\hat{\mathcal{G}}^i, \hat{\mathbf{S}}_1^i, \dots, \hat{\mathbf{S}}_d^i, i \in [N], t$   
**Output:**  $\mathbf{S}_k(t), k \in [d]$   
1 **for**  $k = 1, \dots, d$  **do**  
2      $\mathbf{\Omega}_k^i \leftarrow i$ 'th i.i.d. sample of a Gaussian matrix of size  $\prod_{j \neq k}^d r_j \times r_k, i \in [N];$   
3      $\mathbf{Y}_k \leftarrow \sum_{i=1}^N \hat{\mathbf{S}}_k^i \hat{\mathcal{G}}_{(k)}^i \mathbf{\Omega}_k^i;$   
4      $\mathbf{S}_k(t) \leftarrow \mathbf{Q}$  from QR decomposition of  $\mathbf{Y}_k$ ;  
5 **end**

---

To satisfy the last constraint in (4), the server first recomputes the cores after obtaining the updated factor matrices (see lines 9–11 of Algorithm 1), then applies the FIRM update (3), adapted specifically for the cores.

**Remark.** *Heterogeneous ranks can arise when clients have varying bandwidths or have prior knowledge about the appropriate rank of the local data. Our joint factorization scheme for the server allows heterogeneous ranks, as it does not require dimensions to match and clients can choose their own Tucker rank  $(r_1^i, \dots, r_d^i)$ . Essentially, Algorithms 2 and 3 can be adjusted to find the  $r_k^*$  leading left singular vectors where  $r_k^* = \max(r_k^1, \dots, r_k^N), k \in [d]$ .*

## B. Complexity Analysis

In this section, we discuss the complexity regarding the communication of our proposed method. Without loss of generality, we assume that  $n = n_1 = \dots = n_d$  and  $r = r_1 = \dots = r_d$ . For a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , the communication complexity of transmitting the full-size tensor is  $O(n^d)$ . On the other hand, communicating the rank- $(r, \dots, r)$  Tucker decomposition reduces the complexity to  $O(r^d + dnr)$ .

A key challenge in utilizing tensor decomposition for communication compression is the choice of rank  $r$  that maximizes the compression ratio (defined as  $\phi = \frac{n^d}{r^d + dnr}$ ) while maintaining performance. Dai et al. [4] give an upper bound for the Tucker rank that guarantees a reduction in communication cost per epoch, specifically  $r \leq \frac{n}{(1 + dn)^{1/d}}$ . While this bound provides practical guidance, it is overly restrictive in the range of  $r$ . For example, when  $n = 250$  and  $d = 2$ , the bound suggests  $r \leq 11$ . Such aggressive

compression, and consequently regularization on the underlying structure of the solution, can have a negative impact on the reconstruction quality.

To address this limitation, it is important to find a broader range of values for  $r$ . Proposition 1 provides less restrictive bounds for  $d = 2$  and  $d = 3$  that allow clients to choose a much larger Tucker rank, compared to [4]. For the same example, the acceptable range of  $r$  increases to  $r \leq 103$ . Proposition 1 further shows that as  $d$  and  $n$  increase, the communication overhead savings with Tucker decomposition increase substantially.

**Proposition 1.** *If  $n \geq 3$ , then the improved upper bound reads*

- $r < \frac{n}{\sqrt{2}+1}$  for  $d = 2$ , and
- $r < n \left(\frac{n-3}{n}\right)^{1/3}$  for  $d = 3$ .

*Furthermore, for both values of  $d$ , these bounds are greater than  $\frac{n}{(1+dn)^{1/d}}$ .*

*Proof.* Let  $\beta = \frac{r}{n}$ , resulting in  $\phi = \frac{1}{\beta^d + (d\beta)/n^{d-2}}$ . Note that  $0 < \beta < 1$  since  $r < n$ .

For  $d = 2$ ,  $\phi = \frac{1}{\beta^2 + 2\beta}$ . If  $\beta < 1/(\sqrt{2}+1)$ , then  $\beta^2 + 2\beta < 1$  and  $\phi > 1$ . Since  $(1+dn)^{1/d} \geq \sqrt{5} > \sqrt{2} + 1$ , the bound  $\frac{n}{\sqrt{2}+1} > \frac{n}{(1+2n)^{1/2}}$ .

For  $d = 3$ ,  $\phi = \frac{1}{\beta^3 + 3\beta/n}$ . If  $\beta < \left(\frac{n-3}{n}\right)^{1/3}$ , we have  $\beta^3 + 3\frac{\beta}{n} < \frac{n-3}{n} + \frac{3}{n} = 1$  and  $\phi > 1$ . Since  $1 + 3n > 1 + \frac{3}{n-3} = \frac{n}{n-3}$ , the bound  $n \left(\frac{n-3}{n}\right)^{1/3} > \frac{n}{(1+3n)^{1/3}}$ .  $\square$

We also compare the computational complexity of CompJF and CompRandJF with the baseline approach, where the server reconstructs full-size tensors to aggregate before applying tensor decomposition (termed as FullDecomp). We summarize the operation counts in Table I.

**TABLE I:** Computational complexity of the server per epoch

Approach	Complexity
FullDecomp	$O(N \sum_{k=1}^d (n^{d-k+1} r^k + n^{d-k+2} r^{k-1} + n^{d-k} r^k) + Nn^d)$
CompJF (Alg. 1 & 2)	$O(Nd(nr^d + n^2 r^{d-1}) + N(dnr^2 + dr^{d+1}) + Nr^d)$
CompRandJF (Alg. 1 & 3)	$O(Nd(r^d + r^{d+1} + nr^2 + nr) + dnr^2 + N(dnr^2 + dr^{d+1}) + Nr^d)$

#### IV. EXPERIMENTS AND RESULTS

We evaluate the performance of CompJF and CompRandJF using synthetic data. To show the benefits of our aggregation schemes, we benchmark them against FIRM and FullDecomp, where the server reconstructs full-size tensors to aggregate before applying tensor decomposition for downstream communication. We also experiment with the naive extension of FedAVG in which factor matrices are averaged across clients but this aggregation scheme does not yield meaningful training.

In addition, we demonstrate the communication efficiency achieved by CompJF and CompRandJF. For benchmarking, we apply to FIRM the compressed sparse row (CSR) encoding and Top- $k$  sparsification. Top- $k$  is a magnitude-based sparsification method and has been shown to effectively reduce communication overhead [17], [23]. For Top- $k$  sparsification, we first sort all the values in an array and only keep the the top  $k\%$  of the values; the remaining values are set to zero. Note that Top- $k$  sparsification produces sparse arrays and can further benefit from sparse encoding. Thus, the two benchmarks for communication efficiency are FIRM with only CSR encoding and FIRM with a combination of Top- $k$  and CSR encoding. We refer to the second benchmark as Top- $k$  for short.

Finally, we highlight the ability of our methods to facilitate heterogeneous ranks in which clients determine their own Tucker decomposition ranks.

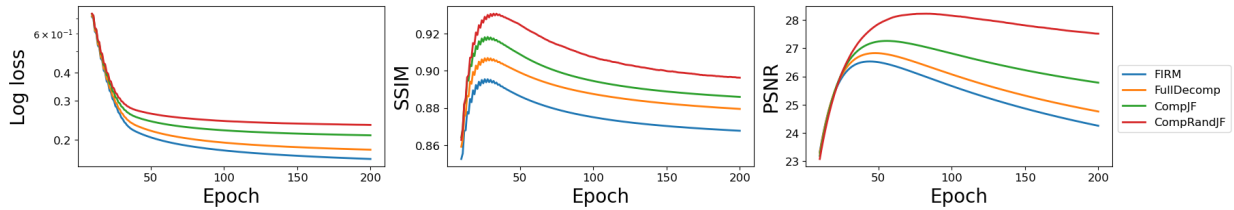
#### A. Experimental settings

Tomographic data, or sinograms, are generated from scanning an object using an energy source with  $\theta$  angles and  $\tau$  discretized beamlets for each angle. Let the tensor  $\mathcal{A} \in \mathbb{R}^{\theta \times \tau \times n_1 \times n_2}$  represent the discrete Radon transform and  $\mathcal{A}_{\theta, \tau, i, j}$  is the intersection length of the angle-beamlet pair  $(\alpha, \beta)$  with the pixel  $(i, j)$ . Let the tensor to be reconstructed and the observed measurements of each client  $i$  be denoted by  $\mathcal{X}^i \in \mathbb{R}^{n_1 \times \dots \times n_d}$  and  $\mathcal{B}^i \in \mathbb{R}^{\theta \times \tau \times n_3 \times \dots \times n_d}$ . The loss function in our tensor estimation problem becomes  $f = \sum_{i=1}^N \|\mathcal{A} \boxtimes \mathcal{X}^i - \mathcal{B}^i\|_F^2$  where  $N$  is the total number of local clients and  $\boxtimes$  denotes the *contracted tensor product* of two tensors.

We evaluate the proposed method on synthetic data, which are generated with parallel geometry and are variants of the Shepp-Logan phantom, a standard image for testing in computed tomography [6]. Following the setting in [2], we experiment with  $N = 4$  clients, one holding data obtained from the *XRT* modality and the other with data produced by the *XRF* modality. The ground truth  $\mathcal{X}^i$  to be reconstructed is of size  $250 \times 250$ , and the Radon transform has  $\tau = 354$  beamlets and  $\theta = 100$  angles. Note that with this number of angles, the measurements are undersampled and the inverse problem is ill-posed.

We experiment with adding speckle noise, which is inherent in the imaging process, to the sinograms, such that  $output = true\ signal + \epsilon * true\ signal$  and  $\epsilon$  is normally distributed. We vary the standard deviation of the added noise as  $\sigma \in \{0.01, 0.05, 0.1\}$ .

For CompJF, CompRandJF and FullDecomp, we use Proposition 1 to obtain the upper bound of the decomposition rank  $r$  given that  $n = 250$ . The results are obtained with  $r \in \{10, 20, 40, 60, 80, 100\}$ . In the heterogeneous rank setting, each client randomly samples a value of  $r$  in the range of [20, 100]. There are two possible scenarios of heterogeneous ranks. Clients a) have prior knowledge of the appropriate rank or fixed communication limit, or b) have unstable communication bandwidths, so the Tucker ranks need adjusting each communication round. Consequently, in our experiments, the Tucker ranks can be a) sampled before the federated



**Fig. 1:** Evaluating the performance of various methods given the strongest noise level ( $\sigma = 0.1$ ). For CompJF, CompRandJF and FullDecomp, we present the results for Tucker rank  $r = 100$ , the largest value that guarantee communication compression. The results for other ranks  $r \geq 40$  is similar.

reconstruction starts or b) resampled every epoch prior to upstream communication.

### B. Performance metrics

1) *Reconstruction quality:* We compare the solutions to the ground-truth images by evaluating two metrics as the Peak Signal-to-Noise Ratio (PSNR) and the multiscale Structural Similarity Index Measure (SSIM). PSNR is inversely proportional to the mean squared error (MSE) and SSIM computes similarities in structural information and incorporates interdependencies among pixels. SSIM is bounded between  $-1$  and  $1$  and when  $SSIM = 0$ , there is no similarity between two images. For both quantities, higher values indicate better reconstruction.

2) *Communication efficiency:* We are interested in measuring the trade-off between reconstruction quality and communication cost. Song et al. [25] propose Gamma Communication Efficiency  $GCE = \frac{\text{Accuracy}}{(1 - \text{Accuracy})^\gamma \sum_{t=1}^T \log_2(V_t + 1)}$  in which  $T$  denotes the number of communication rounds and  $V_t$  is the communication volume, measured in bits. The parameter  $\gamma$  controls the importance of test accuracy in relation to communication, and the two quantities are near proportional if  $\gamma \rightarrow 0$ . A higher value of GCE indicates a better balance between communication and accuracy, which is more desirable. We consider image quality and communication compression to be of equal importance and use  $\gamma = 0.01$ , as in [25], to analyze our results. We also adapt the metrics to images by replacing the test accuracy with SSIM as the SSIM values in our experiments fall in the range  $[0, 1]$ . To calculate  $V_t$ , we combine upstream and downstream communication.

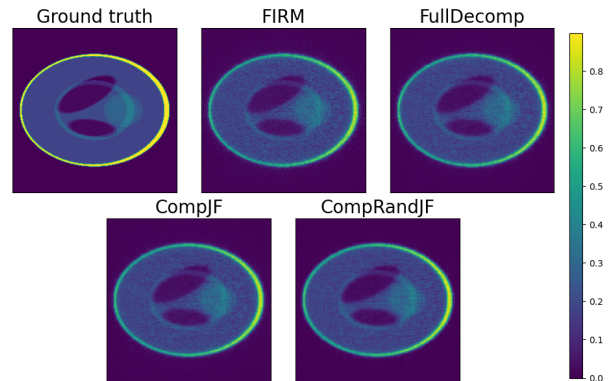
### C. Results

1) *Robustness against noisy data:* We demonstrate in Fig. 1 that CompJF and CompRandJF produce better reconstruction than other approaches in the presence of noise. To understand the effect of noise, we plot the results for the strongest noise level ( $\sigma = 0.1$ ) and use the largest Tucker rank that guarantee communication compression ( $r = 100$ ).

Overfitting to noise, combined with the ill-posed nature of the inverse problem, often leads to a degradation of image quality as the optimization progresses. This is evident in Fig. 1 as both PSNR and SSIM peak before 100 epochs and then decrease over time, while the loss remains non-increasing. However, the low-rank decomposition coupled with joint factorization (as enforced in CompJF and CompRandJF) can mitigate this

effect, and result in a more graceful degradation. This behavior aligns with results from previous works in which low-rank regularization helps constrain the solution space of the inverse problem, suppressing noise and artifacts in the subsequent reconstructed images. This also explains why the other three methods lead to better quality metrics compared to FIRM despite having worse loss function.

We also show the ground truth and the reconstructed images from various methods in Fig. 2. The quality of the images aligns with the numerical results in Fig. 1. It is visible that CompJF and CompRandJF have additional denoising effect on the reconstructed images compared to FIRM.

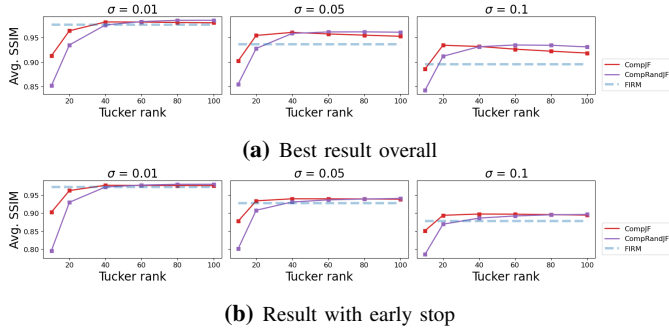


**Fig. 2:** Ground-truth and reconstructed images from various methods for client  $N$  with XRT imaging modality. The images are reconstructed from noisy data ( $\sigma = 0.1$ ). For FullDecomp, CompJF, and CompRandJF, we use Tucker rank  $r = 100$ . The reconstructed images correspond to the highest SSIM values.

2) *Stable performance across Tucker ranks:* In addition to better reconstruction, CompJF and CompRandJF are able to maintain good performance as we decrease the Tucker rank and further compress the data. We present the best result for each rank value as well as the result induced by early stopping in Fig. 3. A common solution to overfitting in tomographic reconstruction is to apply the discrepancy principle, which is an early stopping rule and is widely used in the literature. We use the formulation of the discrepancy principle as in [2] so that the optimization stops when the condition  $f^i(t) \leq \max(\mathcal{B}^i) \sqrt{\theta \tau \sigma}$  is satisfied for every client, where  $\mathcal{B}^i$  are the observed measurements,  $\theta$  is the number of projection angles,  $\tau$  is the number of beamlets and  $\sigma$  is the standard deviation of added noise.

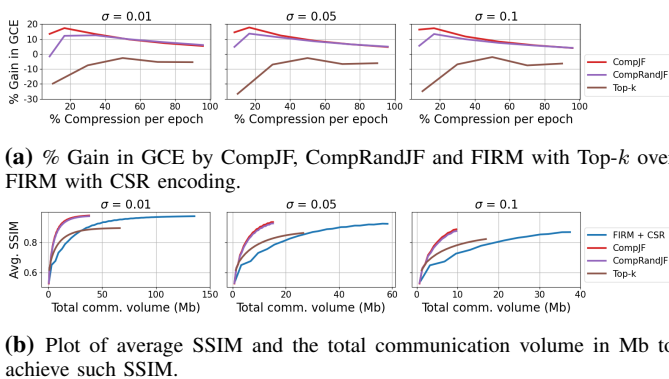


We observe that our proposed approaches produce better results than FIRM for Tucker ranks  $r \geq 40$ , especially as the data become more noisy. This performance holds when early stopping is incorporated. For  $r < 40$ , CompJF exhibits a more gradual decrease in performance compared to CompRandJF. This is due to the randomized sketch used in CompRandJF. As  $r$  further decreases, this sketch is no longer a good approximation of the original matrix and leads to loss of information in the factorization process.



**Fig. 3:** Performance of CompJF and CompRandJF for varying Tucker decomposition ranks and varying level of noise. (a) Includes the highest SSIM value, averaged across clients, obtained by the approaches. (b) Includes the average SSIM obtained when the discrepancy principle is satisfied. With early stopping condition, all three approaches terminate at the same epoch.

3) **Communication compression:** We next illustrate the superior ability to balance reconstruction quality and communication compression of our methods, compared to Top- $k$  sparsification and CSR encoding. With Top- $k$  sparsification, the parameter  $k$  controls the percentage of compression in each communication round, and we vary  $k \in \{10, 30, 50, 70, 90\}$ . For easier comparison, we convert the chosen Tucker ranks to the corresponding percentage of compression. For example, given that  $n = 250$ , if  $r = 40$ , we have a 34.56% compression.

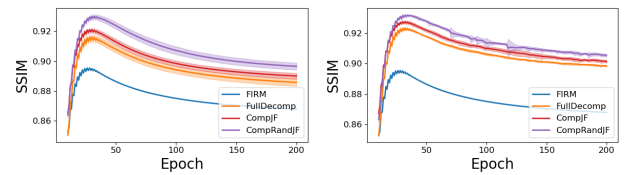


**Fig. 4:** Performance of our methods, CompJF and CompRandJF, Top- $k$  and CSR encoding in balancing reconstruction quality with communication compression. a) GCE is computed using the SSIM values when the early stopping is satisfied and is averaged across clients. b) For CompJF and CompRandJF, the Tucker rank is  $r = 40$  and for Top- $k$ , we use  $k = 30$ . With this choice of hyperparameters, our methods and Top- $k$  have close percentage of compression in each communication round (30% and 34%).

Fig. 4a shows the percentage gain in GCE with respect to

the CSR encoding and 4b highlights the total communication volume needed to achieve high quality reconstruction. The results in Fig. 4 indicate that CompJF and CompRandJF lead to a substantial gain in GCE compared to full-size communication enhanced with CSR encoding and surpass Top- $k$ . When  $r = 10$ , the drop in SSIM overshadows the improvement in communication, thus the sharp decline in GCE.

4) **Heterogeneous ranks:** Finally, we demonstrate the ability of CompRandJF, as well as CompJF, to adapt to heterogeneous ranks. This property is an improvement over other aggregation schemes that rely purely on averaging and require matching dimensions. In Fig. 5, we use FullDecomp, adjusted for heterogeneous ranks, and FIRM as baselines for comparison. We observe that the performance of CompJF and CompRandJF is consistent with the homogeneous case regarding reconstruction quality.



(a) Clients sample personalized ranks at the start. (b) Clients sample ranks before every communication round.

**Fig. 5:** Performance of FullDecomp, CompJF and CompRandJF in heterogeneous rank setting for noisy data ( $\sigma = 0.1$ ). We conduct 10 independent simulations and plot the mean with one standard deviation.

## V. CONCLUSION

In this paper, we present a federated image reconstruction method that leverages Tucker decomposition in conjunction with joint factorization and randomized sketching for efficient aggregation. We compare our approach against two benchmarks: 1) FIRM which addresses federated reconstruction without low-rank regularization and transmits high-dimensional data for each update, and 2) FullDecomp which is a common aggregation technique in tensor decomposition for federated learning that reconstructs the full weights for aggregation and optionally re-decomposes them to reduce downstream communication. Our method demonstrates superior reconstruction quality and more graceful performance degradation under noisy and undersampled conditions. In addition, we achieve better compression than magnitude-based sparsification (Top- $k$ ) and compressed sparse row encoding. By supporting both homogeneous and heterogeneous ranks, our approach delivers high-quality reconstruction across diverse scenarios. For future work, we plan to conduct convergence analyses, explore higher-dimensional datasets, and further investigate heterogeneous ranks—particularly relevant given that true data ranks are often unknown in practice, making randomized rank selection a valuable research avenue.

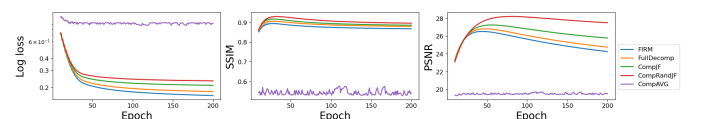
## REFERENCES

- [1] Talal Ahmed, Haroon Raja, and Waheed U Bajwa. Tensor regression using low-rank and sparse tucker decompositions. *SIAM Journal on Mathematics of Data Science*, 2(4):944–966, 2020.

- [2] Geunyeong Byeon, Minseok Ryu, Zichao Wendy Di, and Kibaek Kim. Firm: Federated image reconstruction using multimodal tomographic data. *arXiv preprint arXiv:2501.05642*, 2025.
- [3] Shixuan Chen, Boxuan Cao, Yinda Du, Yaoduo Zhang, Ji He, Zhaoying Bian, Dong Zeng, and Jianhua Ma. Federated condition generalization on low-dose ct reconstruction via cross-domain learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 47–56. Springer, 2023.
- [4] Wei Dai, Jicong Fan, Yiming Miao, and Kai Hwang. Deep learning model compression with rank reduction in tensor decomposition. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [5] Zichao Di, Sven Leyffer, and Stefan M Wild. Optimization-based approach for joint x-ray fluorescence and transmission tomographic inversion. *SIAM Journal on Imaging Sciences*, 9(1):1–23, 2016.
- [6] H Michael Gach, Costin Tanase, and Fernando Boada. 2D & 3D Shepp-Logan phantom standards for MRI. pages 521–526, 2008.
- [7] Yuan Gao, Guangming Zhang, Chunchun Zhang, Jinke Wang, Laurence T Yang, and Yaliang Zhao. Federated tensor decomposition-based feature extraction approach for industrial iot. *IEEE Transactions on Industrial Informatics*, 17(12):8541–8549, 2021.
- [8] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.
- [9] Rungang Han, Rebecca Willett, and Anru R Zhang. An optimal statistical and computational framework for generalized tensor estimation. *The Annals of Statistics*, 50(1):1–29, 2022.
- [10] Guosheng Hu, Yang Hua, Yang Yuan, Zhihong Zhang, Zheng Lu, Sankha S Mukherjee, Timothy M Hospedales, Neil M Robertson, and Yongxin Yang. Attribute-enhanced face recognition with neural tensor fusion networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3744–3753, 2017.
- [11] Sanket R Jantre and Zichao Wendy Di. Low-rank tensor regression for x-ray tomography. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2833–2837. IEEE, 2021.
- [12] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.
- [13] Guangchen Lan, Xiao-Yang Liu, Yijing Zhang, and Xiaodong Wang. Communication-efficient federated learning for resource-constrained edge devices. *IEEE Transactions on Machine Learning in Communications and Networking*, 2023.
- [14] Danyang Li, Hao Wang, Jingyi Liao, Mingqiang Meng, Cuidie Zeng, Yuting Wang, Dong Zeng, and Jianhua Ma. Semi-centralized federated learning network for low-dose CT imaging. In *Medical Imaging 2023: Physics of Medical Imaging*, volume 12463, pages 1024–1028. SPIE, 2023.
- [15] Xiaoshan Li, Da Xu, Hua Zhou, and Lexin Li. Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences*, 10:520–545, 2018.
- [16] Libin Liu and Xiuting Xu. Marvel: Towards efficient federated learning on IoT devices. *Computer Networks*, 245:110375, 2024.
- [17] Amirhossein Malekijoo, Mohammad Javad Fadaeieslam, Hanieh Malekijoo, Morteza Homayounfar, Farshid Alizadeh-Shabdiz, and Reza Rawasizadeh. Fedzip: A compression framework for communication-efficient federated learning. *arXiv preprint arXiv:2102.01593*, 2021.
- [18] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.
- [19] Xin Nie, Laurence T Yang, Zhe Li, Xianjun Deng, Fulan Fan, and Zecan Yang. Interpretable multimodal tucker fusion model with information filtering for multimodal sentiment analysis. *IEEE Transactions on Computational Social Systems*, 2024.
- [20] Haemin Park and Diego Klabjan. Communication-efficient federated low-rank update algorithm and its connection to implicit regularization. *arXiv preprint arXiv:2409.12371*, 2024.
- [21] Jiaying Qi, Zhongzhi Luan, Shaohan Huang, Carol Fung, Hailong Yang, and Depei Qian. FDLORA: Personalized federated learning of large language model via dual LoRA tuning. *arXiv preprint arXiv:2406.07925*, 2024.
- [22] Jose Hoyos Sanchez, Batoul Taki, Waheed U Bajwa, and Anand D Sarwate. Federated learning of tensor generalized linear models with low separation rank. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2136–2140. IEEE, 2024.
- [23] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3400–3413, 2019.
- [24] Oguz Semerci, Ning Hao, Misha E Kilmer, and Eric L Miller. Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. *IEEE Transactions on Image Processing*, 23(4):1678–1693, 2014.
- [25] Rui Song, Dai Liu, Dave Zhenyu Chen, Andreas Festag, Carsten Trinitis, Martin Schulz, and Alois Knoll. Federated learning via decentralized dataset distillation in resource-constrained edge environments. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2023.
- [26] Laurent Sorber, Marc Van Barel, and Lieven De Lathauwer. Structured data fusion. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):586–600, 2015.
- [27] Nick Vannieuwenhoven, Raf Vandebril, and Karl Meerbergen. A new truncation strategy for the higher-order singular value decomposition. *SIAM Journal on Scientific Computing*, 34(2):A1027–A1052, 2012.
- [28] Qing Wu, Zhe Jiang, Kewei Hong, Huazhong Liu, Laurence T Yang, and Jihong Ding. Tensor-based recurrent neural network and multi-modal prediction with its applications in traffic network management. *IEEE Transactions on Network and Service Management*, 18(1):780–792, 2021.
- [29] Yunshu Wu, Uday Singh Saini, Jia Chen, and Evangelos E Papalexakis. Tenalign: Joint tensor alignment and coupled factorization. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 568–577. IEEE, 2022.
- [30] Ziyuan Yang, Wenjun Xia, Zexin Lu, Yingyu Chen, Xiaoxiao Li, and Yi Zhang. Hypernetwork-based physics-driven personalized federated learning for CT imaging. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [31] Liping Yi, Han Yu, Gang Wang, and Xiaoguang Liu. FedLoRA: Model-heterogeneous personalized federated learning with LoRA tuning. *arXiv preprint arXiv:2310.13283*, 2023.
- [32] Xiangtao Zhang, Eleftherios Kofidis, Ce Zhu, Le Zhang, and Yipeng Liu. Federated learning using coupled tensor train decomposition. *arXiv preprint arXiv:2403.02898*, 2024.
- [33] Haifeng Zheng, Min Gao, Zhizhang Chen, and Xinxin Feng. A distributed hierarchical deep computation model for federated learning in edge computing. *IEEE Transactions on Industrial Informatics*, 17(12):7946–7956, 2021.
- [34] Guoxu Zhou, Andrzej Cichocki, and Shengli Xie. Decomposition of big tensors with low multilinear rank. *arXiv preprint arXiv:1412.1885*, 2014.
- [35] 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.
- [36] Jincheng Zhou and Yang Lei. Multi-source heterogeneous data fusion algorithm based on federated learning. In *International Conference on Soft Computing in Data Science*, pages 46–60. Springer, 2023.

## APPENDIX A RESULTS FOR COMPAVG

We present the results of CompAVG, the naive extension of FedAVG in which factor matrices are averaged across clients. Evidently in Fig. 6, CompAVG does not yield meaningful



**Fig. 6:** Evaluating the performance of various methods given noisy data ( $\sigma = 0.1$ ). For CompJF, CompRandJF and FullDecomp, Tucker rank is  $r = 100$ .

training compared to other methods. This is likely because the factor matrices are obtained via tensor decomposition after the full-size tensor update by gradient descent, unlike the weights in FedAVG which are directly optimized.