Technometrics

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/utch20

Federated Multiple Tensor-on-Tensor Regression (FedMTOT) for Multimodal Data Under Data-Sharing Constraints

Zihan Zhang, Shancong Mou, Mostafa Reisi Gahrooei, Massimo Pacella & Jianjun Shi

To cite this article: Zihan Zhang, Shancong Mou, Mostafa Reisi Gahrooei, Massimo Pacella & Jianjun Shi (07 May 2024): Federated Multiple Tensor-on-Tensor Regression (FedMTOT) for Multimodal Data Under Data-Sharing Constraints, Technometrics, DOI: 10.1080/00401706.2024.2333506

To link to this article: https://doi.org/10.1080/00401706.2024.2333506

TECHNOMETRICS
2024, VOL. 00, NO. 0, 1–13
https://doi.org/10.1080/00401706.2024.2333506

Taylor & Francis
Taylor & Francis Group

Federated Multiple Tensor-on-Tensor Regression (FedMTOT) for Multimodal Data Under Data-Sharing Constraints

Zihan Zhang[a], Shancong Mou[a], Mostafa Reisi Gahrooei[b], Massimo Pacella[c], and Jianjun Shi[a]

[a]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA; [b]Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL; [c]Department of Innovation Engineering, University of Salento, Lecce, Italy

ABSTRACT

In recent years, diversified measurements reflect the system dynamics from a more comprehensive perspective in system modeling and analysis, such as scalars, waveform signals, images, and structured point clouds. To handle such multimodal structured high-dimensional (SHD) data, combining a large amount of data from multiple sites is necessary (i) to reduce the inherent population bias from a single site and (ii) to increase the model accuracy. However, impeded by data management policies and storage costs, data could not be easily shared or directly exchanged among different sites. Instead of simplifying or facilitating the data query process, we propose a federated multiple tensor-on-tensor regression (FedMTOT) framework to train the individual system model locally using (i) its own data and (ii) data features (not data itself) from other sites. Specifically, federated computation is executed based on alternating direction method of multipliers (ADMM) to satisfy data-sharing requirements, while the individual model at each site can still benefit from feature knowledge from other sites to improve its own model accuracy. Finally, two simulations and two case studies validate the superiority of the proposed FedMTOT framework.

ARTICLE HISTORY
Received July 2023
Accepted March 2024

KEYWORDS
Data-sharing compliance; Federated learning; Multimodal data fusion; Structured high-dimensional data

1. Introduction

Complex systems generate multimodal data in various forms, such as scalars, waveform signals, images, and video signals. Such datasets are often collected by advanced sensing technologies, such as high sampling frequency sensors and high-resolution cameras that produce structured high-dimensional (SHD) data containing abundant system information. Data collected by one type of instrument is often referred as a data mode and the full dataset is called multimodal dataset (Gaw, Yousefi, and Gahrooei 2022). For example, NO_x Storage Catalyst (NSC) is an emission control system, in which multiple sensors are installed to monitor both the combustion and the exhaust gas after the treatment process. By predicting the normalized relative fuel ratio of the NSC system from multichannel operation signals, engineers can test whether the system satisfies the environmental requirements (Gahrooei et al. 2021). As another example, electronic health records (EHR) are comprehensive repositories of diverse healthcare data sourced from various healthcare providers and medical devices. They encompass a wide range of information such as patients' diagnoses, laboratory test results, and medication usage. EHRs play a crucial role in supporting biomedical and clinical research, providing valuable data for analysis and investigation (Liu et al. 2022).

Many statistical approaches have been proposed to model such multimodal SHD data and benefited numerous applications, including manufacturing processes (Shi 2023; Zhang et al. 2023), structural health monitoring (Gordan et al. 2022), and neuroimaging data analysis (Zhou, Li, and Zhu 2013; Zhao, Reisi Gahrooei, and Gaw 2022). Particularly, SHD regression approaches are designed for developing predictive models that estimate an output given a set of inputs. For example, traditional regression methods, including penalized ordinary least square regression, have been applied to SHD data by considering each observation within an SHD data (e.g., each pixel within an image) as a covariate. However, these methods ignore the dependence among covariates. Consequently, they may result in severe overfitting and inaccurate predictions (Gahrooei et al. 2021). Principal component regression and partial least square regression methods have been used to reduce the data dimension, but they fail to fully exploit the spatial or temporal structure within the SHD data. In addition, functional regression models gained popularity in modeling waveform signals due to their capacity in capturing nonlinear correlation structure and built-in data reduction functionality. However, they require domain knowledge to create basis functions and are often very difficult and expensive to be extended to SHD data beyond waveform signals (Luo and Qi 2017; Gahrooei et al. 2021).

Recently, multi-dimensional analysis (a.k.a., tensor analysis) has been widely studied and showed promising results in many applications, such as process monitoring and modeling (Yan, Paynabar, and Shi 2015), neurological disorders (Zhou, Li, and Zhu 2013), network analysis (Orús 2019), and overlay error estimation in semiconductor industry (Zhong et al. 2023). Particularly, tensor analysis has been used in developing

CONTACT Mostafa Reisi Gahrooei mreisigahrooei@ufl.edu Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/TECH.

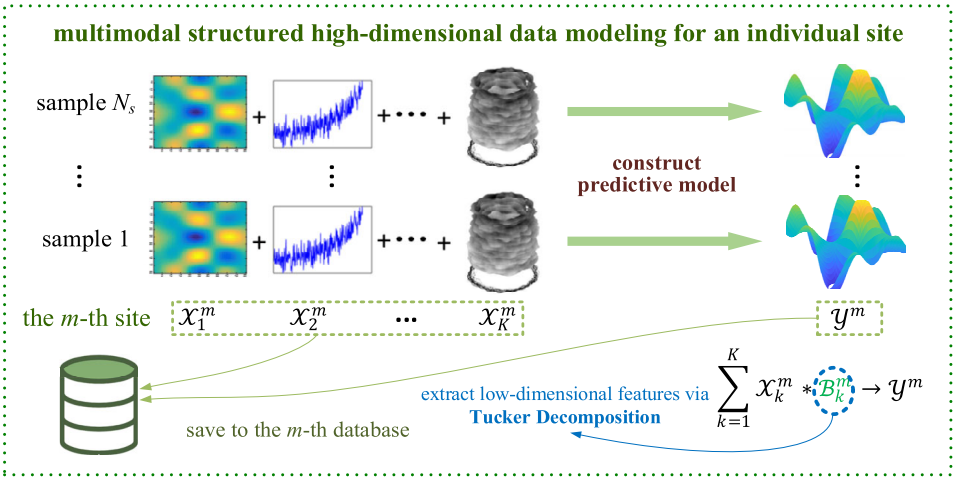

Figure 1. Each site m stores its own multimodal structured high-dimensional data and constructs a predictive model with coefficient sets $\{\mathcal{B}_k^m\}$ based on K input data sources $\{\mathcal{X}_k^m\}$ and one response $\mathcal{Y}^m$.

SHD regression modeling frameworks and involves multiple variations depending on the forms of inputs and output. For example, Zhao et al. (2012) and Fang, Paynabar, and Gebraeel (2019) estimated a scalar response given a tensor input. Yan, Paynabar, and Pacella (2019) predicted a tensor response from a set of scalar inputs. Furthermore, Lock (2018) used tensor analysis to propose a tensor-on-tensor regression that efficiently predicted a tensor output using a tensor input. However, this method only involves a single input and requires that the input and output hold the same rank, which is not appropriate in situations where multimodal input data is available. To overcome these limitations, Gahrooei et al. (2021) developed a multiple tensor-on-tensor (MTOT) regression, which provides a unified regression framework that estimates a scalar, curve, image, or structured point cloud output based on a multimodal set of SHD input variables (see Figure 1). The popularity of tensors for modeling multimodal SHD datasets relates to their capability of representing various data forms without breaking the data structure into vectors and preserving their inner correlation structure (Gahrooei et al. 2021; Lee et al. 2023).

Multimodal SHD datasets are often collected in a decentralized way. This involves various individual sites independently generating and storing similar datasets, which are then used locally to create *local models*, a process illustrated in Figure 2 (left). However, this approach of in-silo data modeling, where the modeling is done in isolation without incorporating or considering external data, limits the generalizability of the models. While one approach to address this limitation is that all sites share their datasets with a global server to create a *global model* as represented in Figure 2 (left), a few challenges make this approach unfavorable. First, data owners may not be willing to share their data due to data management concerns. Second, the demand to upload and store a vast amount of data to the global server incurs high costs. Even if the data transmission is feasible, training a model with moderately large, pooled dataset usually results in significant storage costs. To address these challenges and driven by the growing demand for scalability, resilience, and data-sharing compliance, federated data analysis (FeDA) frameworks have been proposed lately.

FeDA became a promising modeling paradigm for collaboratively extracting knowledge and conducting analysis without direct data sharing (Kontar et al. 2021). Consequently, local datasets are not required to be transferred to a global server; and the global server has no burden to store and to process immense amounts of data. In light of this novel paradigm, various techniques including FedAvg (Brendan McMahan et al. 2016), FedProx (Li et al. 2018), FedDyn (Acar et al. 2021), FedSplit (Pathak and Wainwright 2020), and FedLin (Yue, Kontar, and Gómez 2022) are developed. Specifically, Federated Averaging (FedAvg) is a practical method for federated learning based on iterative model averaging, in which a global server creates a global model by aggregating gradients of locally trained models in an iterative approach (Brendan McMahan et al. 2016). FedAvg degrades significantly when data across individual sites are heterogenous (McMahan et al. 2017). FedProx adds a quadratic regularizer term to the local objective, which enables to train the global model with heterogenous data (Li et al. 2018). Although FedProx can partially alleviate heterogeneity, it is inconsistent with local and global stationary solutions (Kontar et al. 2021). Similarly, FedDyn designed a dynamic regularization to address heterogeneity and to align gradients under partial participation (Acar et al. 2021). FedSplit applies Peaceman-Rachford splitting to formulate a constrained optimization problem (Pathak and Wainwright 2020). Recently, Yue, Kontar, and Gómez (2022) proposed a federated treatment for linear regression by adopting a hierarchical modeling approach. While these methods have demonstrated the benefits of FeDA, they are not designed for tensor data. Recently, federated tensor decomposition techniques have been proposed to handle tensor data via passing features extracted from tenor decomposition. Feng et al. (2020) developed a privacy-preserving tensor decomposition method, which leverages properties of homomorphic encryption. Wang et al. (2022) proposed a personalized federated learning framework named TDPFed, in which tensorized local model and tensorized linear (or convolutional) layers are used to reduce the communication cost. However, these methods are for unsupervised learning and are not designed for multimodal SHD data.

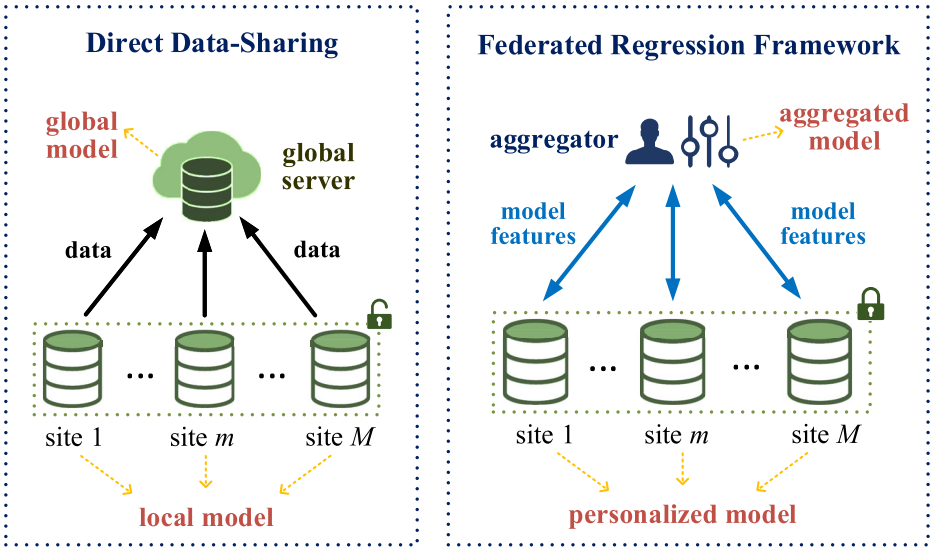

Figure 2. Overview: (left) conventional MTOT models where participating sites directly share their data to a server which creates an MTOT model based on the pooled data; (right) the proposed FedMTOT framework and associated federated models, where each site constructs a site-specific MTOT model and shares the model features with an aggregator.

The goal of this article is to model multimodal SHD data distributed across multiple sites without directly sharing data with a centralized entity by proposing a federated multiple tensor-on-tensor regression (FedMTOT) framework. As shown in Figure 2 (right), a MTOT model is established for each individual site m based on K sources of multimodal SHD inputs and the corresponding output. Here, all the data are assumed to have the low-rank structure. To reduce the modeling costs and follow the data sharing constraints, we adopt Tucker decomposition to extract latent features of model parameters (i.e., core tensor, input bases, and output bases) that are transmitted to the aggregator instead of the raw data. Under the decentralized setting, input bases will be first learned from input tensors via the alternating direction method of multipliers (ADMM). Then, given input bases, the remaining features for regression coefficients are estimated iteratively in a federated fashion. Under the proposed federated framework as shown in Figure 1(c), we can construct *personalized models* at individual sites and an *aggregated model* by the aggregator.

The rest of the article is organized as follows: Section 2 introduces the notations and tensor algebra. Section 3 first discusses the problem background and MTOT models trained by pooled raw data. To handle challenges from distributed data, we propose the federated multiple tensor-on-tensor regression framework and discuss the hyperparameter settings. In Section 4, two sets of simulations are conducted to explore the robustness and applicability of the proposed framework. The first simulation study considers a combination of two images with different sizes, while the second one considers a combination of a functional curve and an image. In each simulation study, we compare federated models, that is, aggregated model and personalized models, with nonfederated models and FedAvg in terms of standardized prediction mean square errors (SPME) for response prediction or the inverse of the signal to noise ratio (ISNR) for image denoising. Two case studies are considered in Section 5. One case study is to predict the normalized relative fuel ratio from operating signals, and the other is to test the denoising performance of the federated approach. Section 6 concludes the article.

2. Notations and Tensor Algebra

In this section, we introduce the notations and basic tensor algebra used in this article. Throughout the article, a letter denotes a scalar, for example, r and R; a boldface letter denotes a vector (e.g., $\mathbf{r}$) or a matrix (e.g., $\mathbf{R}$); a calligraphic letter denotes a tensor, for example, $\mathcal{R}$. For example, an order-n tensor is denoted by $\mathcal{R} \in \mathbb{R}^{I_1 \times \ldots \times I_n}$, where I_i is the dimension of the ith mode of tensor $\mathcal{R}$. The mode-i unfolding (matricization) of tensor $\mathcal{R}$ is $\mathbf{R}_{(i)} \in \mathbb{R}^{I_i \times I_{-i}}$, whose columns are the mode-i fibers of the corresponding tensor $\mathcal{R}$, and $I_{-i} = I_1 \times I_2 \times \ldots \times I_{i-1} \times I_{i+1} \times \ldots \times I_n$. A more general matricization of tensor $\mathcal{R} \in \mathbb{R}^{P_1 \times \ldots \times P_L \times Q_1 \times \ldots \times Q_D}$ can be defined as follows: $\mathbf{R} \in \mathbb{R}^{P \times Q}\left(P = \prod_{l=1}^{L} P_l; Q = \prod_{d=1}^{D} Q_d\right)$ with $\mathbf{R}(p, q) = \mathcal{R}_{p_1 \ldots p_L q_1 \ldots q_D}$, where $p = 1 + \sum_{j=1}^{l} \sum_{i=1}^{j} P_i (p_i - 1)$, and $q = 1 + \sum_{j=1}^{d} \sum_{i=1}^{j} Q_i (q_i - 1)$. The tensor concatenation along the first mode is denoted by $\oplus$. For example, the concatenation of tensor $\mathcal{R}_1 \in \mathbb{R}^{M \times P_1 \times \ldots \times P_L}$ and tensor $\mathcal{R}_2 \in \mathbb{R}^{N \times P_1 \times \ldots \times P_L}$ is $\mathcal{R}^{concate} \in \mathbb{R}^{(M+N) \times P_1 \times \ldots \times P_L}$, that is, $\mathcal{R}^{concate} = \mathcal{R}_1 \oplus \mathcal{R}_2$.

The Frobenius norm of a tensor $\mathcal{R}$ equals to the Frobenius norm of any unfolded format of $\mathcal{R}$, that is, $||\mathcal{R}||_F^2 = \left|\left|\mathbf{R}_{(i)}\right|\right|_F^2$ with $i = 1, \ldots, n$. The mode-i product of a tensor $\mathcal{R}_1$ by a matrix $\mathbf{R}_2 \in \mathbb{R}^{M \times I_i}$ is defined as $\mathcal{R}_1 \times_i \mathbf{R}_2 \in \mathbb{R}^{I_1 \times \ldots \times I_{i-1} \times M \times I_{i+1} \times \ldots \times I_n}$. The contraction product (Einstein product) of two tensors $\mathcal{R}_1 \in \mathbb{R}^{P_1 \times \ldots \times P_L}$ and $\mathcal{R}_2 \in \mathbb{R}^{P_1 \times \ldots \times P_L \times Q_1 \times \ldots \times Q_D}$ is denoted as $\mathcal{R}_1 * \mathcal{R}_2 \in \mathbb{R}^{Q_1 \times \ldots \times Q_D}$. The Tucker decomposition of a tensor $\mathcal{R} \in \mathbb{R}^{P_1 \times \ldots \times P_L \times Q_1 \times \ldots \times Q_D}$ decomposes the tensor into a core tensor $\mathcal{C} \in \mathbb{R}^{\tilde{P}_1 \times \ldots \times \tilde{P}_L \times \tilde{Q}_1 \times \ldots \times \tilde{Q}_D}$, a set of bases $\mathbf{U}_l \in \mathbb{R}^{P_l \times \tilde{P}_l}, (l = 1, \ldots, L)$, and $\mathbf{V}_d \in \mathbb{R}^{Q_d \times \tilde{Q}_d}, (d = 1, \ldots, D)$, that is, $\mathcal{R} = \mathcal{C} \times_1 \mathbf{U}_1 \times_2 \ldots \times_L \mathbf{U}_L \times_{L+1} \mathbf{V}_1 \times_{L+2} \ldots \times_{L+D} \mathbf{V}_D$. The matricized version of this decomposition is written as $\mathbf{R} = \left(\mathbf{U}_L \bigotimes \ldots \bigotimes \mathbf{U}_1\right) \mathbf{C} \left(\mathbf{V}_D \bigotimes \ldots \bigotimes \mathbf{V}_1\right)^T$, where $\mathbf{C} \in \mathbb{R}^{\tilde{P} \times \tilde{Q}}$ is

the unfolded core tensor $\mathcal{C}$ with $\tilde{P} = \prod_{l=1}^{L} \tilde{P}_l$ and $\tilde{Q} = \prod_{d=1}^{D} \tilde{Q}_d$, and $\mathbf{R}$ is the general matricization of $\mathcal{R}$ (Kolda and Bader 2009).

3. Federated Multiple Tensor-on-Tensor Regression Framework

We consider M sites that collaborate to construct a regression model given decentralized SHD data. We assume each site has access to K sources of SHD data as inputs to predict an output tensor; all the data have the low-rank structure; and a specific source of input data (the same data modality) follows the same distribution across different sites. We denote the kth input tensor in the mth site by $\mathcal{X}_k^m \in \mathbb{R}^{N_s^m \times P_{k,1} \times \ldots \times P_{k,L_k}}$, and the output tensor by $\mathcal{Y}^m \in \mathbb{R}^{N_s^m \times Q_1 \times \ldots \times Q_D}$, where N_s^m is the sample size, P_{k,l_k} $(l_k = 1, \ldots, L_k)$ is the dimension of the l_kth mode of the tensor $\mathcal{X}_k^m$ and Q_d $(d = 1, \ldots, D)$ is the dimension of the dth mode of $\mathcal{Y}^m$.

3.1. Background: MTOT for Global and Local Model Construction

Intuitively, each site $(m = 1, \ldots, M)$ may train a **local model** in silo based on the available data in their own database using the MTOT method proposed in Gahrooei et al. (2021) as follows:

$$\mathcal{Y}^m = \sum\nolimits_{k=1}^{K} \mathcal{X}_k^m * \mathcal{B}_k^{l,m} + \mathcal{E}^m, m = 1, \ldots, M; k = 1, \ldots, K, \quad (1)$$

where $\mathcal{B}_k^{l,m} \in \mathbb{R}^{P_{k,1} \times \ldots \times P_{k,L_k} \times Q_1 \times \ldots \times Q_D}$ is the tensor of local model regression coefficient for the mth site, and $\mathcal{E}^m \in \mathbb{R}^{N_s^m \times Q_1 \times \ldots \times Q_D}$ is the error tensor for the mth site. The model parameters can be estimated by the individual site using the estimation procedure discussed in Gahrooei et al. (2021). However, this approach results in models that may lack generalizability, particularly when N_s^m is small compared to number of model parameters.

An alternative approach to create a generalizable regression model is to pool all raw data, that is, $\{\mathcal{X}_k^m\}$ and $\{\mathcal{Y}^m\}$, from all individual sites to a global server to train a **global model** by using the method proposed in Gahrooei et al. (2021):

$$\mathcal{Y} = \sum\nolimits_{k=1}^{K} \mathcal{X}_k * \mathcal{B}_k^g + \mathcal{E}, \quad (2)$$

where $\mathcal{Y} = \mathcal{Y}^1 \oplus \ldots \oplus \mathcal{Y}^m \oplus \ldots \oplus \mathcal{Y}^M$, $\mathcal{X}_k = \mathcal{X}_k^1 \oplus \ldots \oplus \mathcal{X}_k^m \oplus \ldots \oplus \mathcal{X}_k^M$, $\mathcal{E} \in \mathbb{R}^{N_s \times Q_1 \times \ldots \times Q_D}$ with $N_s = \sum_{m=1}^{M} N_s^m$ is an error tensor whose elements are from a random process, and $\mathcal{B}_k^g \in \mathbb{R}^{P_{k,1} \times \ldots \times P_{k,L_k} \times Q_1 \times \ldots \times Q_D}$ is the tensor of global regression coefficient to be estimated. However, the individual sites may not be willing to share the raw data with a global server, which makes the estimation procedure impossible.

3.2. Federated Regression Framework

To balance the generalization and personalization as well as ensuring compliance with data-sharing constraints, we propose a federated multiple tensor-on-tensor (FedMTOT) regression framework to conduct regression analysis between a structured high-dimensional (SHD) response and a set of multimodal input variables. Specifically, an aggregator moderates the model generation process by communicating with all individual sites to receive and send regression model features. At the end of the process, each individual site establishes a *personalized model* with the regression coefficient $\mathcal{B}_k^m$ whose low-dimensional embedding is as follows:

$$\mathcal{B}_k^m = \mathcal{C}_k^m \times_1 \mathbf{U}_{k,1}^m \times_2 \ldots \times_{L_k} \mathbf{U}_{k,L_k}^m \times_{L_k+1} \mathbf{V}_1^m \times_{L_k+2} \ldots \times_{L_k+D} \mathbf{V}_D^m, \quad (3)$$

where $\mathcal{C}_k^m \in \mathbb{R}^{\tilde{P}_{k,1} \times \ldots \times \tilde{P}_{k,L_k} \times \tilde{Q}_1 \times \ldots \times \tilde{Q}_D}$ is a core tensor with $\tilde{P}_{k,l_k} \ll P_{k,l_k}$ $(l_k = 1, \ldots, L_k; k = 1, \ldots, K)$ and $\tilde{Q}_d \ll Q_d$ $(d = 1, \ldots, D)$; $\left\{\mathbf{U}_{k,l_k}^m \in \mathbb{R}^{P_{k,l_k} \times \tilde{P}_{k,l_k}}\right\}$ is a set of bases that span the kth input space; and $\left\{\mathbf{V}_d^m \in \mathbb{R}^{Q_d \times \tilde{Q}_d}\right\}$ is a set of bases that span the dth output space. Please note that $\left\{\tilde{P}_{k,l_k}\right\}$ and $\left\{\tilde{Q}_d\right\}$ are the ranks associated with this Tucker low-dimensional embeddings.

Besides, the aggregator constructs an *aggregated model* with the regression coefficient $\mathcal{B}_k$ whose low-dimensional embedding is as follows:

$$\mathcal{B}_k = \mathcal{C}_k \times_1 \mathbf{U}_{k,1} \times_2 \ldots \times_{L_k} \mathbf{U}_{k,L_k} \times_{L_k+1} \mathbf{V}_1 \times_{L_k+2} \ldots \times_{L_k+D} \mathbf{V}_D, \quad (4)$$

where $\mathcal{C}_k \in \mathbb{R}^{\tilde{P}_{k,1} \times \ldots \times \tilde{P}_{k,L_k} \times \tilde{Q}_1 \times \ldots \times \tilde{Q}_D}$, $\mathbf{U}_{k,l_k} \in \mathbb{R}^{P_{k,l_k} \times \tilde{P}_{k,l_k}}$, and $\mathbf{V}_d \in \mathbb{R}^{Q_d \times \tilde{Q}_d}$ are the aggregated model features constructed based on the communications with all individual sites.

Under the proposed framework, each individual site constructs its personalized model, that is, (3), instead of sharing the raw data, and transmits model features (i.e., site-specific core tensor $\left\{\mathcal{C}_k^m\right\}$, site-specific input bases $\left\{\mathbf{U}_{k,l_k}^m\right\}$, and site-specific output bases $\left\{\mathbf{V}_d^m\right\}$) to an aggregator. These site-specific features will then be combined by the aggregator to construct an aggregated model (4) with corresponding features (i.e., aggregated core tensor $\{\mathcal{C}_k\}$, aggregated input bases $\left\{\mathbf{U}_{k,l_k}\right\}$, aggregated output bases $\{\mathbf{V}_d\}$). The aggregated features will then be broadcast back to each individual site. Each site then uses the aggregated features to update their site-specific features. Therefore, our proposed FedMTOT constructs both *personalized models* at individual sites and an *aggregated model* by the aggregator. The personalized models benefit from the information in other sites through the aggregated model, which improves their generalizability compared to the models constructed in silo.

In general, each site can potentially estimate the core tensors and the input and output bases together using an alternative approach. However, this approach may have a high computational complexity. As an alternative approach, estimating the input bases separately first and fixing them when estimating the core tensors and the output bases reduces the computational complexity of the estimation process with adequate model accuracy (Yan, Paynabar, and Pacella 2019; Gahrooei et al. 2021). Therefore, we propose a two-step federated estimation procedure as shown in Algorithm 1. First, learning the site-specific and aggregated input bases; and second, learning the site-specific and aggregated output bases and core tensors.

Under the proposed federated framework, both Steps 2 and 3 can be conducted using consensus ADMM, which decomposes the federated model construction problem into two parts, that is, (i) the *site-specific optimization*, and (ii) the *aggregated optimization*. The solution of Steps 2 and 3 in Algorithm 1 are explained

Algorithm 1 Federated Multiple Tensor-on-Tensor Regression Algorithm.

1: **Inputs**: $\{\mathcal{Y}^m\}$ and $\left\{\mathcal{X}_k^m\right\}$ stored at individual sites only.
2: **Input Basis Learning**:
Estimate $\left\{\mathbf{U}_{k,l_k}\right\}$ and $\left\{\mathbf{U}_{k,l_k}^m\right\}$. (Algorithm 2 in Section 3.2.1)
3: **Output Basis and Core Tensor Learning**:
Given $\left\{\mathbf{U}_{k,l_k}\right\}$, estimate $\{\mathbf{V}_d\}$, $\left\{\mathbf{V}_d^m\right\}$, $\{\mathcal{C}_k\}$, and $\left\{\mathcal{C}_k^m\right\}$. (Algorithm 4 in Section 3.2.2)

in detail in Algorithm 2 of Section 3.2.1 and Algorithm 4 of Section 3.2.2, respectively. Besides, we discuss the selection of involved hyperparameters and Tucker ranks in Section 3.3 and provided the convergence analysis in Part V of supplementary materials.

3.2.1. Learning the Site-Specific and Aggregated Input Bases

This section discusses the estimation procedure of the site-specific and aggregated input bases, that is, $\left\{\mathbf{U}_{k,l_k}\right\}$ and $\left\{\mathbf{U}_{k,l_k}^m\right\}$, directly from the input data located in each site. For this purpose, the aggregator and all sites collaborate to solve the following *master optimization* problem:

$$\begin{aligned}
&\min_{\left\{\mathbf{U}_{k,i,l_k}^m\right\},\left\{\mathbf{U}_{k,l_k}\right\}} \Bigg\{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{i=1}^{N_s^m} \\
&\quad\left\|\mathcal{X}_{k,i}^m - \mathcal{D}_{k,i}^m \times_1 \mathbf{U}_{k,i,1}^m \times_2 \mathbf{U}_{k,i,2}^m \times_3 \ldots \times_{L_k} \mathbf{U}_{k,i,L_k}^m\right\|_F^2 \\
&\quad+\tfrac{\lambda_u}{2}\sum_{k=1}^{K}\sum_{l_k=1}^{L_k}\left\|\mathbf{I}_{\tilde{P}_k} - \mathbf{U}_{k,l_k}^T\mathbf{U}_{k,l_k}\right\|_F^2\Bigg\}, \\
&\text{subject to } \mathbf{U}_{k,l_k} = \mathbf{U}_{k,i,l_k}^m, \forall l_k, \forall k, \forall m, \forall i \in \left\{1,\ldots,N_s^m\right\},
\end{aligned} \tag{5}$$

where $\left\{\mathcal{D}_{k,i}^m \in \mathbb{R}^{\tilde{P}_{k,1}\times\ldots\times\tilde{P}_{k,l_k}}\right\}$ are site-specific input core tensors, $\left\{\mathbf{U}_{k,i,l_k}^m \in \mathbb{R}^{P_{k,l_k}\times\tilde{P}_{k,l_k}}\right\}$ are site-specific input bases corresponding to the ith sample $\mathcal{X}_{k,i}^m$ at the mth site, λ_u is a hyperparameter, $\mathbf{I}_{\tilde{P}_k}$ is an identity matrix of dimension $\tilde{P}_k \times \tilde{P}_k$, and $\mathcal{X}_{k,i}^m \in \mathbb{R}^{\tilde{P}_{k,1}\times\ldots\times P_{k,l_k}}$ is the ith sample of $\mathcal{X}_k^m$. Here, $\mathcal{X}_{k,i}^m$ has one mode less than $\mathcal{X}_k^m$. The first term in the objective function of (5) minimizes the overall error of inputs reconstruction by summing over all the samples and sites. The second term in the objective function of (5) aims to restrict the space of possible bases in the coefficient decomposition which can alleviate the identifiability and uniqueness issues related to the tensor decomposition (Gahrooei et al. 2021). The constraint $\mathbf{U}_{k,l_k} = \mathbf{U}_{k,i,l_k}^m$ ensures that the individual sites and the aggregator eventually achieve the same set of bases. If all the data were centralized, one could directly solve (5) by replacing $\mathbf{U}_{k,i,l_k}^m$ with $\mathbf{U}_{k,l_k}$ and performing Tucker decomposition on all the input data. Nevertheless, this is not possible under the data sharing constraint because all the data is not accessible by other entities when solving the problem. Therefore, the problem will be solved locally by each individual site and then by the aggregator in an iterative fashion until a consensus is achieved according to the constraint $\mathbf{U}_{k,l_k} = \mathbf{U}_{k,i,l_k}^m$.

Under the federated framework, each individual site minimizes $\sum_{k=1}^{K}\sum_{i=1}^{N_s^m}\left\|\mathcal{X}_{k,i}^m - \mathcal{D}_{k,i}^m \times_1 \mathbf{U}_{k,i,1}^m \times_2 \mathbf{U}_{k,i,2}^m \times_3 \ldots \times_{L_k} \mathbf{U}_{k,i,L_k}^m\right\|_F^2$ based on its own data $\left\{\mathcal{X}_{k,i}^m\right\}$ parallelly. Then, the aggregator works on its task to minimize $\sum_{k=1}^{K}\sum_{l_k=1}^{L_k}\left\|\mathbf{I}_{\tilde{P}_k} - \mathbf{U}_{k,l_k}^T\mathbf{U}_{k,l_k}\right\|_F^2$ based on the transferred features and by imposing the equality constraint $\mathbf{U}_{k,l_k} = \mathbf{U}_{k,i,l_k}^m$ to further update its aggregated input bases. Next, the aggregator broadcasts the aggregated input bases to all individual sites. Here, the equality constraint $\mathbf{U}_{k,l_k} = \mathbf{U}_{k,i,l_k}^m$ is the only bridge to communicate feature information among individual sites and the aggregator, which achieves the goal of avoiding data sharing but encouraging the collaboration.

In order to solve (5) and to achieve a closed-form solution, we first use the term $\left\|\mathbf{I}_{\tilde{P}_k} - \mathbf{U}_{k,l_k}^{A^T}\mathbf{U}_{k,l_k}^B\right\|_F^2$ with equality constraints $\mathbf{U}_{k,l_k}^B = \mathbf{U}_{k,l_k}^A$ to replace the quadratic term $\left\|\mathbf{I}_{\tilde{P}_k} - \mathbf{U}_{k,l_k}^T\mathbf{U}_{k,l_k}\right\|_F^2$. That is, we write (5) as follows:

$$\begin{aligned}
&\min_{\left\{\mathbf{U}_{k,i,l_k}^m\right\},\left\{\mathbf{U}_{k,l_k}^A\right\},\left\{\mathbf{U}_{k,l_k}^B\right\}} \Bigg\{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{i=1}^{N_s^m} \\
&\quad\left\|\mathcal{X}_{k,i}^m - \mathcal{D}_{k,i}^m \times_1 \mathbf{U}_{k,i,1}^m \times_2 \mathbf{U}_{k,i,2}^m \times_3 \ldots \times_{L_k} \mathbf{U}_{k,i,L_k}^m\right\|_F^2 \\
&\quad+\tfrac{\lambda_u}{2}\sum_{k=1}^{K}\sum_{l_k=1}^{L_k}\left\|\mathbf{I}_{\tilde{P}_k} - \mathbf{U}_{k,l_k}^{A^T}\mathbf{U}_{k,l_k}^B\right\|_F^2\Bigg\}, \\
&\text{subject to } \mathbf{U}_{k,l_k}^B = \mathbf{U}_{k,l_k}^A, \mathbf{U}_{k,l_k}^A = \mathbf{U}_{k,i,l_k}^m, \forall l_k, \forall k, \forall m, \forall i \\
&\quad\in \left\{1,\ldots,N_s^m\right\},
\end{aligned} \tag{6}$$

where $\left\{\mathbf{U}_{k,l_k}^A\right\}$ and $\left\{\mathbf{U}_{k,l_k}^B\right\}$ are duplicated aggregated input bases. Please note that the equality constraint $\mathbf{U}_{k,l_k}^B = \mathbf{U}_{k,l_k}^A$ only assists to provide the closed-form solution. Since $\mathbf{U}_{k,l_k}^A$ and $\mathbf{U}_{k,l_k}^B$ play the same role, we select $\mathbf{U}_{k,l_k}^A$ to be transferred to individual sites where the equality constraint $\mathbf{U}_{k,l_k}^A = \mathbf{U}_{k,i,l_k}^m$ allows individual sites and the aggregator to reach a consensus over several iterations and communications. To solve (6), we use an ADMM algorithm and write the augmented Lagrangian function $\mathcal{L}_U$ of (6) as follows:

$$\begin{aligned}
\mathcal{L}_U = &\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{i=1}^{N_s^m} \\
&\left\|\mathcal{X}_{k,i}^m - \mathcal{D}_{k,i}^m \times_1 \mathbf{U}_{k,i,1}^m \times_2 \mathbf{U}_{k,i,2}^m \times_3 \ldots \times_{L_k} \mathbf{U}_{k,i,L_k}^m\right\|_F^2 \\
&+\tfrac{\lambda_u}{2}\sum_{k=1}^{K}\sum_{l_k=1}^{L_k}\left\|\mathbf{I}_{\tilde{P}_k} - \mathbf{U}_{k,l_k}^{A^T}\mathbf{U}_{k,l_k}^B\right\|_F^2 \\
&+\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{i=1}^{N_s^m}\sum_{l_k=1}^{L_k}\left(\mathbf{W}_{k,i,l_k}^{m^T}\left(\mathbf{U}_{k,l_k}^A - \mathbf{U}_{k,i,l_k}^m\right)\right. \\
&\left.+\tfrac{\rho_u}{2}\left\|\mathbf{U}_{k,l_k}^A - \mathbf{U}_{k,i,l_k}^m\right\|_F^2\right) + \sum_{k=1}^{K}\sum_{l_k=1}^{L_k} \\
&\left(\mathbf{S}_{k,l_k}^T\left(\mathbf{U}_{k,l_k}^B - \mathbf{U}_{k,l_k}^A\right) + \tfrac{\mu_u}{2}\left\|\mathbf{U}_{k,l_k}^B - \mathbf{U}_{k,l_k}^A\right\|_F^2\right),
\end{aligned} \tag{7}$$

where $\mathbf{W}_{k,i,l_k}^m$ and $\mathbf{S}_{k,l_k}$ are the site-specific and aggregated Lagrangian multipliers, respectively. Although $\mathbf{W}_{k,i,l_k}^m$ and $\mathbf{S}_{k,l_k}$ are all Lagrangian multipliers, they play different roles in the optimization. Specifically, $\mathbf{W}_{k,i,l_k}^m$ assists $\mathbf{U}_{k,i,l_k}^m$ to integrate the feature information from $\mathbf{U}_{k,l_k}^A$; while $\mathbf{S}_{k,l_k}$ helps the aggregator to handle the equality constraint $\mathbf{U}_{k,l_k}^B = \mathbf{U}_{k,l_k}^A$. The penalty terms that are multiplied by parameter ρ_u and μ_u help $\mathcal{L}_U$ to enhance the convergence property within the federated framework.

In the following sections, we will discuss how to distribute the problem of minimizing (7) to individual sites and the aggregator, and how to solve this problem.

3.2.1.1 Site-Specific Optimization. Under the proposed federated framework, each individual site m updates site-specific input core tensors $\left\{\mathcal{D}_{k,i}^m\right\}$ and the input bases $\left\{\mathbf{U}_{k,i,l_k}^m\right\}$ by solving the following subproblem (the objective function is a subpart of (7)), assuming that the aggregated feature $\mathbf{U}_{k,l_k}^A$ is known (i.e., provided by the aggregator):

$$\begin{aligned}\min_{\left\{\mathcal{D}_{k,i}^m\right\},\left\{\mathbf{U}_{k,i,l_k}^m\right\}} & \left\{\left|\left|\mathcal{X}_{k,i}^m-\mathcal{D}_{k,i}^m\times_1\mathbf{U}_{k,i,1}^m\times_2\mathbf{U}_{k,i,2}^m\times_3\ldots\times_{L_k}\mathbf{U}_{k,i,L_k}^m\right|\right|_F^2\right. \\ & \left.+\mathbf{W}_{k,i,l_k}^{m^T}\left(\mathbf{U}_{k,l_k}^A-\mathbf{U}_{k,i,l_k}^m\right)+\frac{\rho_u}{2}\left|\left|\mathbf{U}_{k,l_k}^A-\mathbf{U}_{k,i,l_k}^m\right|\right|_F^2\right\},\end{aligned} \tag{8}$$

by using the alternative least square approach. Notably, although $\left\{\mathcal{D}_{k,i}^m\right\}$ are estimated when performing the Tucker decomposition on $\left\{\mathcal{X}_{k,i}^m\right\}$, they are not used in estimating the model parameters $\left\{\mathcal{B}_k^m\right\}$. More specifically, the site-specific input core tensors $\left\{\mathcal{D}_{k,i}^m\right\}$ is first estimated given the input bases $\left\{\mathbf{U}_{k,l_k}^m\right\}$ as follows:

$$\begin{aligned}\mathcal{D}_{k,i}^m=\mathcal{X}_{k,i}^m\times_1\left(\mathbf{U}_{k,i,1}^{m^T}\mathbf{U}_{k,i,1}^m\right)^{-1}\mathbf{U}_{k,i,1}^{m^T} \\ \times_2\ldots\times_{L_k}\left(\mathbf{U}_{k,i,L_k}^{m^T}\mathbf{U}_{k,i,L_k}^m\right)^{-1}\mathbf{U}_{k,i,L_k}^{m^T},\forall k,\forall i.\end{aligned} \tag{9}$$

Here, $\left\{\mathbf{U}_{k,i,l_k}^m\right\}$ are orthonormal and nonsingular. When $\left\{\mathbf{U}_{k,i,l_k}^m\right\}$ become orthonormal, (9) is equivalent to $\mathcal{D}_{k,i}^m=\mathcal{X}_{k,i}^m\times_1\mathbf{U}_{k,i,1}^{m^T}\times_2\mathbf{U}_{k,i,2}^{m^T}\times_3\ldots\times_{L_k}\mathbf{U}_{k,i,L_k}^{m^T}$.

Next, given the raw data $\mathcal{X}_{k,i}^m$, the site-specific input core tensor $\mathcal{D}_{k,i}^m$, the site-specific Lagrangian multipler $\mathbf{W}_{k,i,l_k}^m$, and other site-specific input bases $\left\{\mathbf{U}_{k,i,l_k'}^m\right\}\left(l_k'\neq l_k\right)$, the individual site can obtain a closed-form solution for $\mathbf{U}_{k,i,l_k}^m$ as follows:

$$\begin{aligned}\mathbf{U}_{k,i,l_k}^m=\left(\rho_u\mathbf{U}_{k,l_k}^A+\mathbf{W}_{k,i,l_k}^m+2\mathbf{X}_{k,i(l_k)}^m\mathbf{R}_{k,i}^{m^T}\right) \\ \left(2\mathbf{R}_{k,i}^m\mathbf{R}_{k,i}^{m^T}+\rho_u\mathbf{I}_{\tilde{P}_k}\right)^{-1},\end{aligned} \tag{10}$$

where $\mathbf{D}_{k,i(l_k)}^m$ is the mode-l_k matricization of $\mathcal{D}_{k,i}^m$, $\mathbf{R}_{k,i}^m=\mathbf{D}_{k,i(l_k)}^m\left(\mathbf{U}_{k,i,l_k^+}^m\otimes\mathbf{U}_{k,i,l_k^-}^m\right)^T$, $\mathbf{U}_{k,i,l_k^+}^m=\mathbf{U}_{k,i,L_k}^m\otimes\ldots\otimes\mathbf{U}_{k,i,(l_k+1)}^m$, and $\mathbf{U}_{k,i,l_k^-}^m=\mathbf{U}_{k,i,(l_k-1)}^m\otimes\ldots\otimes\mathbf{U}_{k,i,1}^m$. If the input is a functional curve, that is, $L_k=1$, we have $\mathbf{U}_{k,i,1}^m=\left(2\mathbf{X}_{k,i}^{m^T}\mathbf{D}_{k,i}^m+\mathbf{W}_{k,i,1}^m+\rho_u\mathbf{U}_{k,1}^A\right)\left(2\mathbf{D}_{k,i}^{m^T}\mathbf{D}_{k,i}^m+\rho_u\mathbf{I}_{\tilde{P}_k}\right)^{-1}$. The details of the derivation can be found in Part II of supplementary materials.

Once individual sites solve (8), they send their updated site-specific features together with the site-specific Lagrangian multipliers (which have not been updated) to the aggregator. After the aggregator solves the aggregated optimization, individual site m receives the updated aggregated features and then updates their site-specific Lagrangian multipliers to adjust the gap between site-specific and aggregated input bases:

$$\mathbf{W}_{k,i,l_k}^m\leftarrow\mathbf{W}_{k,i,l_k}^m+\rho_u\left(\mathbf{U}_{k,l_k}^A-\mathbf{U}_{k,i,l_k}^m\right),\forall k,\forall l_k,\forall i,\forall m. \tag{11}$$

Please note that the site-specific Lagrangian multipliers are not updated immediately after solving (8) but updated after individual sites receive the updated aggregated features from the aggregator. It is because such a design not only follows the ADMM convention, but also can save computation resources by only updating Lagrangian multipliers once after individual sites and the aggregator has completed their own tasks at one run.

3.2.1.2 Aggregated Optimization. By pooling all site-specific input bases $\left\{\mathbf{U}_{k,l_k}^m\right\}$ and Lagrangian multipliers $\left\{\mathbf{W}_{k,i,l_k}^m\right\}$ from all individual sites, the aggregator estimates the aggregated input bases $\left\{\mathbf{U}_{k,l_k}^A\right\}$, $\left\{\mathbf{U}_{k,l_k}^B\right\}$ and the Lagrangian multipliers $\left\{\mathbf{S}_{k,l_k}^T\right\}$ by minimizing (7), in an iterative manner. First, assuming $\mathbf{U}_{k,l_k}^B$ and $\mathbf{S}_{k,l_k}^T$ are given, $\mathbf{U}_{k,l_k}^A$ is estimated by solving the following subproblem:

$$\begin{aligned}\min_{\mathbf{U}_{k,l_k}^A} & \left\{\frac{\lambda_u}{2}\left|\left|\mathbf{I}_{\tilde{P}_k}-\mathbf{U}_{k,l_k}^{A^T}\mathbf{U}_{k,l_k}^B\right|\right|_F^2\right. \\ & +\sum_{m=1}^M\sum_{i=1}^{N_s^m}\left(\mathbf{W}_{k,i,l_k}^{m^T}\left(\mathbf{U}_{k,l_k}^A-\mathbf{U}_{k,i,l_k}^m\right)+\frac{\rho_u}{2}\left|\left|\mathbf{U}_{k,l_k}^A-\mathbf{U}_{k,i,l_k}^m\right|\right|_F^2\right) \\ & \left.+\mathbf{S}_{k,l_k}^T\left(\mathbf{U}_{k,l_k}^B-\mathbf{U}_{k,l_k}^A\right)+\frac{\mu_u}{2}\left|\left|\mathbf{U}_{k,l_k}^B-\mathbf{U}_{k,l_k}^A\right|\right|_F^2\right\},\end{aligned} \tag{12}$$

which has a closed-form solution as follows:

$$\begin{aligned}\mathbf{U}_{k,l_k}^A=\left(\lambda_u\mathbf{U}_{k,l_k}^B\mathbf{U}_{k,l_k}^{B^T}+\left(MN_s^m\rho_u+\mu_u\right)\mathbf{I}_{\tilde{P}_k}\right)^{-1} \\ \left(\left(\lambda_u+\mu_u\right)\mathbf{U}_{k,l_k}^B+\mathbf{S}_{k,l_k}+\sum_{m=1}^M\sum_{i=1}^{N_s^m}\right. \\ \left.\left(\rho_u\mathbf{U}_{k,i,l_k}^m-\mathbf{W}_{k,i,l_k}^m\right)\right).\end{aligned} \tag{13}$$

Next, assuming $\mathbf{U}_{k,l_k}^A$ and $\mathbf{S}_{k,l_k}^T$ are given, $\mathbf{U}_{k,l_k}^B$ is estimated by solving the following minimization problem:

$$\begin{aligned}\min_{\mathbf{U}_{k,l_k}^B} & \left\{\frac{\lambda_u}{2}\left|\left|\mathbf{I}_{\tilde{P}_k}-\mathbf{U}_{k,l_k}^{A^T}\mathbf{U}_{k,l_k}^B\right|\right|_F^2+\mathbf{S}_{k,l_k}^T\left(\mathbf{U}_{k,l_k}^B-\mathbf{U}_{k,l_k}^A\right)\right. \\ & \left.+\frac{\mu_u}{2}\left|\left|\mathbf{U}_{k,l_k}^B-\mathbf{U}_{k,l_k}^A\right|\right|_F^2\right\},\end{aligned} \tag{14}$$

which results in the following closed-form solution:

$$\mathbf{U}_{k,l_k}^B=\left(\lambda_u\mathbf{U}_{k,l_k}^A\mathbf{U}_{k,l_k}^{A^T}+\mu_u\mathbf{I}_{\tilde{P}_k}\right)^{-1}\left(\left(\lambda_u+\mu_u\right)\mathbf{U}_{k,l_k}^A-\mathbf{S}_{k,l_k}\right). \tag{15}$$

Finally, the aggregated Lagrangian multipliers are updated by the aggregator to adjust the gap between the duplicated aggregated input bases as follows:

$$\mathbf{S}_{k,l_k}\leftarrow\mathbf{S}_{k,l_k}+\mu_u\left(\mathbf{U}_{k,l_k}^B-\mathbf{U}_{k,l_k}^A\right),\forall k,\forall l_k. \tag{16}$$

To summarize, the entire algorithm for updating input bases is shown in Algorithm 2 and Figure 3 of Part I in supplementary materials. The aggregator and individual sites repeat this procedure until the minimization problem of (7) converges. During the procedure, the aggregated input bases and the site-specific ones reach a consensus without directly accessing the raw data. The degree to which the aggregated and site-specific bases match, depends on the stopping criteria used in the algorithm that is explained next.

The stopping criteria for the convergence include whether the iteration number reaches the predefined maximal value,

Algorithm 2 Input Basis Learning.

Inputs: $\left\{\mathcal{X}_{k,i}^m\right\}$.
Initialize $\mathbf{U}_{k,i,l_k}^m$, $\mathbf{W}_{k,i,l_k}^m$ using Tucker decomposition of $\left\{\mathcal{X}_{k,i}^m\right\}, \forall k, \forall l_k, \forall i, \forall m$.
Initialize $\mathbf{U}_{k,l_k}^B = \mathbf{U}_{k,l_k}^A = \mathbf{S}_{k,l_k} = \mathbf{U}_{k,1,l_k}^1, \forall k, \forall l$.
Loop
  Update $\left\{\mathcal{D}_{k,i}^m\right\}$ using (9), $\forall i, \forall k, \forall m$.
  For $k \in \{1, \ldots, K\}, l_k \in \{1, \ldots, L_k\}$
    Update $\mathbf{U}_{k,i,l_k}^m$ using (10), $\forall i, \forall m$.
    Update $\mathbf{U}_{k,l_k}^A$ using (13) and update $\mathbf{U}_{k,l_k}^B$ using (15).
    Update $\mathbf{S}_{k,l_k}$ using (16).
    Update $\mathbf{W}_{k,i,l_k}^m$ using (11), $\forall i, \forall m$.
  End for
End Until Convergence

$r_{Am}^u \le \epsilon_r$, $r_{BA}^u \le \epsilon_r$, $s_{Am}^u \le \epsilon_s$, $s_{BA}^u \le \epsilon_s$, and $\sum_{m=1}^M \sum_{k=1}^K \sum_{i=1}^{N_s^m} \left|\left|\mathcal{X}_{k,i}^m - \mathcal{D}_{k,i}^m \times_1 \mathbf{U}_{k,i,1}^m \times_2 \mathbf{U}_{k,i,2}^m \times_3 \ldots \times_{L_k} \mathbf{U}_{k,i,L_k}^m\right|\right|_F^2 \le \epsilon_{\mathcal{X}}$, where $r_{Am}^u = \sum_{m=1}^M \sum_{k=1}^K \sum_{i=1}^{N_s^m} \sum_{l_k=1}^{L_k} \left|\left|\mathbf{U}_{k,l_k}^A - \mathbf{U}_{k,i,l_k}^m\right|\right|_F^2$ and $r_{BA}^u = \sum_{k=1}^K \sum_{l_k=1}^{L_k} \left|\left|\mathbf{U}_{k,l_k}^B - \mathbf{U}_{k,l_k}^A\right|\right|_F^2$ evaluate the satisfaction of the equality constraints $\mathbf{U}_{k,l_k}^B = \mathbf{U}_{k,l_k}^A, \mathbf{U}_{k,l_k}^A = \mathbf{U}_{k,i,l_k}^m$ in (6); $s_{Am}^u = \sum_{m=1}^M \sum_{k=1}^K \sum_{i=1}^{N_s^m} \sum_{l_k=1}^{L_k} \left|\left|\mathbf{U}_{k,i,l_k}^{m(t+1)} - \mathbf{U}_{k,i,l_k}^{m(t)}\right|\right|_F^2$ and $s_{BA}^u = \sum_{k=1}^K \sum_{l_k=1}^{L_k} \left|\left|\mathbf{U}_{k,l_k}^{A(t+1)} - \mathbf{U}_{k,l_k}^{A(t)}\right|\right|_F^2$ with t representing the tth iteration monitor the algorithm convergence; the last criterion evaluates data fitness; and $\epsilon_{\mathcal{X}}, \epsilon_r, \epsilon_s$ are predefined thresholds depending on the availability of computation resources and the accuracy requirement for data fitting.

3.2.2. Federated Core Tensor and Output Basis Learning

This section discusses the estimation of the core tensors and the output tensors assuming that the site-specific and the aggregated input bases are known or estimated through the procedure discussed in Section 4.1. Given $\left\{\mathbf{U}_{k,l_k}\right\}$ obtained from Algorithm 2, the aggregator coordinates with all individual sites to update core tensors and output bases by solving the following *master optimization* problem:

$$\begin{aligned}
&\min_{\{\mathbf{V}_d^m\},\{\mathcal{C}_k^m\},\{\mathbf{V}_d\},\{\mathcal{C}_k\}} \left\{\sum_{m=1}^M \left|\left|\mathcal{Y}^m - \sum_{k=1}^K \mathcal{X}_k^m * \mathcal{B}_k^m\right|\right|_F^2 \right.\\
&+\frac{\lambda_v}{2}\sum_{d=1}^D \left|\left|\mathbf{I}_{\tilde{Q}} - \mathbf{V}_d{}^T\mathbf{V}_d\right|\right|_F^2 + \frac{\mu_v}{2}\sum_{m=1}^M\sum_{d=1}^D \left|\left|\mathbf{V}_d - \mathbf{V}_d^m\right|\right|_F^2\\
&\left.+\sum_{m=1}^M\sum_{k=1}^K \frac{\gamma_c}{2}\left|\left|\mathbf{C}_k^m - \mathbf{C}_k\right|\right|_F^2\right\},\\
&\text{subject to } \mathcal{B}_k^m = \mathcal{C}_k^m \times_1 \mathbf{U}_{k,1} \times_2 \ldots \times_{L_k} \mathbf{U}_{k,L_k} \times_{L_k+1} \mathbf{V}_1^m\\
&\quad \times_{L_k+2} \ldots \times_{L_k+D} \mathbf{V}_D^m,
\end{aligned} \tag{17}$$

where $\lambda_v, \mu_v, \gamma_c$ are hyperparameters, and $\mathbf{I}_{\tilde{Q}}$ is an identity matrix of dimension $\tilde{Q} \times \tilde{Q}$. Please note that unlike Section 4.1, we need to share the site-specific core tensors $\left\{\mathcal{C}_k^m\right\}$ with the aggregator to enhance the collaboration and eventually the generalizability of the models.

By employing the federated ADMM framework similar to the discussion in the previous section, individual site m handles $\left|\left|\mathcal{Y}^m - \sum_{k=1}^K \mathcal{X}_k^m * \mathcal{B}_k^m\right|\right|_F^2$ and the aggregator handles the orthogonality term $\sum_{d=1}^D \left|\left|\mathbf{I}_{\tilde{Q}} - \mathbf{V}_d{}^T\mathbf{V}_d\right|\right|_F^2$. To reserve the modeling flexibility, we allow the deviation of site-specific output bases and core tensors from the aggregated ones. That is, we consider proximity penalties $\left|\left|\mathbf{V}_d - \mathbf{V}_d^m\right|\right|_F^2$ and $\left|\left|\mathbf{C}_k^m - \mathbf{C}_k\right|\right|_F^2$ instead of equality constraints. This will allow the models at each individual site to be more flexible and deviate from the aggregated model.

When solving (17), we first introduce duplicated aggregated output bases $\left\{\mathbf{V}_d^A\right\}$ and $\left\{\mathbf{V}_d^B\right\}$ and rewrite (17) as follows:

$$\begin{aligned}
&\min_{\{\mathbf{V}_d^m\},\{\mathcal{C}_k^m\},\{\mathbf{V}_d^A\},\{\mathbf{V}_d^B\},\{\mathcal{C}_k\}} \left\{\sum_{m=1}^M \left|\left|\mathcal{Y}^m - \sum_{k=1}^K \mathcal{X}_k^m * \mathcal{B}_k^m\right|\right|_F^2 \right.\\
&+\frac{\mu_v}{2}\sum_{m=1}^M\sum_{d=1}^D \left|\left|\mathbf{V}_d^A - \mathbf{V}_d^m\right|\right|_F^2 + \frac{\lambda_v}{2}\sum_{d=1}^D \left|\left|\mathbf{I}_{\tilde{Q}} - \mathbf{V}_d^{A^T}\mathbf{V}_d^B\right|\right|_F^2\\
&\left.+\sum_{m=1}^M\sum_{k=1}^K \frac{\gamma_c}{2}\left|\left|\mathbf{C}_k^m - \mathbf{C}_k\right|\right|_F^2\right\},\\
&\text{subject to} \mathcal{B}_k^m = \mathcal{C}_k^m \times_1 \mathbf{U}_{k,1} \times_2 \ldots \times_{L_k} \mathbf{U}_{k,L_k} \times_{L_k+1} \mathbf{V}_1^m\\
&\quad \times_{L_k+2} \ldots \times_{L_k+D} \mathbf{V}_D^m, \mathbf{V}_d^B = \mathbf{V}_d^A, \forall d,
\end{aligned} \tag{18}$$

where λ_v is a hyperparameter. Accordingly, the augmented Lagrangian function $\mathcal{L}_V$ can be written as

$$\begin{aligned}
\mathcal{L}_V = &\sum_{m=1}^M \left|\left|\mathcal{Y}^m - \sum_{k=1}^K \mathcal{X}_k^m * \mathcal{B}_k^m\right|\right|_F^2\\
&+\frac{\mu_v}{2}\sum_{m=1}^M\sum_{d=1}^D \left|\left|\mathbf{V}_d^A - \mathbf{V}_d^m\right|\right|_F^2\\
&+\frac{\lambda_v}{2}\sum_{d=1}^D \left|\left|\mathbf{I}_{\tilde{Q}} - \mathbf{V}_d^{A^T}\mathbf{V}_d^B\right|\right|_F^2\\
&+\sum_{d=1}^D \left(\mathbf{H}_d^T\left(\mathbf{V}_d^A - \mathbf{V}_d^B\right) + \frac{\rho_v}{2}\left|\left|\mathbf{V}_d^A - \mathbf{V}_d^B\right|\right|_F^2\right)\\
&+\sum_{m=1}^M\sum_{k=1}^K \frac{\gamma_c}{2}\left|\left|\mathbf{C}_k^m - \mathbf{C}_k\right|\right|_F^2,
\end{aligned} \tag{19}$$

where $\mathbf{H}_d$ is the aggregated Lagrangian multipliers, and ρ_v is a hyperparameter. In the following sections, we distribute the optimization of (19) into individual sites and the aggregator and further discuss its solutions.

3.2.2.1 Site-Specific Optimization. Assuming that the aggregator provides $\mathbf{V}_d^A$ and $\mathbf{V}_d^B$, each individual site estimates the site-specific core tensors and output bases by minimizing (19). Specifically, by following the ADMM framework, the output bases $\mathbf{V}_d^m$ are estimated by solving the following subproblem:

$$\min_{\mathbf{V}_d^m} \left\{\left|\left|\mathcal{Y}^m - \sum_{k=1}^K \mathcal{X}_k^m * \mathcal{B}_k^m\right|\right|_F^2 + \frac{\mu_v}{2}\left|\left|\mathbf{V}_d^A - \mathbf{V}_d^m\right|\right|_F^2\right\}. \tag{20}$$

Given aggregated input bases $\left\{\mathbf{U}_{k,l_k}\right\}$, the raw data $\mathcal{Y}^m$, $\left\{\mathcal{X}_k^m\right\}$, site-specific core tensors $\left\{\mathcal{C}_k^m\right\}$, and remaining site-specific output bases $\left\{\mathbf{V}_{d'}^m\right\}$ $(d' \ne d)$, site m can set the gradient of (20) to be zero and update $\mathbf{V}_d^m$ as follows:

$$\mathbf{V}_d^m = \left(\mu_v\mathbf{V}_d^A + 2\mathbf{Y}_{(d+1)}^m\mathbf{A}^{mT}\right)\left(2\mathbf{A}^m\mathbf{A}^{mT} + \mu_v\mathbf{I}_{\tilde{Q}}\right)^{-1}. \tag{21}$$

where $\mathbf{A}^m = \sum_{k=1}^K \mathbf{A}_k^m$, $\mathbf{A}_k^m = \mathbf{C}_{k(L_k+d)}^m\left(\mathbf{V}_{d^+}^m \bigotimes \mathbf{V}_{d^-}^m \bigotimes \mathbf{Z}_k^m\right)^T$, $\mathbf{V}_{d^+}^m = \mathbf{V}_D^m \bigotimes \ldots \bigotimes \mathbf{V}_{d+1}^m$, $\mathbf{V}_{d^-}^m = \mathbf{V}_{d-1}^m \bigotimes \ldots \bigotimes \mathbf{V}_1^m$, and $\mathbf{Z}_k^m = \mathbf{X}_{k(1)}^m\left(\mathbf{U}_{k,L_k} \bigotimes \ldots \bigotimes \mathbf{U}_{k,1}\right)$. The derivation details are summarized in Part III of supplementary materials.

By rewriting (20) regarding $\mathcal{C}_k^m$, we can get the following subproblem for updating $\mathcal{C}_k^m$:

$$\begin{aligned}\underset{\text{vec}(\mathcal{C}_k^m)}{\text{argmin}} \Big\{ \Big|\Big| \text{vec}\left(\mathbf{Y}_{(1)}^m\right) - \textstyle\sum_{r=1,r\neq k}^{K} \left(\mathbf{V}_D^m \bigotimes \ldots \bigotimes \mathbf{V}_1^m \bigotimes \mathbf{Z}_r^m\right) \\ \text{vec}\left(\mathcal{C}_r^m\right) - \left(\mathbf{V}_D^m \bigotimes \ldots \bigotimes \mathbf{V}_1^m \bigotimes \mathbf{Z}_k^m\right) \text{vec}\left(\mathcal{C}_k^m\right) \Big|\Big|_F^2 \\ + \tfrac{\gamma_v}{2} \Big|\Big| \mathbf{C}_k - \mathbf{C}_k^m \Big|\Big|_F^2 \Big\}. \end{aligned} \tag{22}$$

By setting the gradient of (22) to be zero, we update $\mathcal{C}_k^m$ as follows:

$$\begin{aligned}\text{vec}\left(\mathcal{C}_k^m\right) = \Big(2 \left(\mathbf{V}_D^m \bigotimes \ldots \bigotimes \mathbf{V}_1^m \bigotimes \mathbf{Z}_k^m\right)^T \\ \left(\mathbf{V}_D^m \bigotimes \ldots \bigotimes \mathbf{V}_1^m \bigotimes \mathbf{Z}_k^m\right) + \gamma_v \mathbf{I}_{\tilde{Q}} \Big)^{-1} \left(\gamma_v \text{vec}\left(\mathcal{C}_k\right)\right. \\ + 2 \left(\mathbf{V}_D^m \bigotimes \ldots \bigotimes \mathbf{V}_1^m \bigotimes \mathbf{Z}_k^m\right)^T \Big(\text{vec}\left(\mathbf{Y}_{(1)}^m\right) \\ \left. - \textstyle\sum_{r=1,r\neq k}^{K} \left(\mathbf{V}_D^m \bigotimes \ldots \bigotimes \mathbf{V}_1^m \bigotimes \mathbf{Z}_r^m\right) \text{vec}\left(\mathcal{C}_r^m\right) \Big) \Big). \end{aligned} \tag{23}$$

After individual sites update their site-specific core tensor and output bases, they will send their site features to the aggregator.

3.2.2.2 Aggregated Optimization. By initializing aggregated output bases $\{\mathbf{V}_d^A\}$, $\{\mathbf{V}_d^B\}$, and their aggregated Lagrangian multipliers $\{\mathbf{H}_d\}$ using personalized output bases, the aggregator adopts ADMM to handle the equality constraint $\mathbf{V}_d^B = \mathbf{V}_d^A$ to update $\mathbf{V}_d^A$, $\mathbf{V}_d^B$, and $\mathbf{H}_d$ as summarized in Algorithm 3. Specifically, $\mathbf{V}_d^A$ ($\mathbf{V}_d^B$) can be updated via (19) by assuming that other variables are given. The derivation details can be found in Part IV of supplementary materials.

Algorithm 3 Update Aggregated Output Bases.

Initialize $\mathbf{V}_d^B = \mathbf{V}_d^A = \mathbf{H}_d = \mathbf{V}_d^1, \forall d$.
Loop
  $\mathbf{V}_d^A = \left(\lambda_v \mathbf{V}_d^B \mathbf{V}_d^{B^T} + (M\mu_v + \rho_v)\, \mathbf{I}_{\tilde{Q}}\right)^{-1} \left((\lambda_v + \rho_v)\, \mathbf{V}_d^B + \mu_v \textstyle\sum_{m=1}^{M} \mathbf{V}_d^m - \mathbf{H}_d\right)$.
  $\mathbf{V}_d^B = \left(\lambda_v \mathbf{V}_d^A \mathbf{V}_d^{A^T} + \rho_v \mathbf{I}_{\tilde{Q}}\right)^{-1} \left((\lambda_v + \rho_v)\, \mathbf{V}_d^A + \mathbf{H}_d\right)$.
  $\mathbf{H}_d \leftarrow \mathbf{H}_d + \rho_v \left(\mathbf{V}_d^A - \mathbf{V}_d^B\right)$.
End Until Convergence

Apart from updating aggregated output bases, the aggregator pools all site-specific core tensors $\{\mathcal{C}_k^m\}$, the aggregated core tensor $\{\mathcal{C}_k\}$ and then update the aggregated core tensor $\mathcal{C}_k$ by assuming that other terms are given from (19):

$$\min_{\mathbf{C}_k} \left\{ \sum_{m=1}^{M} \frac{\gamma_c}{2} \Big|\Big| \mathbf{C}_k^m - \mathbf{C}_k \Big|\Big|_F^2 \right\} \tag{24}$$

By setting the gradient of (24) to be zero, we get

$$\mathbf{C}_k = \frac{1}{M} \sum_{m=1}^{M} \mathbf{C}_k^m. \tag{25}$$

The entire algorithm for output basis and core tensor learning is summarized in Algorithm 4 and Figure 1.

The stopping criteria for this algorithm include whether the iteration number reaches the predefined maximal value, $r_{Am}^v \leq \epsilon_r$, $r_{BA}^v \leq \epsilon_r$, $s_{Am}^v \leq \epsilon_s$, $s_{BA}^v \leq \epsilon_s$, and $\sum_{m=1}^{M} \Big|\Big| \mathcal{Y}^m - \sum_{k=1}^{K} \mathcal{X}_k^m * \mathcal{B}_k^m \Big|\Big|_F^2 \leq \epsilon_{\mathcal{Y}}$, where $r_{Am}^v = \sum_{m=1}^{M} \sum_{d=1}^{D} \Big|\Big| \mathbf{V}_d^A - \mathbf{V}_d^m \Big|\Big|_F^2$ controls the deviation of site-specific features from the aggregate features; $r_{BA}^v = \sum_{d=1}^{D} \Big|\Big| \mathbf{V}_d^B - \mathbf{V}_d^A \Big|\Big|_F^2$ evaluates the satisfaction level of the equality constraint $\mathbf{V}_d^B = \mathbf{V}_d^A$ in (18); $s_{Am}^v = \sum_{m=1}^{M} \sum_{d=1}^{D} \Big|\Big| \mathbf{V}_d^{m^{(t+1)}} - \mathbf{V}_d^{m^{(t)}} \Big|\Big|_F^2$ and $s_{BA}^v = \sum_{d=1}^{D} \Big|\Big| \mathbf{V}_d^{A^{(t+1)}} - \mathbf{V}_d^{A^{(t)}} \Big|\Big|_F^2$ with t representing the tth iteration help to monitor the algorithm convergence; the last criterion evaluates the model fitness; and $\epsilon_{\mathcal{Y}}, \epsilon_r, \epsilon_s$ are predefined thresholds determined by computation capability and fitness requirement.

Algorithm 4 Update Core Tensors and Output Bases.

Inputs: $\{\mathcal{Y}^m\}$, $\left\{\mathcal{X}_k^m\right\}$, and $\left\{\mathbf{U}_{k,l_k}\right\}$.
Initialize $\mathbf{V}_d^m$ and $\mathbf{H}_d^m$ using Tucker decomposition of $\mathcal{Y}^m$, $\forall m$.
Initialize $\mathbf{V}_d^B = \mathbf{V}_d^A = \mathbf{H}_d = \mathbf{V}_d^1, \forall d$.
Loop
  For $k \in \{1, \ldots, K\}$
    Loop
      For $m \in \{1, \ldots, M\}$
        For $d \in \{1, \ldots, D\}$
          Update $\mathbf{V}_d^m$ using (21).
          Algorithm 3.
        End for
        Update $\mathcal{C}_k^m$ using (23).
      End For
      Update $\mathcal{C}_k$ using (25).
    End Until Convergence
  End for
End Until Convergence

3.3. Determining Hyperparameters and Tucker Ranks

In the proposed federated framework, tuning a set of hyperparameters (i.e., λ_u, λ_v, ρ_u, ρ_v, and μ_u) are essential. The hyperparameters λ_u and λ_v are tied to the orthonormality constraints, while ρ_u, ρ_v, and μ_u are associated with the Lagrangian multipliers. We conduct empirical experiments for selecting these values to ensure feasible solutions. Alternatively, we can initialize these values and incrementally adjust them across algorithm iterations, enhancing the solution's feasibility as suggested by Lee et al. (2023). Additionally, the hyperparameters μ_v and γ_c regulate proximity penalties, maintaining the local models' alignment with the aggregated model while allowing for necessary flexibility. High values for these hyperparameters promote uniformity in estimating site-specific models, which is beneficial when sites are homogeneous. Conversely, smaller values afford more flexibility, enabling deviation when individual sites have heterogeneous models. Thus, these hyperparameters should be chosen based on domain expertise and empirical experimentation tailored to the specific application (Konyar and Reisi Gahrooei 2023).

Next, we determine how to select the Tucker ranks under each hyperparameter setting. Specifically, each individual site

will first apply singular value decomposition (SVD) to the matricizated raw data $\mathbf{Y}^m_{(d+1)}$ and $\left\{\mathbf{X}^m_{k,i,(l_k)}\right\}$. Based on the top-$r$ singular values which explain most of the variance (e.g., 80%), we determine the Tucker ranks $\left\{\tilde{P}_{k,l_k}\right\}$ and $\left\{\tilde{Q}_d\right\}$ and then share the estimated ranks to the aggregator. Based on the values of estimated Tucker ranks, the aggregator will first rank them from the lowest to the highest and then communicate with all individual sites to test each rank set in sequence. Specifically, the aggregator will assign each rank set to all individual sites and then start the proposed federated framework. After the algorithm stops, each individual site calculates the following Akaike Information Criterion AIC_m (Roy and Michailidis 2022; Lee et al. 2023) and then sends it back to the aggregator:

$$AIC_m = 2\sum_{k=1}^{K}\sum_{l_k=1}^{L_k} l_k + 2\sum_{d=1}^{D} d - 2\sum_{i=1}^{N_s^m}\left|\left|\mathcal{Y}^m - \sum_{k=1}^{K}\mathcal{X}_k^m * \mathcal{B}_k^m\right|\right|_F^2. \tag{27}$$

The aggregator sums up all AIC_m, that is, $AIC = \sum_{m=1}^{M} AIC_m$, and selects the rank set which results in the lowest AIC. Thus, for each hyperparameter setting, we can determine its corresponding best Tucker rank set.

By selecting the lowest AIC from the hyperparameter setting and associated best Tucker rank set, we finalize the selection of both hyperparameters and Tucker ranks.

4. Performance Evaluation Using Simulation Studies

In this section, we conduct two sets of simulation studies to evaluate the performance of the proposed method, considering two scenarios: (i) the inputs are a functional curve and an image, and (ii) the inputs are two images. Using the proposed framework, we obtain two types of models: aggregated model (3), and personalized models (4). Specifically, we first run Algorithm 2 to learn input bases, that is, site-specific input bases $\left\{\mathbf{U}^m_{k,i,l_k}\right\}$ and aggregated input bases $\left\{\mathbf{U}_{k,l_k}\right\}$. Since we have the equality constraint $\mathbf{U}_{k,l_k} = \mathbf{U}^m_{k,i,l_k}$ in (6), $\left\{\mathbf{U}^m_{k,i,l_k}\right\}$ and $\left\{\mathbf{U}_{k,l_k}\right\}$ share the same information, which does not require for further personalization among individual sites. However, for (17), we allow the deviations of site-specific features (i.e., $\left\{\mathcal{C}^m_k\right\}$, and $\left\{\mathbf{V}^m_d\right\}$) from the aggregated features (i.e., $\{\mathcal{C}_k\}$, and $\{\mathbf{V}_d\}$) by adding penalty terms (instead of equality constraints) to capture the heterogeneity among sites (Li et al. 2018). When Algorithm 4 converges after T iterations, we obtain the aggregated model with parameter tensors $\{\mathcal{B}_k\}$ constructed from the aggregated features. Then, to emphasize the differences among individual sites and achieve a better local fitting, we further personalize output bases $\left\{\mathbf{V}^m_d\right\}$ and core tensors $\left\{\mathcal{C}^m_k\right\}$ at each site by running an additional site-specific optimization in Section 3.2.2.1, which results in personalized output bases $\left\{\mathbf{V}^m_d\right\}$ and core tensors $\left\{\mathcal{C}^m_k\right\}$ locally. Finally, each site constructs a personalized model based on these personalized features.

We consider three types of models as benchmarks: (i) local models (1), that is, models trained locally using MTOT based on data from each individual site; (ii) a global model (2), that is, a model trained using MTOT based on the pooled data from all individual sites; and (iii) FedAvg (Brendan McMahan et al. 2016). The core concept of FedAvg is to average the individual features from each individual site, producing an aggregated feature at the aggregator level. However, this approach is not inherently designed for regression modeling for multimodal high-dimensional data sources. To gauge its efficacy, we adapted its central principle: each individual site updates its features by running MTOT independently, but subsequently sends their own features to the aggregator every ψ local updates ($\psi = 5$ in simulations). The aggregator then computes the average of these features and dispatches them back to the sites. These averaged features as used by individual sites to continue their local updates. Specifically, it is expected that the global model outperforms others since it learns model from the pooled raw data (Kim et al. 2017). The standardized prediction mean square error (SPME) is used as the evaluation metric, which is defined by $\text{SPME} = \left|\left|\mathcal{Y} - \hat{\mathcal{Y}}\right|\right|_F / ||\mathcal{Y}||_F$.

4.1. Simulation Setting

We simulate waveform surfaces $\mathcal{Y}^m$ based on two input tensors, $\mathcal{X}^m_1 \in \mathbb{R}^{N_s^m \times P_{1,1} \times \ldots \times P_{1,L_1}}$ and $\mathcal{X}^m_2 \in \mathbb{R}^{N_s^m \times P_{2,1} \times \ldots \times P_{2,L_2}}$, where N^m_s is the sample size of the mth site. Accordingly, we have

$$\mathcal{Y}^m = \sum_{k=1}^{2}\mathcal{X}_k^m * \mathcal{B}_k^m + \tau\mathcal{E}^m,$$
$$\mathcal{B}_k^m = \mathcal{C}_k^m \times_1 \mathbf{U}^m_{k,1} \times_2 \ldots \times_{L_k} \mathbf{U}^m_{k,L_k} \times_{L_k+1} \mathbf{V}^m_1 \times_{L_k+2} \ldots \times_{L_k+D} \mathbf{V}^m_D,$$

where τ is the noise level, and $\mathcal{E}^m$ is the error tensor. More simulation details can be found in Part VI of supplementary materials.

In Scenario 1, we assume that each individual site has $N^m_s = 80$ samples. The input is a combination of two types of images, that is, $\mathcal{X}^m_{1,i} \in \mathbb{R}^{80\times25\times20}$, $\mathcal{X}^m_{2,i} \in \mathbb{R}^{80\times20\times15}$, and the output is $\mathcal{Y}^m \in \mathbb{R}^{80\times15\times15}$. We set $\tilde{P}_{1,1} = \tilde{P}_{1,2} = 6, \tilde{P}_{2,1} = \tilde{P}_{2,2} = 5, \tilde{Q}_1 = \tilde{Q}_2 = 5$. It implies that $\mathcal{C}^m_1 \in \mathbb{R}^{6\times6\times5\times5}$ and $\mathcal{C}^m_2 \in \mathbb{R}^{5\times5\times5\times5}$. In Scenario 2, we generate a response from a functional curve and an image signal. Assuming that each individual site has $N^m_s = 60$ samples, we simulate $\mathcal{X}^m_{1,i} \in \mathbb{R}^{60\times20}$, $\mathcal{X}^m_{2,i} \in \mathbb{R}^{60\times20\times15}$, and $\mathcal{Y}^m \in \mathbb{R}^{60\times15\times15}$ with $\tilde{P}_{1,1} = 20, \tilde{P}_{2,1} = \tilde{P}_{2,2} = 6, \tilde{Q}_1 = \tilde{Q}_2 = 5$, which implies that $\mathcal{C}^m_1 \in \mathbb{R}^{20\times5\times5}$ and $\mathcal{C}^m_2 \in \mathbb{R}^{6\times6\times5\times5}$.

Besides, we randomly select 80% of data in each individual site for model training and use the remaining data for performance testing. We train the model using the training set and then calculate the SPME based on the test data. We replicate this process 30 times for each experimental setting to compute the mean and standard deviation of the performance metric.

4.2. Performance Evaluation: Impact of Noise

To test the model robustness, we evaluate the model performance under varying noise levels, that is, τ to be four levels as 0.0001, 0.001, 0.01, and 0.1. Here, we keep models across different sites to be homogeneous, that is, the distributions of $\left\{\mathcal{B}^m_k\right\}$ are the same among individual sites. The SPME results are reported in Table 1 (Scenario 1) and Table 2 (Scenario 2). The visualizations are provided in Figure 4 of Part I in supplementary materials. As it is presented in the tables, the personalized, aggregated, and global models significantly outperform the local models and models developed by FedAvg. Besides, personalized

Table 1. Testing errors in scenario 1 under different noise levels.

τ	Personalized	Aggregated	Local	Global	FedAvg
0.0001	3.63E-4 (2.92E-8)	3.63E-4 (2.92E-8)	1.51E-1 (6.59E-3)	1.81E-4 (7.29E-9)	1.84E0 (3.53E-3)
0.001	1.51E-3 (4.08E-9)	1.51E-4 (4.14E-9)	1.38E-1 (5.74E-3)	7.53E-4 (1.03E-9)	1.85E0 (4.35E-3)
0.01	1.42E-2 (8.98E-8)	1.42E-2 (8.51E-8)	1.55E-1 (8.79E-3)	7.08E-3 (1.95E-8)	1.85E0 (3.45E-3)
0.1	1.45E-1 (8.68E-6)	1.45E-1 (8.85E-6)	2.99E-1 (3.32E-3)	7.22E-2 (2.16E-6)	1.84E0 (3.51E-3)

NOTE: Variance in the bracket.

Table 2. Testing errors in scenario 2 under different noise levels.

τ	Personalized	Aggregated	Local	Global	FedAvg
0.0001	8.26E-4 (2.71E-7)	9.43E-4 (2.77E-7)	2.12E0 (2.89E-2)	4.01E-4 (6.94E-8)	1.87E0 (6.61E-3)
0.001	1.67E-3 (2.31E-8)	1.75E-3 (4.00E-8)	2.12E0 (4.08E-2)	8.25E-4 (6.03E-9)	2.90E0 (1.97E-3)
0.01	1.52E-2 (1.72E-7)	1.52E-2 (1.77E-7)	2.09E0 (2.74E-2)	7.55E-3 (4.48E-8)	3.91E0 (8.35E-4)
0.1	1.50E-1 (1.46E-5)	1.50E-1 (1.49E-5)	2.10E0 (3.46E-2)	7.44E-2 (4.29E-6)	1.87E0 (4.15E-3)

NOTE: Variance in the bracket

and aggregated models achieve a comparable prediction accuracy compared to the global model under relatively low noise levels. For example, as it is reported in Table 1, when $\tau = 0.01$, the mean SPME of the personalized and aggregated model are 0.0142 and 0.0142, respectively, which are significantly smaller than the mean SPME of the local model (0.155). This example further demonstrates the benefit of collaboration in model construction.

Moreover, federated models have a stable performance under relatively low noise levels. However, when the noise level increases, the global model achieves better performance compared to the federated models. This superior performance is because the global model pools all raw data directly and learn from the raw data while the federated models learn the information of transferred features. The increasing noise disturbs the information transmission and poses a challenge for federated models, which exhibits the importance of sample size to model robustness.

4.3. Performance Evaluation: Impact of Number of Sites and Model Heterogeneity

To assess the impact of number of sites and model heterogeneity, where the distributions of $\{\mathcal{B}_k^m\}$ vary across different sites, we evaluate the model performance in both homogeneous and heterogeneous settings across M sites ($M = 2, 3, 4$) under the noise level $\tau = 0.0001$. In the homogeneous setting, $\{\mathcal{B}_k^m\}$ is generated following Section 4.2. For the heterogeneous setting, we use $\{\mathcal{C}_k^m\}$ from Section 4.2 as an initial value. To this, we add an additional random value drawn from a distribution of $0.001 * \mathcal{N}(0, 1)$. This random addition to the initial value introduces heterogeneity to $\{\mathcal{B}_k^m\}$ across different sites. Tables 3 and 4 provide a summary of the SPME results for Scenarios 1 and 2. Both personalized and aggregated models still present comparable performance to the global model, and they significantly outperform the local model and the model constructed by FedAvg.

In the homogeneous settings, we observe that both personalized and aggregated models offer the similar prediction accuracy for Scenario 1 under different site numbers. However, for Scenario 2, the personalized model outperforms the aggregated model. For example, as it is reported in Table 4, when $M = 4$, the mean SPME of the personalized and aggregated model is 8.06×10^{-4} and 2.91×10^{-3}, respectively. This result underscores the significance of personalization, particularly when the system uses multimodal input types. Specifically, in Scenario 2, the inputs are a functional curve and an image, whereas Scenario 1 uses has two different-sized images.

In the heterogeneous setting, we find that the personalized model achieves much better performance than aggregated model, local model, and FedAvg, which further magnifies the necessity of personalization when model heterogeneity exists. For instance, when $M = 2$, the mean SPME of the personalized and aggregated model is 2.05×10^{-2} and 2.94×10^{-2}, respectively, when models are heterogeneous across sites, while the mean SPME is the same (3.63×10^{-4}) under the homogeneous setting. Moreover, Figure 5 in Part I of supplementary materials illustrates the performance metrics for each site across a range of site number M. While the personalized approach generally outperforms the aggregated model, the extent of performance enhancement is inconsistent across various sites. This variation could be attributed to the level of model heterogeneity, wherein the addition of new involved sites might exert either beneficial or detrimental effects on the model construction.

Since the local model has obviously worse performance and the aggregated model achieves the comparable result as the personalized model, we only show the personalized model and the global model in Figure 6 of Part I in supplementary materials to have a closer comparison. As it is shown from Figure 6, when the number of sites increases, the performance fluctuation of the global model is smaller than the personalized model. It further demonstrates the importance of sample size for model construction.

5. Case Studies

In this section, we conduct two case studies. The first case is to predict relative fuel ratio from operating signals in vehicle catalyst system. The second one is to evaluate the performance of the proposed method in collaborative image recovery.

5.1. Case I: Catalyst Stoichiometry Prediction

In this section, we consider that smart vehicles collaborate in the processing of sensor data to assist in safe navigation, pollution

Table 3. Testing errors in scenario 1 under varying site numbers.

M	Model heterogeneity	Personalized	Aggregated	Local	Global	FedAvg
2	Homogeneous	3.63E-4 (2.92E-8)	3.63E-4 (2.92E-8)	1.51E-1 (6.59E-3)	1.81E-4 (7.29E-9)	1.84E0 (3.53E-3)
	Heterogeneous	2.05E-2 (4.30E-5)	2.94E-2 (1.17E-4)	1.62E-1 (7.45E-3)	1.78E-2 (5.17E-5)	1.84E0 (4.07E-3)
3	Homogeneous	5.67E-4 (7.47E-8)	5.66E-4 (7.47E-8)	2.34E-1 (1.07E-2)	1.88E-4 (8.27E-9)	2.90E0 (2.34E-3)
	Heterogeneous	3.87E-2 (1.06E-4)	5.56E-2 (1.55E-4)	2.90E-1 (1.10E-2)	2.14E-2 (2.63E-5)	2.90E0 (2.10E-3)
4	Homogeneous	5.28E-4 (3.29E-8)	5.28E-4 (3.29E-8)	3.18E-1 (1.08E-2)	1.32E-4 (2.04E-9)	3.94E0 (6.37E-4)
	Heterogeneous	4.39E-2 (1.56E-4)	6.35E-2 (2.75E-4)	3.68E-1 (1.94E-2)	1.83E-2 (2.37E-5)	3.95E0 (5.43E-4)

NOTE: Variance in the bracket

Table 4. Testing errors in scenario 2 under varying site numbers. (variance in the bracket).

M	Model heterogeneity	Personalized	Aggregated	Local	Global	FedAvg
2	Homogeneous	8.26E-4 (2.71E-7)	9.43E-4 (2.77E-7)	2.12E0 (2.89E-2)	4.01E-4 (6.94E-8)	1.87E0 (6.61E-3)
	Heterogeneous	2.90E-2 (4.98E-4)	4.36E-2 (4.39E-4)	2.12E0 (3.26E-2)	2.65E-2 (1.36E-4)	1.89E0 (2.65E-3)
3	Homogeneous	4.02E-4 (9.80E-9)	1.29E-3 (4.72E-8)	3.2E0 (1.14E-1)	1.18E-4 (1.40E-9)	1.87E0 (3.87E-3)
	Heterogeneous	3.44E-2 (1.46E-4)	5.88E-2 (2.01E-4)	3.15E0 (9.98E-2)	2.44E-2 (3.84E-5)	2.90E0 (1.65E-3)
4	Homogeneous	8.06E-4 (9.24E-8)	2.91E-3 (2.97E-7)	4.34E0 (1.11E-1)	1.80E-4 (6.79E-9)	1.88E0 (4.93E-3)
	Heterogeneous	4.98E-2 (2.04E-4)	8.80E-2 (3.51E-4)	4.37E0 (1.14E-1)	2.59E-2 (4.52E-5)	3.91E0 (6.22E-4)

Table 5. SPME results for catalyst stoichiometry prediction (variance in the bracket).

Personalized	Aggregated	Local	Global	FedAvg
5.96E-1 (8.76E-4)	5.95E-1 (9.87E-4)	2.43E0 (1.20E-4)	4.04E-1 (6.20E-4)	6.25E0 (1.96E-4)

control, and traffic management. Specifically, we consider two onboard catalyst systems from different vehicles collaborate in the data processing, but the system owners are not willing to share their data directly. Here, the catalyst system designed to treat the exhaust gas produced by vehicles , that is, NO_x Storage Catalyst (NSC). The NSC process has two alternating stages: (i) absorption, that is, NO_x molecules are absorbed by zeolites coated converter support; and (ii) regeneration: that is, the stored NO_x is reduced by catalyst when the absorber is saturated. Typically, the optimal combustion is required to ensure the ideal conversion rate of the catalytic converter for the second stage. Besides, NSC only works efficiently at stoichiometric status, which requires combustion in a rich-air-to-fuel condition. To indicate whether the regeneration stage is in good condition, we use the relative fuel ratio normalized by stoichiometry, which is measured runtime by a sensor upstream of the NSC. Thus, it is worth developing a generalizable model that could provide a good estimation of the stoichiometry signal based on the operation signals collected by onboard sensors, such as rotational speed and inner torque.

Each system performs 171 experiments to gather 171 sample pairs, containing five operating signals as inputs and one stoichiometry signal as the model response (Gahrooei et al. 2019, 2021). Figure 7 in Part I of supplementary materials illustrates the sample of the real data. Specifically, each system collects one measurement for each signal every 2 sec and has 203 measurements in total for each signal. For each site, we randomly select 136 samples as the training set and the remaining samples are used for model testing. Based on the training set, we estimate the model parameters and then calculate the SPME using the test data.

Table 5 reports that the personalized model achieves a comparable performance as the global model while improving the performance by around 68.5% compared with the local model. Besides, since the data are collected from two real operating systems and the systems could not be identical, the personalized model has relatively better performance than the aggregated model under the federated framework, which again validates the importance of personalization in the real model construction.

5.2. Case II: Image Denoising

In this section, we conduct experiments to validate the image denoising application of our proposed method motivated by Zhou, Li, and Zhu (2013). In this application, we assume two individual sites collaborate to recover two corrupted images. We denote the kth noisy image at the mth site by $I^n_{m,k} \in \mathbb{R}^{P_{k,1}\times P_{k,2}}$. To apply our method in the image recovering application, we perform the following procedure in each site. First, each site m generates a set of random observation tensor (with sample size N^m_s), denoted as $\mathcal{X}^m_k \in \mathbb{R}^{P_{k,1}\times P_{k,2}}$, for the kth image and then combines the weighted observation $\mathcal{X}^m_k * I^n_{m,k}$, $k \in \{1,2\}$, and noise $\tau\mathcal{E}^m$, $\mathcal{E}^m \sim \mathcal{N}(0,1)$, to produce $\mathcal{Y}^m$ as follows,

$$\mathcal{Y}^m = \sum\nolimits_{k=1}^{2} \mathcal{X}^m_k * I^n_{m,k} + \tau\mathcal{E}^m, m = \{1,2\}.$$

Each observation tensor $\mathcal{X}^m_k$ is generated as follows: the core tensor is generated from $\mathcal{N}(0,1)$ and bases $\left\{\mathbf{U}^m_{k,l_k}\right\}$ are learned from the Tucker decomposition of $I^n_{m,k}$. Given N_s pairs of observations and response tensors $(\mathcal{Y}^m, \left\{\mathcal{X}^m_k\right\})$, each individual site aims to recover the $I^n_{m,k}$ by applying the proposed method.

The denoising problem can be formulated as a learning problem that can be solved by MTOT whose estimated parameters are the recovered version of the clean image I_k. We denote the kth denoised image at site m based on the global model, local model, personalized model, aggregated model by I^g_k, $I^l_{m,k}$, $I^p_{m,k}$, and $I^a_{m,k}$, respectively. To test denoising effects, we use the

Table 6. ISNR results for image denoising and reconstruction.

Model Scenario	Personalized	Global	Local	FedAvg	Noisy
1	1.33E-1	1.16E-1	1.10E0	2.00E0	2.59E0
2	1.74E-1	1.70E-1	1.46E0	1.91E0	2.59E0

inverse of the signal to noise ratio (ISNR), that is, $ISNR = \sum_{m=1}^{M}\sum_{k=1}^{K}\frac{\left|\left|I^n_{m,k}-I_k\right|\right|_F}{||I_k||_F}$, to be the evaluation metric.

We consider two scenarios of different types of $I^n_{m,k}$ as shown in Figure 8 of Part I in supplementary materials. The noise level τ is 0.00002 and 0.00003 for site 1 and 2, respectively. We assume that each individual site has 50 and 80 samples in Scenarios 1 and 2, respectively. By applying the proposed framework and benchmark methods, we estimate the model parameters (i.e., the recovered images). The ISNR results are summarized in Table 6 and denoised images are illustrated in Figures 9 and 10 of Part I in supplementary materials. As it is reported, the proposed personalized model significantly outperforms local models and achieves comparable performance to the global model in denoising images. As shown in Figures 9 and 10, local models fail to learn the background under the red rectangular, while the personalized model can address the issue due to the collaboration under the proposed federated framework.

6. Conclusion

This article proposes a federated multiple tensor-on-tensor regression (FedMTOT) framework to follow the data management policies and decrease data storage costs. In the proposed framework, the input bases, core tensor, and output bases from multimodal data sources are learned iteratively in a federated fashion to avoid direct data sharing but still maintain a similar model performance. Finally, we use two sets of simulations and two case studies to test the model effectiveness in both response prediction and image denoising. Our results show that the personalized model under the federated setting outperforms the model trained only using local data via MTOT, which validates the superiority of the proposed framework. Several future directions can be envisioned. First, this article assumes all the local sites have access to all data modalities which allows them to construct the same models to be aggregated. However, missing data modality and samples is possible and requires further investigations. Furthermore, the proposed method is an offline method. However, often data is continuously generated and can be used to improve the model. Developing the online versions of the proposed method should be investigated in future research. Finally, the positive and negative impact of each involved site in collaboratively constructing an aggregated model should be quantified as a future direction of research.

Supplementary Materials

PDF supplement: In the online supplementary materials of this article, we provide a PDF file that contain further simulation and case study results, detailed derivations of variable updates, and convergence analysis of the proposed algorithm. ***Matlab code***: We provide Matlab implementation of the proposed algorithm for reproducing Figure 9 in this article.

Acknowledgments

We would like to thank the editor, associate editor, and the referees for their constructive comments and suggestions that considerably improved the article.

Disclosure Statement

The authors report there are no competing interests to declare.

Funding

This work has been partially supported by the National Science Foundation (NSF) award 2212878.

ORCID

Zihan Zhang http://orcid.org/0000-0002-5882-055X

References

Acar, D. A. E., Zhao, Y., Navarro, R. M., Mattina, M., Whatmough, P. N., and Saligrama, V. (2021), "Federated Learning Based on Dynamic Regularization," arXiv:2111.04263. [2]

Brendan McMahan, H., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. Y. (2016), "Communication-Efficient Learning of Deep Networks from Decentralized Data," arXiv:1602.05629. [2,9]

Fang, X., Paynabar, K., and Gebraeel, N. (2019), "Image-Based Prognostics Using Penalized Tensor Regression," *Technometrics*, 61, 369–384. [2]

Feng, J., Yang, L. T., Zhu, Q., and Choo, K. R. (2020), "Privacy-Preserving Tensor Decomposition over Encrypted Data in a Federated Cloud Environment," *IEEE Transactions on Dependable and Secure Computing*, 17, 857–868. [2]

Gahrooei, M. R., Paynabar, K., Pacella, M., and Shi, J. (2019), "Process Modeling and Prediction with Large Number of High-Dimensional Variables Using Functional Regression," *IEEE Transactions on Automation Science and Engineering*, 17, 684–696. [11]

Gahrooei, M. R., Yan, H., Paynabar, K., and Shi, J. (2021), "Multiple Tensor-on-Tensor Regression: An Approach for Modeling Processes with Heterogeneous Sources of Data," *Technometrics*, 63, 147–159. [1,2,4,5,11]

Gaw, N., Yousefi, S., and Gahrooei, M. R. (2022), "Multimodal Data Fusion for Systems Improvement: A Review," *IISE Transactions*, 54, 1098–1116. [1]

Gordan, M., Sabbagh-Yazdi, S., Ismail, Z., Ghaedi, K., Carroll, P., McCrum, D., and Samali, B. (2022), "State-of-the-Art Review on Advantages of Data Mining in Structural Health Monitoring," *Measurement*, 193, 110939. [1]

Kim, Y., Sun, J., Yu, H., and Jiang, X. (2017), "Federated Tensor Factorization for Computational Phenotyping," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 887–895. [9]

Kolda, T. G., and Bader, B. W. (2009), "Tensor Decompositions and Applications," *SIAM Review*, 51, 455–500. [4]

Kontar, R., Shi, N., Yue, X., Chung, S., Byon, E., Chowdhury, M., Jin, J., Kontar, W., Masoud, N., Nouiehed, M., Okwudire, C. E., Raskutti, G., Saigal, R., Singh, K., and Ye, Z.-S. (2021), "The Internet of Federated Things (IoFT)," *IEEE Access.*, 9, 156071–156113. [2]

Konyar, E., and Reisi Gahrooei, M. (2023), "Federated Generalized Scalar-on-Tensor Regression," *Journal of Quality Technology*, 56, 20–37. DOI:10.1080/00224065.2023.2246600 [8]

Lee, H. Y., Reisi Gahrooei, M., Liu, H., and Pacella, M. (2023), "Robust Tensor-on-Tensor Regression for Multidimensional Data Modelling," *IISE Transactions*, 56, 43–53. DOI:10.1080/24725854.2023.2183440 [2,8,9]

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2018), "Federated Optimization in Heterogeneous Networks," arXiv:1812.06127. [2,9]

Liu, X., Duan, R., Luo, C., Ogdie, A., Moore, J. H., Kranzler, H. R., Bian, J., and Chen, Y. (2022), "Multisite Learning of High-Dimensional Heterogeneous Data with Applications to Opioid Use Disorder Study of 15,000 Patients across 5 Clinical Sites," *Science Report*, 12, 11073. [1]

Lock, E. F. (2018), "Tensor-on-Tensor Regression," *Journal of Computational and Graphical Statistics*, 27, 638–647. [2]

Luo, R., and Qi, X. (2017), "Function-on-Function Linear Regression by Signal Compression," *Journal of the American Statistical Association*, 112, 690–705. [1]

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. Y. (2017), "Communication-Efficient Learning of Deep Networks from Decentralized Data," Artificial Intelligence and Statistics, arXiv:1602.05629. [2]

Orús, R. (2019), "Tensor Networks for Complex Quantum Systems," *Nature Reviews Physics*, 1, 538–550. [1]

Pathak, R., and Wainwright, M. J. (2020), "FedSplit: An Algorithmic Framework for Fast Federated Optimization," *Advances in Neural Information Processing Systems*, 33, 7057–7066. [2]

Roy, S., and Michailidis, G. (2022), "Regularized High Dimension Low Tubal-Rank Tensor Regression," *Electronic Journal of Statistics*, 16, 2683–2723. [9]

Shi, J. (2023), "In-Process Quality Improvement: concepts, Methodologies, and Applications," *IISE Transactions*, 55, 2–21. [1]

Wang, Q., Jin, J., Liu, X., Zong, H., Shao, Y., and Li, Y. (2022), "Tensor Decomposition Based Personalized Federated Learning," arXiv:2208.12959 [2]

Yan, H., Paynabar, K., and Pacella, M. (2019), "Structured Point Cloud Data Analysis via Regularized Tensor Regression for Process Modeling and Optimization," *Technometrics*, 61, 385–395. [2,4]

Yan, H., Paynabar, K., and Shi, J. (2015), "Image-Based Process Monitoring Using Low-Rank Tensor Decomposition," *IEEE Transactions on Automation Science and Engineering*, 12, 216–227. [1]

Yue, X., Kontar, R. A., and Gómez, A. M. E. (2022), "Federated Data Analytics: A Study on Linear Models," *IISE Transactions*, 56, 16–28. DOI:10.1080/24725854.2022.2157912 [2]

Zhang, Z., Mou, S., Paynabar, K., and Shi, J. (2023), "Tensor-Based Temporal Control for Partially Observed High-Dimensional Streaming Data," *Technometrics*. DOI:10.1080/00401706.2023.2271060 [1]

Zhao, Q., Caiafa, C. F., Mandic, D. P., Chao, Z. C., Nagasaka, Y., Fujii, N., Zhang, L., and Cichocki, A. (2012), "Higher Order Partial Least Squares (HOPLS): a Generalized Multi- Linear Regression Method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 1660–1673. [2]

Zhao, M., Reisi Gahrooei, M., and Gaw, N. (2022), "Robust Coupled Tensor Decomposition and Feature Extraction for Multimodal Medical Data," *IISE Transactions on Healthcare Systems Engineering*, 13, 117–131. DOI:10.1080/24725579.2022.2141929 [1]

Zhong, Z., Paynabar, K., and Shi, J. (2023), "Image-Based Feedback Control Using Tensor Analysis," *Technometrics*, 65, 305–314. DOI:10.1080/00401706.2022.2157880 [1]

Zhou, H., Li, L., and Zhu, H. (2013), "Tensor Regression with Applications in Neuroimaging Data Analysis," *Journal of the American Statistical Association*, 108, 540–552. [1,11]