Communication Efficient Federated Generalized Tensor Factorization for Collaborative Health Data Analytics

Jing Ma, Qiuchen Zhang, Jian Lou*, Li Xiong, Joyce C. Ho
Emory University
jing.ma,qiuchen.zhang,jian.lou,lxiong,joyce.c.ho@emory.edu

ABSTRACT

Modern healthcare systems knitted by a web of entities (e.g., hospitals, clinics, pharmacy companies) are collecting a huge volume of healthcare data from a large number of individuals with various medical procedures, medications, diagnosis, and lab tests. To extract meaningful medical concepts (i.e., phenotypes) from such higher-arity relational healthcare data, tensor factorization has been proven to be an effective approach and received increasing research attention, due to their intrinsic capability to represent the high-dimensional data. Recently, federated learning offers a privacy-preserving paradigm for collaborative learning among different entities, which seemingly provides an ideal potential to further enhance the tensor factorization-based collaborative phenotyping to handle sensitive personal health data. However, existing attempts to federated tensor factorization come with various limitations, including restrictions to the classic tensor factorization, high communication cost and reduced accuracy. We propose a *communication efficient* federated *generalized* tensor factorization, which is flexible enough to choose from a variate of losses to best suit different types of data in practice. We design a three-level communication reduction strategy tailored to the generalized tensor factorization, which is able to reduce the uplink communication cost up to 99.90%. In addition, we theoretically prove that our algorithm does not compromise convergence speed despite the aggressive communication compression. Extensive experiments on two real-world electronics health record datasets demonstrate the efficiency improvements in terms of computation and communication cost.

CCS CONCEPTS

• **Information systems → Data extraction and integration**; • **Applied computing → Health informatics**.

KEYWORDS

Electronic Health Records (EHR), Tensor Factorization, Federated Learning, Computational Phenotyping

ACM Reference Format:
Jing Ma, Qiuchen Zhang, Jian Lou*, Li Xiong, Joyce C. Ho. 2021. Communication Efficient Federated Generalized Tensor Factorization for Collaborative Health Data Analytics. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3449832

1 INTRODUCTION

Recent years have witnessed an unprecedented growth of health data (e.g., in the form of EHR, electronic health records) being collected from a variety of institutions, including hospitals, clinics, pharmaceutical companies, and health insurance providers. Computational phenotyping, the process of extracting meaningful and concise medical concepts (i.e., phenotypes) from the health data, is an indispensable stepping stone towards in-depth medical decision-making, including precision medicine, influenza surveillance, drug discovery, to name a few. Computational phenotyping is known to be challenging, given the fact that health data are collected from a large number of individuals with each one's medical record consisting of various of medical procedures, medications, diagnosis and lab tests. That is, the health data is massive and multidimensional. In addition, in order to collaboratively learn phenotypes from the data belonging to different institutes (known as collaborative phenotyping), the sensitive nature of the health data serves as an additional restriction.

To learn phenotypes from the multidimensional EHR data, tensor factorization has received increasing interest [12–14, 20, 27, 28, 36]. Tensor has the intrinsic capability to succinctly represent the multidimensional data [21] and has applications beyond health data analytics, e.g., recommender systems [18], spatio-temporal data analysis [26], computer vision [35], and signal processing [32]. The CANDECOMP/PARAFAC or canonical polyadic (CP) tensor factorization (TF) [7, 11] and its generalization GTF [15] are fundamental tools for analyzing the tensors. Despite their effectiveness and wide applications, the scalability is often a major issue preventing it from being applied to larger scale health datasets, which are commonly encountered nowadays. To improve the scalability of TF, distributed tensor factorization (DTF) methods [6, 9, 12, 20, 27, 31, 41] are capable of processing large tensors that cannot be dealt by a single machine. It also complies with the practical scenario for the health data which is collected and held across multiple physically distributed medical institutions.

Most recently, federated tensor factorization (FTF) methods [20, 27] are proposed as a better DTF paradigm for decentralized data in terms of privacy protection, while maintaining similar computational and storage scalability. It avoids communicating both the raw tensor and individual mode related variables to the server, which shares the same spirit of the more general federated learning [17], i.e., to learn a joint model across all the clients without communicating individual-level data. By avoiding sharing the raw tensor and the patient mode related variables (e.g., patient factor and partial

*Corresponding Author.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449832

gradient along the patient mode), FTF offers better patient privacy protection.

Besides computational complexity and alleviating storage usage which are the focus of most existing DTF methods, the communication overhead can be a third important bottleneck, especially for the federated setting, where the participating institutions do not have a dedicated communication network for communication purposes, e.g., hospitals, clinics. Considering the asymmetric bandwidths, the uplink communication (i.e. the communication from the client to the server) can quickly become the bottleneck preventing these clients from participating in the FTF. In federated computational phenotyping, due to the great variety of the attributes (e.g., types of medication can be thousands), the high dimensional tensor incurs high communications cost to communicate the intermediate variables during each communication cycle.

1.1 Contributions

In this paper, we investigate how to reduce the uplink communication cost of the federated tensor factorization-based collaborative phenotyping with guaranteed convergence and quality preservation. It is a challenging task, especially considering the communication efficiency issue is under studied in the broader distributed tensor factorization literature. To be more flexible and suitable for a variety of applications, we consider the federated generalized tensor factorization (FGTF), which greatly extends the existing federated classic TF [20, 27].

First, we aim to reduce the uplink communication cost in each communication round. We design a two-level per-round communication reduction strategy: block-level and element-level, which reduce $(1 - \frac{1}{D})$ and over 96.8% of the uplink communication, correspondingly, where D is the number of blocks. For the block-level, we exploit the multi-factor structure of TF/GTF by utilizing the randomized block update. It enables each client to send only the partial gradient of the sampled block, rather than the full gradient of all blocks. For the element-level, we introduce gradient compression techniques, which have found success in deep learning training [2, 4, 19, 37, 42], to compress each element of the communicated partial gradient from the floating point representation to low-precision representation. Since there exists error between the true partial gradient and the compressed one, the convergence can be slower and the output quality can be lower. We further introduce the error-feedback mechanism [19] which records such error and feeds it back to restore the shift.

With both levels of per-round communication reduction, we propose the federated GTF with communication compression and error-feedback **(FedGTF-EF)**. We analyze the convergence of **FedGTF-EF** and obtain the $O(\frac{1}{\sqrt{T}})$ rate after T iterations (Thm.4.1) under common and mild assumptions (Assumptions 4.1–4.5). The convergence is equivalent to the distributed stochastic gradient descent (SGD) with full precision gradient communication and distributed SGD with gradient compression and error-feedback [42]. In addition, since constraints and nonsmooth regularizations are common in GTF, we further extend the convergence result to the proximal setting (4.2) where the additional "simple regularizer" in Assumption 4.6 is satisfied. Compared to the existing analysis with gradient compression and error-feedback, our convergence analysis accounts for both the block randomized update strategy and the proximal operation.

Table 1: Symbols and notations used in this paper

Symbol	Definition
$\mathbf{x}, \mathbf{X}, \mathcal{X}$	Vector, Matrix, Tensor
$\mathcal{X}_{<d>}$	Mode-d matricization of $\mathcal{X}$
$\|\cdot\|_1$	ℓ_1-norm
$\|\cdot\|_F$	Frobenius norm
$\circledast$	Hadamard (element-wise) multiplication
$\odot$	Khatri Rao product
$\circ$	Outer product
$\langle\cdot,\cdot\rangle$	Inner product

Second, we reduce the number of communication rounds to further reduce the uplink communication. To do so, we introduce periodic communication [4, 23, 33] into **FedGTF-EF** and denote this algorithm as **FedGTF-EF-PC**, in which the clients send the update to the server after $\tau > 1$ local iterations instead of communicating after every iteration. A key question is whether the periodic communication will slow down the convergence. If so, the number of iterations will increase and the overall number of communications may not reduce. We analyze the convergence of **FedGTF-EF-PC** in Thm. 4.3 and obtain the same convergence $O(\frac{1}{\sqrt{T}})$ rate with **FedGTF-EF** under the same set of assumptions. This indicates that **FedGTF-EF-PC** can indeed further reduce the uplink communication cost by $1 - \frac{1}{\tau}$. As a result, our proposed **FedGTF-EF-PC** can reduce up to $1 - \frac{1}{32D\tau}$ uplink communication cost if the `Sign` compressor (Def.2.1) is used.

Third, we evaluate **FedGTF-EF** and **FedGTF-EF-PC** in the federated collaborative phenotyping task. We conduct experiments on two real-world EHR datasets, which show that the proposed method can effectively reduce uplink communication cost (99.90% reduction), without compromising convergence and factorization quality.

2 PRELIMINARIES AND BACKGROUND

2.1 Notation

The frequently used notation in this paper is summarized in Table 1. We denote an order D tensor by $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_D}$, its $(i_1, ..., i_D)$-th element by MATLAB representation $\mathcal{X}(i_1, ..., i_D)$. Let $\mathcal{I}$ denote the index set of all tensor entries, $|\mathcal{I}| = I_\Pi = \prod_{d=1}^{D} I_d$. The mode-$d$ unfolding (also called matricization) is denoted by $\mathbf{X}_{<d>} \in \mathbb{R}^{I_d \times I_\Pi / I_d}$, where $(\mathbf{X}_{<d>})(i_d, j)$ and $\mathcal{X}(i_1, i_2, ..., i_D)$ has the **index mapping:** $j = 1 + \sum_{\substack{k=1,\\ k\neq d}}^{D} (i_k - 1)J_k,\ J_k = \prod_{\substack{q=1,\\ q\neq d}}^{k-1} I_q$. Each column $\mathbf{X}_{<d>}(:, j)$ is called a mode-d fiber of $\mathcal{X}$.

2.2 Generalized Tensor Factorization

As illustrated in Fig.1, let us consider the EHR tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times, \dots, \times I_D}$, which consists of patient mode (I_1), diagnosis mode (I_2), medication mode (I_3), and so on. The regularized Generalized CANDECOMP-PARAFAC (GTF) [15] extracts the phenotypes by decomposing the EHR tensor into R phenotyps, where each consists of a patient factor, diagnosis factor, and a medication factor. GTF has the following

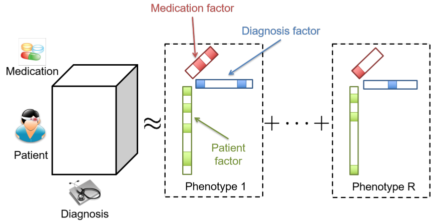

Figure 1: Illustration of EHR tensor and phenotype extraction via tensor factorization [14].

objective function:

$$\begin{aligned} \arg\min_{\mathcal{A}} F(\mathcal{A},\mathcal{X}) &= \sum_{i\in\mathcal{I}} f(\mathcal{A}(i),\mathcal{X}(i)) + \sum_{d=1}^{D} r_d(\mathbf{A}_{(d)}), \\ s.t.\ \mathcal{A} &= \sum_{i=1}^{R} \mathbf{A}_{(1)}(:,i)\circ\ldots\circ\mathbf{A}_{(D)}(:,i), \end{aligned} \tag{1}$$

which breaks down into three parts:

(1) Factorization constraint: The constraint of $\mathcal{A} = \sum_{i=1}^{R}\mathbf{A}_{(1)}(:,i)\circ\ldots\circ\mathbf{A}_{(D)}(:,i)$ approximates the low-rank CP tensor $\mathcal{A}\in\mathbb{R}^{I_1\times,\ldots,\times I_D}$ as the sum of R rank-one tensors, where $\mathbf{A}_{(d)}\in\mathbb{R}^{I_d\times R}$ is the d-th factor matrix and $\mathbf{A}_{(d)}(:,i)$ is its i-th column. For phenotyping, $\mathbf{A}_{(1)},\mathbf{A}_{(2)},\mathbf{A}_{(3)}$ correspond to the patient factor, diagnosis factor, and medication factor, correspondingly.

(2) Element-wise loss function: $f(\mathcal{A}(i),\mathcal{X}(i))$ is the element-wise loss between the low-rank CP tensor $\mathcal{A}$ and the input tensor $\mathcal{X}$. For the classic CP [7, 11], $f(\mathcal{A}(i),\mathcal{X}(i)) := \frac{1}{2}(\mathcal{A}(i)-\mathcal{X}(i))^2$, which is the least square loss. GCP is more generalized in the sense that the loss function can take other forms to best suit the property of the input tensor. For example, $f(\cdot)$ can be chosen based on the distribution of the tensor entries, e.g. logit loss for binary data: $f_{\text{logit}} = \log(1+\mathcal{A}(i)) - \mathcal{X}(i)\mathcal{A}(i)$, for all $i\in\mathcal{I}$, or $f(\cdot)$ can be the Huber loss for robustness purpose.

(3) Regularization: $r_d(\cdot)$ is the regularization applied to the factor $\mathbf{A}_d$, which can be the smooth $\|\mathbf{A}_{(d)}\|_F^2$ norm or the nonsmooth $\|\mathbf{A}_{(d)}\|_1$ norm. In practice, the regularization can improve the interpretability of the phenotypes.

Existing federated computational phenotyping. Two recent papers [20] and [27] consider federated tensor factorization and apply it to the federated phenotyping. They have the following limitations. 1) Both are limited to the CP model and [20] applies least square solver as its client side local updater, which is difficult to be extended to more general losses other than least square loss. 2) Although extensible to using infrequent communication, each communication round still incurs high communication cost since both requires sending all factors in full precision. In addition, [20] also requires communication of the Lagrangian dual variables which doubles the communication cost. 3) Both alter the original objective function by introducing extra terms to enforcing consensus of factors among all clients: [20] introduces linear constraint and transforms it to Lagrangian dual formulation while [27] introduces elastic penalty terms. These terms can lead the extracted factors to deviate from the centralized solution, thus negatively impacting the phenotyping accuracy.

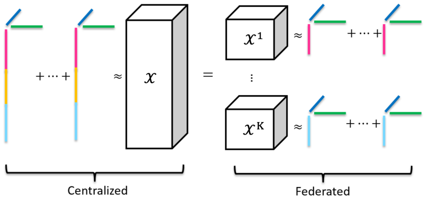

Figure 2: Illustration of collaborative phenotyping via federated tensor factorization [20].

2.3 SGD with Gradient Compression, Error-Feedback and Periodic Communication

Gradient Compression. Recently, one of the most successful approaches to mitigating the communication overhead is via gradient compression, which compresses the gradient to be communicated from the full precision representation (e.g. float or double number representation) to a much lower precision representation (e.g. aggressively compressed to 1-bit). The following definition introduces one of the most popular compressors:

Definition 2.1. (Sign Compressor) For an input tensor $\mathbf{x}\in\mathbb{R}^d$, its compression via $\texttt{Sign}(\cdot)$ is $\texttt{Sign}(\mathbf{x}) = \|\mathbf{x}\|_1/d\cdot \mathit{sign}(\mathbf{x})$, where *sign* takes the sign of each element of $\mathbf{x}$.

Error-Feedback. Due to aggressive compression, the algorithm can converge slower (or even diverge) compared to the full precision counterpart. The main cause is the error between the full precision gradient and the compressed one. Error-feedback [19, 34, 42] is a technique that memorizes this error in the current iteration and feeds it back to the gradient of the next iteration. By doing so, it can rigorously guarantee uncompromised convergence compared to the full-precision SGD.

Periodic Communication. Instead of reducing the communication cost per-communication round, periodic communication or local SGD [23, 33] reduces it by decreasing the communication frequency in hope that the total number of communications rounds can be reduced. Each clients will execute $\tau > 1$ local updates before communicating to the server. [4] shows that it is possible to combine communication compression and periodic communication together. [34] provides a unified framework by error-feedback to analyze the convergence of gradient compression and local SGD.

3 PROPOSED METHODS

Under the federated setting as illustrated in Fig.2, the EHR tensor $\mathcal{X}\in\mathbb{R}^{I_1\times,\ldots,\times I_D}$ will be collectively held by K institutions. The k-th client's local tensor is denoted by $\mathcal{X}^k\in\mathbb{R}^{I_{1k}\times I_2\times\ldots\times I_D}$, which contains information about I_{1k} individuals, such that $\sum_{k=1}^{K} I_{1k} = I_1$. That is, we consider the horizontally partitioned setting where different hospitals share the same feature space. We also note that there are related works addressing other settings like vertically partitioned settings [8, 24, 25, 39] which are complementary to our work. The aim of the federated computational phenotying is to collaboratively compute the phenotyes from EHR tensor across K institutions without sharing the raw tensor and patient mode

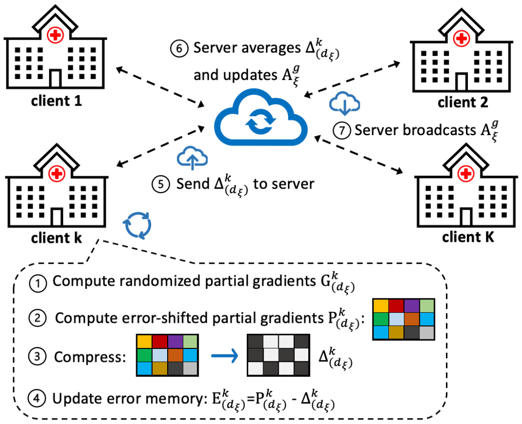
client 1
⑥ Server averages $\Delta^k_{(d_\xi)}$
and updates A^g_ξ
client 2
⑦ Server broadcasts A^g_ξ
client k
⑤ Send $\Delta^k_{(d_\xi)}$ to server
client K
① Compute randomized partial gradients $G^k_{(d_\xi)}$
② Compute error-shifted partial gradients $P^k_{(d_\xi)}$:
③ Compress:
$\Delta^k_{(d_\xi)}$
④ Update error memory: $E^k_{(d_\xi)}$=$P^k_{(d_\xi)}$ - $\Delta^k_{(d_\xi)}$

Figure 3: Illustration of the execution of FedGTF-EF and FedGTF-EF-PC.

variables. The objective function of the federated GTF is as follows

$$\underset{(\mathbf{A}_{(1)},\ldots,\mathbf{A}_{(D)})}{\text{argmin}} \sum_{k=1}^{K} F(\mathcal{A},\mathfrak{X}^k) + \sum_{d=1}^{D} r_d(\mathbf{A}_{(d)}),\quad s.t.\ \mathcal{A} = \mathbf{A}_{(1)} \circ \ldots \circ \mathbf{A}_{(D)}. \tag{2}$$

In fact, the above formulation can be extended to general multi-block problems as well. Thus, our algorithms are not limited to federated GTF problems but also to other nonconvex problems possessing a multi-block decision variable structure, e.g. [40]. In the following, we propose the federated generalized tensor factorization with communication efficiency improvements via block randomization, gradient compression, error feedback and periodic communication. The execution of the proposed algorithm is illustrated in Fig.3.

3.1 FedGTF-EF: Communication Efficient GTF with Block Randomization, Gradient Compression and Error-Feedback

We reduce the uplink communication in each communication round at two levels: block-level and element-level. The detailed algorithm is displayed in Algorithm 1 with functionalities of key steps annotated. At the block-level, to avoid sending all factors, we use a randomized block (i.e., randomized factor) update, which only requires the communication of the partial gradient of the factor being sampled (the computation of the partial gradient will be detailed in Sec.3.3). At the element-level, we compress each element of the communication to a low-precision representation before sending to the server (Line 6). Each client k keeps D local pairs of $\mathbf{P}^k_{(d)}$ (the error-shifted full-precision partial gradient), $\Delta^k_{(d)}$ (the compressed gradient to be communicated), $\mathbf{E}^k_{(d)}$ (error record between the full precision gradient and the compressed gradient), for all $d = 1, \ldots, D$ factors. Depending on whether the regularizer is smooth or not, either simple gradient descent (Line 8) or proximal gradient descent (Line 9) can be chosen to update the sampled factor, respectively.

Algorithm 1 FedGTF-EF: Communication Efficient GTF with Block Randomization, Gradient Compression and Error-Feedback

```
Input: $\mathfrak{X}, \gamma[t], \mathbf{A}[0]$, randomized block sampling sequence
    $d_\xi[0], \ldots, d_\xi[T]$;
1:  for $t = 0, \ldots, T$ do
2:      On Each Client Nodes $k \in 1, \ldots, K$:
3:      if $d = d_{(\xi)}[t]$ then
4:          Compute stochastic gradient $\mathbf{G}^k_{(d)}[t]$ by eq.(4);
5:          $\mathbf{P}^k_{(d)}[t] = \gamma[t]\mathbf{G}^k_{(d)}[t] + \mathbf{E}^k_{(d)}[t]$; %% error feedback
6:          $\Delta^k_{(d)}[t] = \texttt{Compress}(\mathbf{P}^k_{(d)}[t])$, Send $\Delta^k_{(d)}[t]$ (i.e. $\Delta^k_{(d_\xi[t])}[t]$)
            to the server; %% compression
7:          Receive $\frac{1}{K}\sum_{k=1}^{K}\Delta^k_{(d)}[t]$ (i.e. $\frac{1}{K}\sum_{k=1}^{K}\Delta^k_{(d_\xi[t])}[t]$) from the
            server;
8:          Smooth regularization case: $\mathbf{A}_{(d)}[t+1] = \mathbf{A}_{(d)}[t] -$
            $\frac{1}{K}\sum_{k=1}^{K}\Delta^k_{(d)}[t]$; %% update factor
9:          Nonsmooth regularization case: $\mathbf{A}_{(d)}[t+1] = \texttt{Prox}_{r_d}(\mathbf{A}_{(d)}[t] -$
            $\frac{1}{K}\sum_{k=1}^{K}\Delta^k_{(d)}[t])$;
10:         $\mathbf{E}^k_{(d)}[t+1] = \mathbf{P}^k_{(d)}[t] - \Delta^k_{(d)}[t]$; %% update error memory
11:     else if $d \neq d_\xi[t]$ then
12:         $\mathbf{A}_{(d)}[t+1] = \mathbf{A}_{(d)}[t]$, $\mathbf{E}^k_{(d)}[t+1] = \mathbf{E}^k_{(d)}[t]$; %% unselected
            blocks are kept unchanged
13:     end if
14:     On Server Node:
15:     Receive $\Delta^k_{(d_\xi[t])}[t]$ from all client nodes; Broadcast
        $\frac{1}{K}\sum_{k=1}^{K}\Delta^k_{(d_\xi[t])}[t]$ to all client nodes;
16: end for
```

3.2 FedGTF-EF-PC: Further Communication Reduction by Periodic Communication

We further reduce the uplink communication cost by introducing a third communication compression level: round level. That is, we decrease the communication frequency from one iteration per-communication to $\tau > 1$ iterations per-communication, which manifests a periodic communication behaviour [4, 23, 33]. The detailed algorithm is provided in Algorithm 2. The major difference with Algorithm 1 is that each client compresses and sends the collective updates across τ iterations (Line 9-10), instead of the partial gradient in a single iteration. The error feedback (Line 9) and error memory (Line 7, 13) are adjusted accordingly.

3.3 Efficient Partial Stochastic Gradient Computation for FedGTF

After presenting the overall algorithms, we now present an efficient partial stochastic gradient computation subroutine to compute $\mathbf{G}^k_{(d)}[t]$ in Step 1 of Fig.3 and Line 4 of Algorithm 1 and 2. The first mode (i.e., I_1) is the individual mode (e.g., patient mode) which can be kept local to each client. Thus, when $d_\xi[t] = 1$, we skip the communication, which not only further reduces the communication cost, but also is beneficial to the privacy since the individual-level information is not shared.

Next, we specify the computation of the partial stochastic gradient $\mathbf{G}^k_{(d)}[t]$ based on the efficient fiber sampling technique [5, 10]. The deterministic partial gradient is $\nabla_{\mathbf{A}_{(d)}} F(\mathbf{A}) = \mathbf{Y}_{<d>}\mathbf{H}_d$ [15], where $\mathbf{H}_d \in \mathbb{R}^{I_\Pi/I_d \times R}$ is the mode-d Khatri-Rao product of the all

Algorithm 2 FedGTF-EF-PC: Further Reducing Communication Cost by Periodic Communication

```
Input: $\mathcal{X}, \gamma[t], \mathbf{A}[0], \mathbf{A}^k[0] = \mathbf{A}[0], \forall k = 1, ..., K$, randomized block sampling sequence $d_\xi[0], ..., d_\xi[T]$;
for $t = 0, ..., T$ do
  On Each Client Nodes $k \in 1, ..., K$:
  if $d = d_{(\xi)}[t]$ then
    Compute stochastic gradient $\mathbf{G}^k_{(d)}[t]$ by eq.(4);
    $\mathbf{A}^k_{(d)}[t+\frac{1}{2}] = \mathbf{A}^k_{(d)}[t] - \gamma[t]\mathbf{G}^k_{(d)}[t]$; %% local update by stochastic gradient descent
    if $(t \mod \tau) \neq 0$ then
      $\mathbf{E}^k_{(d)}[t+1] = \mathbf{E}^k_{(d)}[t], \mathbf{A}^k_{(d)}[t+1] = \mathbf{A}^k_{(d)}[t+\frac{1}{2}], \mathbf{A}^g_{(d)}[t+1] = \mathbf{A}^g_{(d)}[t]$; %% no communication
    else
      $\mathbf{P}^k_{(d)}[t] = (\mathbf{A}^g_{(d)}[t] - \mathbf{A}^k_{(d)}[t+\frac{1}{2}]) + \mathbf{E}^k_{(d)}[t]$; %% error feedback to accumulated update
      $\Delta^k_{(d)}[t] = \texttt{Compress}(\mathbf{P}^k_{(d)}[t])$, Send $\Delta^k_{(d)}[t]$ (i.e. $\Delta^k_{(d_\xi[t])}[t]$) to the server;
      Receive $\mathbf{A}^g_{(d)}[t+1]$ from the server, $\mathbf{A}^k_{(d)}[t+1] = \mathbf{A}^g_{(d)}[t+1]$; %% compression
    end if
    $\mathbf{E}^k_{(d)}[t+1] = \mathbf{P}^k_{(d)}[t] - \Delta^k_{(d)}[t]$; %% update error memory
  else if $d \neq d_\xi[t]$ then
    $\mathbf{A}^k_{(d)}[t+1] = \mathbf{A}^k_{(d)}[t], \mathbf{E}^k_{(d)}[t+1] = \mathbf{E}^k_{(d)}[t]$;
  end if
  On Server Node:
  Receive $\Delta^k_{(d_\xi[t])}[t]$ from all client nodes; Broadcast $\mathbf{A}^g_{(d_\xi[t])}[t+1] = \mathbf{A}^g_{(d_\xi[t])}[t] - \frac{1}{K}\sum_{k=1}^K \Delta^k_{(d_\xi[t])}[t]$ to all client nodes;
end for
```

factors except the d-th, i.e. $\mathbf{H}_d = \mathbf{A}_{(D)} \odot ... \odot \mathbf{A}_{(d+1)} \odot \mathbf{A}_{(d-1)} ... \odot \mathbf{A}_{(1)}$; and $\mathbf{Y}_{<d>}$ is the d-unfolding of the element-wise partial gradient $\mathcal{Y} \in \mathbb{R}^{I_1 \times ... \times I_D}$, where $\mathcal{Y}(i) = \frac{\partial f(\mathcal{A}(i), \mathcal{X}(i))}{\partial \mathcal{A}(i)}$, for all $i \in \mathcal{I}$. We approximate $\nabla_{\mathbf{A}_{(d)}} F(\mathbf{A})$ by sampling $|\mathcal{S}|$ fibers (i.e. $|\mathcal{S}|$ columns of $\mathbf{Y}_{(d)}$) and the corresponding $|\mathcal{S}|$ rows of $\mathbf{H}_d$, where $\mathcal{S}$ denotes the index of the sampled fibers. The stochastic partial gradient is then

$$\mathbf{G}_{(d)}[t] = \mathbf{Y}_{<d>}(:, \mathcal{S})\mathbf{H}_d(\mathcal{S}, :), \tag{3}$$

where both $\mathbf{Y}_{<d>}(:, \mathcal{S})$ and $\mathbf{H}_d(\mathcal{S}, :)$ can be efficiently computed, because: 1) the computation of $\mathbf{Y}_{<d>}(:, \mathcal{S})$ only involves $I_d \times |\mathcal{S}|$ element-wise partial gradient computation [22] and 2) the computation of $\mathbf{H}_d(\mathcal{S}, :)$ can be obtained without forming the full Khatri-Rao product of $\mathbf{H}_d$ [32]. For the s-th row of $\mathbf{H}_d$, its index $(i_1^s, ..., i_D^s)$ can be obtained by the **index mapping** in Section 2.1. Then, $\mathbf{H}(s, :) = \mathbf{A}_{(1)}(i_1^s, :) \circledast ... \circledast \mathbf{A}_{(d-1)}(i_{d-1}^s, :) \circledast \mathbf{A}_{(d+1)}(i_{d+1}^s, :) \circledast ... \circledast \mathbf{A}_{(D)}(i_D^s, :)$, where $\circledast$ is the Hadamard product. Finally, the local stochatic gradient $\mathbf{G}^k_{(d)}[t]$ can be efficiently computed by substituting its local tensor partition $\mathbf{Y}^k$ and local factors $\mathbf{A}^k_{(d)}$ into eq.(3), which gives

$$\mathbf{G}^k_{(d)}[t] = \mathbf{Y}^k_{<d>}(:, \mathcal{S})\mathbf{H}^k_d(\mathcal{S}, :), \tag{4}$$

where $\mathbf{H}^k(s, :) = \mathbf{A}^k_{(1)}(i_1^s, :) \circledast ... \circledast \mathbf{A}^k_{(d-1)}(i_{d-1}^s, :) \circledast \mathbf{A}^k_{(d+1)}(i_{d+1}^s, :) \circledast ... \circledast \mathbf{A}^k_{(D)}(i_D^s, :)$. According to the complexity analysis, our gradient computation in eq.(4) matches the state-of-the-art efficiency of GTF computation, e.g., [10].

4 ALGORITHM ANALYSIS

This section presents the convergence analysis and complexity analysis of FedGTF-EF and FedGTF-EF-PC. A proof sketch of the convergence analysis is provided in the appendix.

4.1 Convergence Analysis

Assumptions. In order to analyze the convergence, we make the following assumptions which are common to many machine learning problems [4, 10, 34, 42]. Let the randomness of computing stochastic gradient of $\mathbf{G}_{(d_\xi[t])}[t]$ be $\zeta[t]$, the randomness of sampling the block be $\xi[t]$, the filtration upon iteration t be $\mathcal{F}[t] = \{\zeta[0], \xi[0], ..., \zeta[t-1], \xi[t-1]\}$.

Assumption 4.1. *(Block-wise Smoothness of the Loss Function) $F(\cdot)$ is $L_{(d)}$-block-wise smooth, for $d = 1, ..., D$, i.e. for all $\mathbf{A}, \mathbf{B}$, $F(\mathbf{B}) \leq F(\mathbf{A}) + \langle \nabla_{\mathbf{A}_{(d)}}, \mathbf{B}_{(d)} - \mathbf{A}_{(d)} \rangle + \frac{L_{(d)}}{2} \|\mathbf{B}_{(d)} - \mathbf{A}_{(d)}\|_F^2$.*

Assumption 4.2. *(Unbiased Gradient Estimation) The stochastic gradient is unbiased: $\mathbb{E}_{\zeta[t]}\left[\mathbf{G}^k_{d_{\xi[t]}}[t] \middle| \mathcal{F}[t], \xi[t]\right] = \nabla_{\mathbf{A}_{(d_{\xi[t]})}} F(\mathbf{A}[t])$.*

Assumption 4.3. *(Bounded Variance) The stochastic gradient has bounded variance:*
$\mathbb{E}_{\zeta[t]}\left[\|\mathbf{G}^k_{(d_\xi[t])}[t] - \nabla_{\mathbf{A}_{(d_\xi[t])}} F(\mathbf{A}[t])\|_F^2 \middle| \mathcal{F}[t], \xi[t]\right] \leq \sigma_d^2$.

Assumption 4.4. *(Bounded Gradient) $\|\nabla_{\mathbf{A}_{(d)}} F(\mathbf{A}[t])\|_F^2 \leq \omega_d^2$.*

Assumption 4.5. *(δ-approximated Compression [19]) An operator $\texttt{Compress} : \mathbb{R}^d \longrightarrow \mathbb{R}^d$ is an δ-approximate compressor for $\delta \in (0, 1]$ if $\|\texttt{Compress}(\mathbf{x}) - \mathbf{x}\|_2^2 \leq (1 - \delta)\|\mathbf{x}\|_2^2$.*

Many compressors satisfy the above condition [4]: top-k or random k-sparsification, stochastic k-level quantization, stochastic rotated quantization and the `Sign` compressor in Definition 2.3.

Assumption 4.6. *(Simple Regularization Function) The regularization functions $r_d(\cdot)$, $d = 1, ..., D$, are convex, lower semi-continuous and admit closed-form proximal operator:*
$\texttt{Prox}_{r_d}(\mathbf{B}_d) = \operatorname{argmin}_{\mathbf{A}_{(d)}} \frac{1}{2}\|\mathbf{A}_{(d)} - \mathbf{B}_{(d)}\|_F^2 + r_d(\mathbf{A}_{(d)})$.

Many common regularizations satisfy this assumption, for example, the ℓ_1-norm for inducing sparsity which has the soft-thresholding operator as its proximal operator.

4.1.1 *Convergence Analysis of Algorithm 1.*

Smooth regularization case. To prove the convergence, we extend the delayed gradient perspective in [19] to our block randomized setting by introducing the following virtual variables only for the proof: $\widetilde{\mathbf{A}}_{(d)}[t] := \mathbf{A}_{(d)}[t] - \frac{1}{K}\sum_{k=1}^K \mathbf{E}^k_{(d)}[t]$. Then, we have the following virtual recurrence: if $d = d_\xi[t]$, $\widetilde{\mathbf{A}}_{(d)}[t+1] = \mathbf{A}_{(d)}[t+1] - \frac{1}{K}\sum_{k=1}^K \mathbf{E}_{(d)}[t+1] = \widetilde{\mathbf{A}}_{(d)}[t] - \gamma[t]\frac{1}{K}\sum_{k=1}^K \mathbf{G}^k_{(d)}[t]$; else if $d \neq d_{\xi[t]}$, $\widetilde{\mathbf{A}}_{(d)}[t+1] = \widetilde{\mathbf{A}}_{(d)}[t]$. Thus, the recurrence can be viewed as the block randomized SGD with the variable $\widetilde{\mathbf{A}}_{(d)}[t]$ which corresponds to $\mathbf{A}_{(d)}[t]$ with delayed information $\frac{1}{K}\sum_{k=1}^K \mathbf{E}^k_{(d)}[t]$ added. The convergence of Algorithm 1 applied to the smooth smooth regularization is as follows.

Theorem 4.1. *Suppose that Assumptions 4.1-4.5 hold. Let $(\mathbf{A}_{(1)}[t], ..., \mathbf{A}_{(D)}[t])$ be the iterates of Algorithm 1 with Line 8.*

Let $\gamma = \min\{\frac{1}{2L}, \frac{\varrho}{\sqrt{T+1}/\sqrt{K}+\frac{(1-\delta)^{1/3}}{\delta^{2/3}}T^{1/3}}\}$*, for some* $\varrho > 0$*. We have*

$$\mathbb{E}[\frac{1}{D}\sum_{d=1}^{D} \|\nabla_{\mathbf{A}_{(d)}} F(\mathbf{A}[\mathtt{Output}])\|_F^2]$$
$$\leq \frac{8L}{T+1}(F(\mathbf{A}[0]) - F^*) + \Big[\frac{4}{\varrho}(F(\mathbf{A}[0]) - F^*) + \frac{2L\sigma^2\varrho}{D}\Big]\frac{1}{\sqrt{M(T+1)}}$$
$$+ \Big[\frac{4}{\varrho}(F(\mathbf{A}[0]) - F^*) + \frac{8L^2\varrho^2(\sigma^2+\omega^2)}{D}\Big]\frac{(1-\delta)^{1/3}}{\delta^{2/3}(T+1)^{2/3}},$$

where $\mathbf{A}[\mathtt{Output}] = (\mathbf{A}_{(1)}[\mathtt{Output}], ..., \mathbf{A}_{(D)}[\mathtt{Output}])$ *is sampled from* $\mathbf{A}[0]$ *to* $\mathbf{A}[T]$ *with uniform distribution,* F^* *is the optimal value,* $\sigma^2 = \sum_{d=1}^{D} \sigma_d^2$ *and* $\omega^2 = \sum_{d=1}^{D} \omega_d^2$*.*

Remark 1. Under the similar assumptions, our convergence rate matches the rates of the distributed synchronize SGD and the distributed SGD with gradient compression and error-feedback [42]. Thus, we can further reduce computation and uplink communication from a full-length gradient update and communication [4, 42] to a single randomized block of the partial gradient update and communication without slowing down the convergence rate.

Nonsmooth regularization case. This case corresponds to the execution of Line 9 in Algorithm 1. An appropriate optimally condition is based on the generalized gradient measure [10, 29, 30, 38]: $\widetilde{\mathbf{G}}_{(d)}[t] = \frac{1}{\gamma[t]}(\mathbf{A}_{(d)} - \mathsf{Prox}_{\gamma[t], r_d}(\mathcal{A}_{(d)}[t] - \gamma[t]\nabla_{\mathbf{A}_{(d)}} F(\mathbf{A}[t])))$. The following theorem shows the convergence of Algorithm 1 for the nonsmooth regularization case.

Theorem 4.2. *Suppose that Assumptions 4.1-4.6 hold. Let* $(\mathbf{A}_{(1)}[t], ..., \mathbf{A}_{(D)}[t])$ *be the iterates of Algorithm 1 with proximal operator (Line 9). Assume* $\gamma[t] = \frac{1}{4L}$*. We have*

$$\mathbb{E}\Big[\sum_{d=1}^{D} \frac{1}{D}\|\widetilde{\mathbf{G}}_{(d)}[\mathtt{Output}]\|_F^2\Big] \leq \frac{16L}{T+1}(\Phi(\mathbf{A}[0]) - \Phi^*)$$
$$+ \frac{4\sigma^2}{DK} + \frac{32(1-\delta)}{D\delta^2}(\sigma^2 + \omega^2), \tag{5}$$

where $\mathbf{A}[\mathtt{Output}]$ *is sampled from* $\mathbf{A}[0]$ *to* $\mathbf{A}[T]$ *with uniform distribution,* $\Phi(\mathbf{A}[0]) = F(\mathbf{A}[0]) + \sum_{d=1}^{D} r_d(\mathbf{A}[0])$ *and* Φ^* *is the optimal value.*

Remark 2. In the nonsmooth regularization case, the above convergence result is weaker than the previous smooth case in that we only ensure the difference between the initial loss and the optimal value will get smaller, but the generalized gradient is not guaranteed to approach 0 given that the variance and gradient norm related terms will dominate with increasing T. However, our empirical results show that the algorithm is able to converge to small losses.

4.1.2 Convergence Analysis of Algorithm 2. Now, we provide the convergence rate of Algorithm 2 by extending the proof in [4] to the block randomized setting, which is obtained under the same assumptions with Theorem 4.1. The main idea for the analysis is to introduce the virtual sequence of $\widetilde{\mathbf{A}}_{(d)}^{avg}[t+1] = \widetilde{\mathbf{A}}_{(d)}^{avg}[t] - \gamma[t]\frac{1}{K}\sum_{k=1}^{R} G_{(d)}^{k}[t]$ and build an iterative descent relation for it. Meanwhile, we keep track of the error between the true and virtual averages of $\mathbf{A}_{(d)}^{avg}[t] - \widetilde{\mathbf{A}}_{(d)}^{avg}[t]$, and the deviation between the local variables and the true average of $\mathbf{A}^{avg}[t] - \mathbf{A}^{k}[t]$. Since both deviations are well-bounded, it means $\mathbf{A}^{k}[t], \mathbf{A}^{avg}[t], \widetilde{\mathbf{A}}_{(d)}^{avg}[t]$ are close to each other. Finally, we can obtain the convergence result for the true sequence $\mathbf{A}^{k}[t]$ by substituting the deviations into the descent relation obtained for $\widetilde{\mathbf{A}}_{(d)}^{avg}[t]$.

Theorem 4.3. *Suppose that Assumptions 4.1-4.5 hold. Let* $(\mathbf{A}_{(1)}^{k}[t], ..., \mathbf{A}_{(D)}^{k}[t])$ *be the iterates of Algorithm 2, for* $k = 1, ..., K$ *and* $t = 0, ..., T$*. Let* $\gamma[t] = \frac{C}{\sqrt{T+1}}$ *with* $0 < C \leq \frac{1}{L}$*. We have*

$$\mathbb{E}[\sum_{d=1}^{D} \frac{1}{D}\|\nabla_{\mathbf{A}_{(d)}} F(\mathbf{A}[\mathtt{Output}])\|_F^2] \leq (4C[F(\mathbf{A}[0]) - F^*] + 2CL\sigma^2)\frac{1}{\sqrt{T+1}}$$
$$+ \Big(\frac{32C^2L^2(1-\delta^2)(\sigma^2+\omega^2)}{D\delta^2} + \frac{8C^2L^2(\sigma^2+\omega^2)}{DK}\Big)\frac{\tau^2}{T+1},$$

where $\mathbf{A}[\mathtt{Output}] = (\mathbf{A}_{(1)}[\mathtt{Output}], ..., \mathbf{A}_{(D)}[\mathtt{Output}])$ *is sampled from* $\mathbf{A}^{k}[0]$ *to* $\mathbf{A}^{k}[T]$*, for all* $k = 1, ..., K$*, with uniform distribution,* F^* *is the optimal value,* $\sigma^2 = \sum_{d=1}^{D} \sigma_d^2$ *and* $\omega^2 = \sum_{d=1}^{D} \omega_d^2$*.*

Remark 3. Algorithm 2 maintains the same convergence rate of $O(\frac{1}{\sqrt{T+1}})$ as Algorithm 1, despite the periodic communication. The communication gap τ only affects the term with order $O(\frac{1}{T+1})$, which is insignificant compared to the $O(\frac{1}{\sqrt{T+1}})$ overall convergence rate. Thus, without increasing the iteration complexity, the periodic communication can further reduce communication cost.

4.2 Complexity Analysis

We provide the computation, storage and communication complexities for FedGTF-EF and FedGTF-EF-PC given $|\mathcal{S}|$ fibers being sampled by each client and the rank of the GTF being R.

Computational Complexity. Our method is very efficient when compared to the following methods: 1) the classic CP-ALS and the full gradient descent-based GTF, which cost $O(DR\prod_{d=1}^{D} I_d)$; 2) the sampled randomized CP-ALS in [5] and SGD-based GTF in [15] with the same number of elements sampled, which cost $O(R\mathcal{S}|\sum_{d=1}^{D} I_d)$; and 3) the same complexity as the full precision block randomized SGD-based TF [10].

Theorem 4.4. *The per-iteration computational complexity of Algorithm **FedGTF-EF** and **FedGTF-EF-PC** for each client is* $O(\frac{1}{D}(\sum_{d=1}^{D} I_d)R|\mathcal{S}|)$*.*

Communication Complexity. Assume we are using the Sign compressor and comparing with full precision distributed SGD with all blocks communicated. Let $D = 4, \tau = 8$, **FedGTF-EF** and **FedGTF-EF-PC** reduces up to 99.22% and 99.90% uplink communications. In general, we have:

Theorem 4.5. ***FedGTF-EF*** *reduces up to* $1 - \frac{1}{32D}$ *uplink communication and* ***FedGTF-EF-PC*** *reduces up to* $1 - \frac{1}{32D\tau}$ *uplink communication.*

Storage Complexity. The fiber sampling based stochastic partial gradient avoids forming the whole element-wise partial gradient tensor $\mathcal{Y}$, which reduces the storage for this step from $O(\prod_{d=1}^{D} I_d)$ to $O(|\mathcal{S}|\frac{1}{D}\sum_{d=1}^{D} I_d)$, thus achieving the same cost efficiency with sampling-based random CP-ALS [5], full precision SGD [15] and block randomized full precision SGD [10].

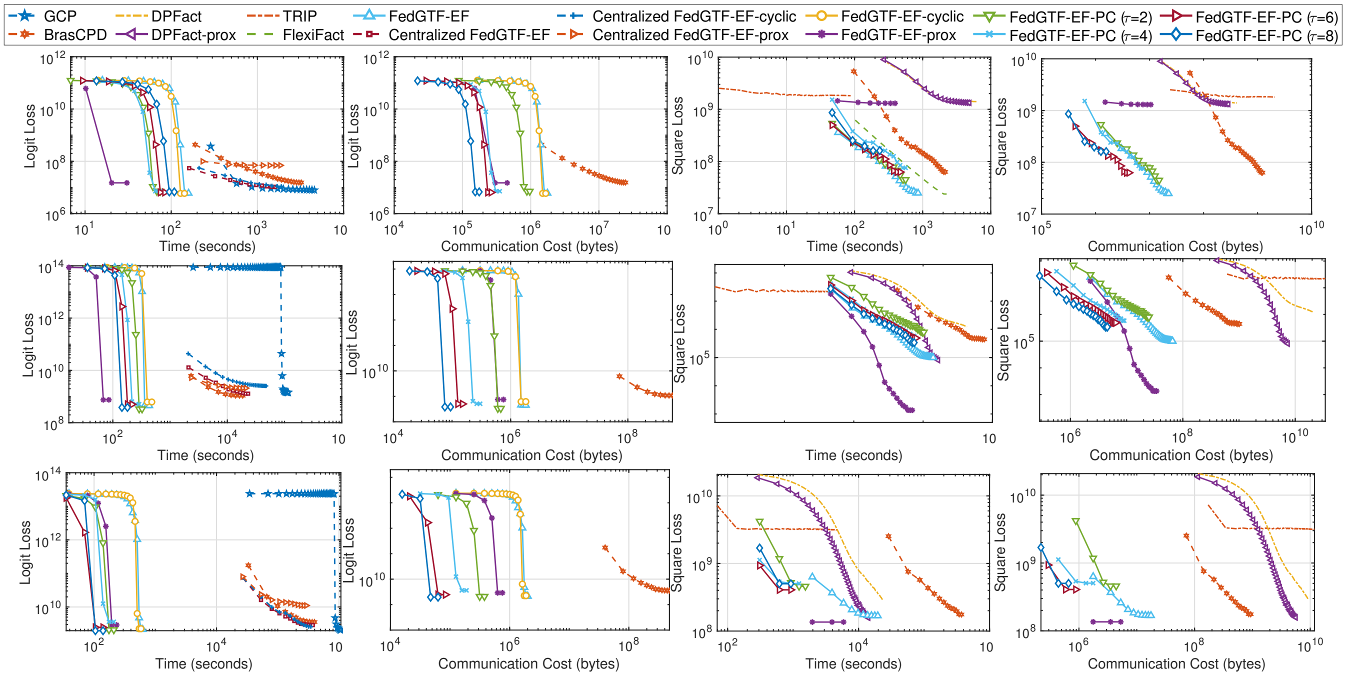

Figure 4: Loss decrease with respect to 1) computation time measured by seconds (column 1, 3 for Bernoulli Logit Loss and Least Square Loss respectively); 2) uplink communication cost measured by number of bytes (column 2, 4 for Bernoulli Logit Loss and Least Square Loss respectively). Top: 3-rd order CMS; Middle: 4-th order CMS; Bottom: MIMIC-III.

5 EXPERIMENT

5.1 Experimental Setup

Datasets. We consider two real-world EHR datasets[1], as well as a synthetic dataset, which are introduced below,

i). CMS [1] : A publicly available healthcare dataset with patients' information protected. We adopt the rules in [20] to select the top 500 frequently observed diagnoses, procedures, and medications to form a 4th order tensor of size $125,961 \times 500 \times 500 \times 500$ and a 3rd order tensor of size $91999 \times 500 \times 500$ (with medication mode omitted).

ii). MIMIC-III [16] : It is a publicly available relational dataset that describes the patients information of the Intensive Care Units (ICUs). Similar to CMS dataset, we form a 4 mode tensor representing patients-diagnoses-procedures-medications with size $34,272 \times 500 \times 500 \times 500$.

iii). Synthetic data : Synthetic data with size $4000 \times 500 \times 500 \times 500$ is generated as follows: for the nonzero entries, their values are sampled from uniform distribution for the least square loss setting and from binomial distribution for the logit loss setting, while positions of the non-zero entries are the same for both loss settings which are uniformly sampled from all entries with 10^{-4} non-zero ratio.

Algorithms for comparison. We consider two different loss functions: the Bernoulli logit loss f_{logit} and the least square loss. For the Bernoulli logit loss, we compare with: i) **GCP** (centralized) [22]; ii) **BrasCPD** (centralized) [10]; iii) **Centralized versions of FedGTF-EF**, iv) **FedGTF-EF-cyclic** and v) **FedGTF-EF-prox**. For the least square loss, we compare with: i) **BrasCPD** (centralized) [10]; ii) **FlexiFact** [6, 12]: a distributed tensor factorization algorithm; iii) **TRIP** [20]: a federated tensor factorization algorithm optimized with ADMM, which has deterministic per-iteration update solved in closed-form; iv) **DPFact** [27]: a federated SGD algorithm designed for collaborative tensor factorization. For fair comparison, we remove the differential privacy part of DPFact and substitute the $l_{2,1}$ regularization with the l_1 regularization as a new variant, DPFact-prox.

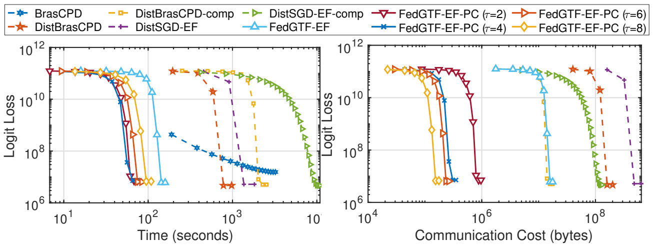

Figure 5: Ablation Study on 3-rd order CMS for Bernoulli Logit Loss.

Ablation study. We conduct ablation studies to illustrate the contribution of each communication reduction mechanism to the overall communication efficiency, which includes i) DistBrasCPD: the distributed version of BrasCPD [10] or FGTF with only the block-randomized technique; ii) DistBrasCPD-comp: FGTF with both block-randomized and gradient compression techniques; iii) DistSGD-EF: distributed SGD with error-feedback that communicates full gradients and all blocks; iv) DistSGD-EF-comp: DistSGD-EF with gradient compression. Table 2 summarize the comparison with the proposed algorithms.

[1] Code available at: https://github.com/jma78/FedGTF-EF

Table 2: Comparison of algorithms in ablation study.

Algorithm	Element-level Reduction	Block-level Reduction	Round-level Reduction	Convergence Guarantee	Compression Ratio
DistBrasCPD	✗	✓	✗	✗	$1 - 1/D$
DistBrasCPD-comp	✓	✓	✗	✗	$1 - 1/32D$
DistSGD-EF	✗	✗	✗	✗	0
DistSGD-EF-comp	✓	✗	✗	✗	$1 - 1/32$
FedGTF-EF	✓	✓	✗	✓	$1 - 1/32D$
FedGTF-EF-PC	✓	✓	✓	✓	$1 - 1/32D\tau$

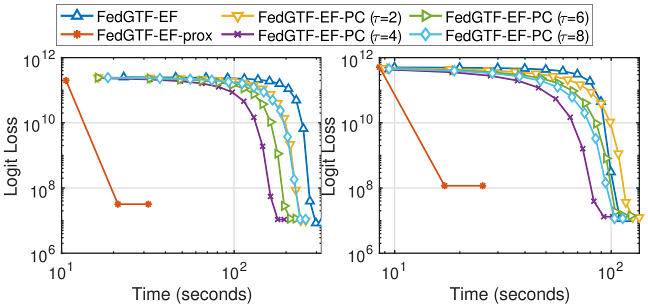

Figure 6: Comparison of different number of workers on 3-rd order CMS for Bernoulli Logit Loss.

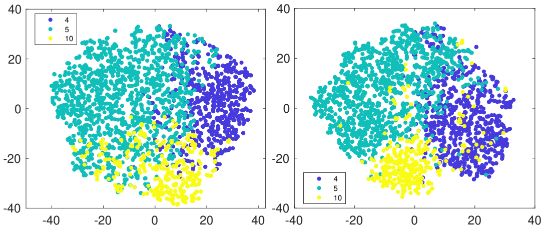

Figure 7: tSNE visualization of the patient representation learned by BrasCPD (left) and FedGTF-EF-PC(τ = 8) (right). Each point represents a patient which is colored according to the highest-valued coordinate in the patient representation vector among the top 3 phenotypes extracted based on the factor weights $\lambda_r = \|\mathbf{A}_{(1)}(:,r)\|_F \|\mathbf{A}_{(2)}(:,r)\|_F \cdots \|\mathbf{A}_{(D)}(:,r)\|_F$.

For our proposed algorithms, in addition to FedGTF-EF and FedGTF-EF-PC, we consider two variants: FedGTF-EF-cyclic (a variant of FedGTF-EF with cyclic mode updates), FedGTF-EF-prox (FedGTF-EF with l_1 regularization). We vary the value of τ in $\{2, 4, 6, 8\}$ for FedGTF-EF-PC.

Experiment results. Our experiments show that FedGTF-EF and FedGTF-EF-PC are able to greatly improve the communication efficiency without slowing down the convergence and deteriorating the factorization quality. In detail, we have the following four observations: **i)** FedGTF-EF and its variants reduce loss faster with much less communication cost, for both the Bernoulli Logit Loss (Fig.4 first two columns) and the Least Square loss (Fig.4 last two columns) compared to the baseline methods. The communication cost per communication round is further reduced by increasing the local update iterations τ from 2 to 8 without hurting the performance of the Bernoulli logit loss and with a slightly worse loss for the least square loss. **ii)** FedGTF-EF, FedGTF-EF-PC and their variants are computationally efficient due to the fiber-sampling technique, i.e., they use lower computation cost compared to the baselines. By Fig.4, for both objective functions, FedGTF-EF-PC, FedGTF-EF and its variants converges to similar losses as their centralized counterparts, while cost less time because more workers are involved in the updating process for the federated setting. Note that although TRIP converges faster in terms of time, but it tends to be trapped into bad local minima caused potentially by its deterministic per-iteration update. **iii)** FedGTF-EF, FedGTF-EF-PC and their variants converge

Table 3: Top 3 phenptypes extracted by FedGTF-EF-PC(τ = 8) on MIMIC-III data. Red, blue, and green indicate diagnoses, procedures, and medication, respectively.

P10: Diabetic Heart Failure
Diabetes mellitus without mention of complication
Background diabetic retinopathy
Acute systolic heart failure
Acute on chronic systolic heart failure
Chronic diastolic heart failure
Acute on chronic combined systolic and diastolic heart failure
Insertion of one vascular stent
Open heart valvuloplasty of tricuspid valve without replacement
Operations on other structures adjacent to valves of heart
(Aorto)coronary bypass of three coronary arteries
Captopril (ACE inhibitor), Insulin, Pyridostigmine Bromide, Isosorbide Dinitrate
P5: Hypertensive Heart Failure
Pure hypercholesterolemia
Cardiac tamponade
Ventricular fibrillation
Cardiac arrest
Acute systolic heart failure
Percutaneous insertion of carotid artery stent(s)
Pericardiocentesis
Extracorporeal circulation auxiliary to open heart surgery
Other endovascular procedures on other vessels
Rosuvastatin Calcium, Isosorbide Dinitrate, Hydrochlorothiazide, Digoxin, Clonidine HCl
P4: Peripheral Arterial Disease
Congestive heart failure
Atherosclerosis of native arteries of the extremities
– with intermittent claudication
Acute venous embolism and thrombosis of
–superficial veins of upper extremity
Insertion of drug-eluting coronary artery stent(s)
(Aorto)coronary bypass of two coronary arteries
Interruption of the vena cava
Suture of artery
Angioplasty of other non-coronary vessel(s)
Carvedilol, Metoprolol succinate, Amiodarone HCl, Nitroglycerin, Calcium Chloride

to similar losses as the centralized counterpart, which indicates communication efficiency can be improved without sacrificing the factorization quality. **iv)** FedGTF-EF and FedGTF-EF-PC converge faster in terms of running time with more workers. As shown in Fig.4 upper left and Fig.6, with the number of workers increased from 8 to 16, the time for FedGTF-EF to converge reduces 65.58%.

From the ablation study (Fig.5), we can see: **i)** Block-randomized update and gradient compression can greatly reduce the communication cost by 75.00% and 96.88%, respectively. Therefore, gradient compression plays a more important role in communication reduction. **ii)** With both block-randomized and gradient compression, FedGTF-EF achieves a gradient reduction of 98.90% over FGTF. **iii)** Periodic communication further reduces the communication cost over FGTF by 99.94%, 99.97, 99.98%, and 99.99% with $\{2, 4, 6, 8\}$ rounds of local communications respectively.

Finally, we evaluate the quality of the federated factorization factors by considering the patient subgroup identification following [28], as illustrated in Fig.7. We use tSNE to map the R dimensional vectors into the 2 dimensional space. We first identify the top 3 phenotypes that have the largest factor weights, which are the phenotypes $\#4, \#5, \#10$ in Fig.7 (phenotype details are shown in Table 3). Then, we color the patients by assigning each patient to one of the top 3 phenotypes using the largest patient weight among the top 3 along the representation vector. Fig.7 shows FedGTF-EF-PC with $\tau = 8$ local updates has comparable performance to the centralized baseline BrasCPD in clustering the patients with the same phenotype together. This demonstrates that our method can achieve communication compression without sacrificing the factorization quality.

6 CONCLUSION

In this paper, we study the under explored communication efficiency problem in federated (more broadly the distributed) generalized tensor factorization for collaborative phenotyping. We propose FedGTF-EF with communication efficient designs of block randomized update and gradient compression with error-feedback, which encompassed two levels of uplink communication reduction: reduced number of blocks and reduced per-element communication. We further reduce the communication rounds by periodic averaging to develop the FedGTF-EF-PC algorithm. The convergence guarantee is provided under common assumptions applied not only to generalized tensor factorization problems but also to more general machine learning problems possessing a multi-block structure. Our algorithm can maintain low computational and storage complexity while occupying much lower uplink communication cost. We demonstrate its superior efficiency and uncompromised quality on synthetic and two real-world EHR datasets.

ACKNOWLEDGMENTS

We sincerely thank all anonymous reviewers for their valuable comments. This work was supported by the National Science Foundation, awards IIS-#1838200 and CNS-1952192, National Institutes of Health (NIH) awards R01GM118609, 5K01LM012924, and CTSA UL1TR002378.

APPENDIX

A ADDITIONAL MATERIALS FOR EXPERIMENTS

A.1 Parameter Settings

For MIMIC-III, CMS and synthetic datasets, each algorithm is run for 500 iterations per epoch until converge, while for delicious dataset, each algorithm is run for 1000 iterations per epoch. For GCP algorithm, we tune the stepsize within the range of $\{10^{-8}, 10^{-9}, 10^{-10}, 10^{-11}\}$, while for the rest algorithms, we tune the stepsize by grid search through $\{2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, ..., 2^{-11}\}$. The parameter for the proximal operator is set to 10^{-4} for all the algorithms with the proximal operators (FedGTF-EF-prox, DPFact-prox). For all the federated algorithms, we by default horizontally partition the tensor (along I_1 mode) into 8 tensors without overlapping and distribute each of them to 8 client nodes respectively. We also test different numbers of workers (16 workers and 32 workers), where the stepsizes are set to the same as for 8 workers. The best stepsizes for each algorithms for different datasets are set as in Table 4 and 5.

Each experiment is averaged over 5 repetitions. All experiments are run on Matlab 2019a on an `r5.12xlarge` instance of AWS EC2 with Tensor Toolbox Version 3.1 [3].

A.2 Additional Experiments

Two additional groups of figures are presented here. Fig.8 shows the loss decrease for both the Bernoulli loss and the Least Square loss with respect to time and communication for the synthetic data. Fig.9 shows the Bernoulli loss and the Least Square loss decrease with respect to epochs in supplementary to the figures showed in the main paper with respects to time and communication. Similar conclusions can be drawn with the real-world EHR datasets in the main paper. That is, the proposed algorithms achieve more efficient convergence than the centralized baselines under the Bernoulli

Table 4: Best Stepsizes for the Bernoulli Logit Loss

Algorithm	MIMIC-III	4th order CMS	3rd order CMS	Synthetic
GCP	10^{-10}	10^{-10}	10^{-10}	10^{-9}
BrasCPD	2^{-4}	2^{-1}	2^{-4}	2^{-5}
Centralized FedGTF-EF	2^{-3}	2^{-1}	2^{-2}	2^{-4}
Centralized FedGTF-EF-cyclic	2^{-2}	2^{-2}	2^{-2}	2^{-4}
Centralized FedGTF-EF-prox	2^{-2}	2^{-0}	2^{-2}	2^{-2}
FedGTF-EF	2^{-3}	2^{-2}	2^{-2}	2^{-4}
FedGTF-EF-cyclic	2^{-4}	2^{-2}	2^{-2}	2^{-4}
FedGTF-EF-prox	2^{-2}	2^{-3}	2^{-4}	2^{-1}
FedGTF-EF-PC($\tau = 2$)	2^{-5}	2^{-5}	2^{-2}	2^{-4}
FedGTF-EF-PC($\tau = 4$)	2^{-5}	2^{-5}	2^{-2}	2^{-4}
FedGTF-EF-PC($\tau = 6$)	2^{-5}	2^{-5}	2^{-2}	2^{-4}
FedGTF-EF-PC($\tau = 8$)	2^{-5}	2^{-5}	2^{-2}	2^{-4}

Table 5: Best Stepsizes for the Least Square Loss

Algorithm	MIMIC-III	4-th order CMS	3-rd order CMS	Synthetic
BrasCPD	2^{-5}	2^0	10^{-4}	2^{-2}
FlexiFact	-	-	2	-
DPFact	2^{-4}	2^1	2^{-10}	2^{-2}
DPFact-prox	2^{-4}	2^1	2^{-10}	2^{-2}
FedGTF-EF	2^{-4}	2^0	2^{-11}	2^{-2}
FedGTF-EF-prox	2^{-5}	2^0	2^{-10}	2^{-2}
FedGTF-EF-PC($\tau = 2$)	2^{-4}	2^0	2^{-10}	2^{-2}
FedGTF-EF-PC($\tau = 4$)	2^{-4}	2^0	2^{-10}	2^{-2}
FedGTF-EF-PC($\tau = 6$)	2^{-4}	2^0	2^{-10}	2^{-2}
FedGTF-EF-PC($\tau = 8$)	2^{-4}	2^0	2^{-10}	2^{-2}

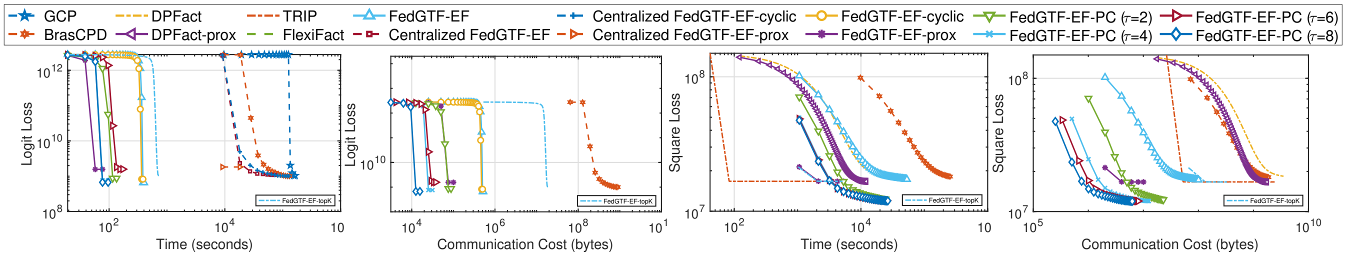

Figure 8: Bernoulli Logit Loss and Square Loss with respect to computation time and communication for synthetic data.

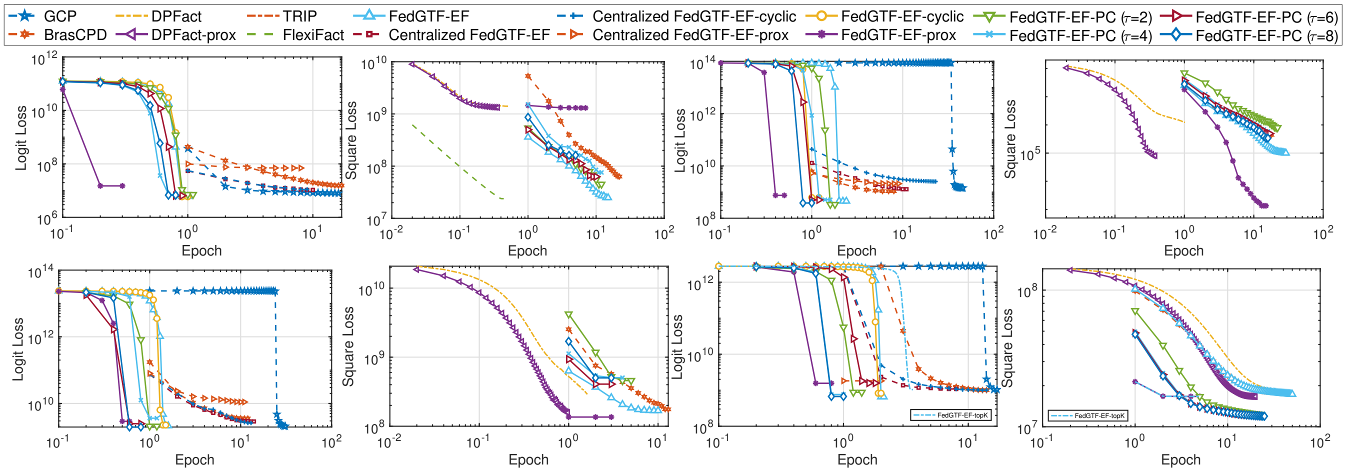

Figure 9: Bernoulli logit loss (column 1,2) and Least Square loss (column 3,4) decrease with respect to epochs.

logit loss and the distributed baseline under the least square loss. It is also more communication-efficient than the algorithms without gradient compressor (BrasCPD distributed version) and without the block randomized mechanism (DPFact and its variants).

B CONVERGENCE ANALYSIS OF ALGORITHM 1

B.1 Proof Sketch of Theorem 4.1

B.1.1 Auxiliary variables for the proof and iterative relation. The following auxillary variables and virtual iterations are introduced only for the proof: $\widetilde{\mathbf{A}}_{(d)}[t] := \mathbf{A}_{(d)}[t] - \frac{1}{K}\sum_{k=1}^{K}\mathbf{E}_{(d)}^{k}[t]$. Given the auxiliary variable $\widetilde{\mathbf{A}}_{(d)}[t]$, we have the following iterative relation: if $d = d_{\xi[t]}$, $\widetilde{\mathbf{A}}_{(d)}[t+1] = \widetilde{\mathbf{A}}_{(d)}[t] - \gamma[t]\frac{1}{K}\sum_{k=1}^{K}\mathbf{G}_{(d)}^{k}[t]$; else if $d \neq d_{\xi[t]}$, $\widetilde{\mathbf{A}}_{(d)}[t+1] = \widetilde{\mathbf{A}}_{(d)}[t]$.

B.1.2 Additional Lemma. The following lemma extends Lemma 3 in [19] to our block randomized case.

Lemma B.1. *(Bounding the expectation of the block-wise feedback error averaged among client nodes) For $d = 1, ..., D$ and for $t = 0, ..., T$, assuming constant step size $\gamma[t] = \gamma$, we have*

$$\mathbb{E}\Big[\|\frac{1}{K}\sum_{k=1}^{K}\mathbf{E}_{(d)}^{k}[t+1]\|_F^2\Big] \leq \frac{4(1-\delta)}{\delta^2}\gamma^2(\sigma_d^2+\omega_d^2). \qquad (6)$$

B.1.3 Main proof sketch of Theorem 4.1. By block-wise Lipschitz smoothness assumption of the loss function:

$$F(\widetilde{\mathbf{A}}[t+1]) \leq F(\widetilde{\mathbf{A}}[t]) - \gamma[t]\langle\nabla_{\mathbf{A}_{(d_{\xi[t]})}}F(\widetilde{\mathbf{A}}[t]), \frac{1}{K}\sum_{k=1}^{K}\mathbf{G}_{(d_{\xi[t]})}^{k}[t]\rangle$$
$$+\frac{L_{d_{\xi}[t]}(\gamma[t])^2}{2}\|\frac{1}{K}\sum_{k=1}^{K}\mathbf{G}_{(d_{\xi[t]})}^{k}[t]\|_F^2.$$

By Assumption 4.2 that $\mathbb{E}_{\zeta[t]}\Big[\frac{1}{K}\sum_{k=1}^{K}\mathbf{G}_{d_{\xi[t]}}^{k}[t]\Big|\mathcal{F}[t],\xi[t]\Big] = \nabla_{\mathbf{A}_{(d_{\xi[t]})}}F(\mathbf{A}[t])$, we have

$$\mathbb{E}_{\zeta[t]}\Big[\|\frac{1}{K}\sum_{k=1}^{K}\mathbf{G}_{(d_{\xi[t]})}^{k}[t] - \nabla_{\mathbf{A}_{(d_{\xi[t]})}}F(\mathbf{A}[t])\|_F^2\Big|\mathcal{F}[t],\xi[t]\Big]$$
$$= \mathbb{E}_{\zeta[t]}\Big[\|\frac{1}{K}\sum_{k=1}^{K}\mathbf{G}_{(d_{\xi[t]})}^{k}[t]\|_F^2\Big|\mathcal{F}[t],\xi[t]\Big] - \|\nabla_{\mathbf{A}_{(d_{\xi[t]})}}F(\mathbf{A}[t])\|_F^2. \qquad (7)$$

Taking conditional expectation on both sides of eq.(B.1.3) with respect to filtration $\mathcal{F}[t]$ and randomness of $\zeta[t]$ during the stochastic gradient computation and plugging eq.(7) in, we have

$$\mathbb{E}_{\zeta[t]}\Big[F(\widetilde{\mathbf{A}}[t+1])\Big|\mathcal{F}[t],\xi[t]\Big]$$
$$\leq F(\widetilde{\mathbf{A}}[t]) - \gamma[t]\Big(1 - \frac{L_{d_{\xi}[t]}\gamma[t]}{2}\Big)\|\nabla_{\mathbf{A}_{(d_{\xi[t]})}}F(\mathbf{A}[t])\|_F^2 + \frac{L_{d_{\xi}[t]}(\gamma[t])^2}{2K}\sigma_{d_{\xi}[t]}^2$$
$$+\gamma[t]\langle\nabla_{\mathbf{A}_{(d_{\xi[t]})}}F(\mathbf{A}[t]) - \nabla_{\mathbf{A}_{(d_{\xi[t]})}}F(\widetilde{\mathbf{A}}[t]), \nabla_{\mathbf{A}_{(d_{\xi[t]})}}F(\mathbf{A}[t])\rangle.$$

We bound $\langle \nabla_{\mathbf{A}_{(d_{\xi[t]})}} F(\mathbf{A}[t]) - \nabla_{\mathbf{A}_{(d_{\xi[t]})}} F(\widetilde{\mathbf{A}}[t]), \nabla_{\mathbf{A}_{(d_{\xi[t]})}} F(\mathbf{A}[t])\rangle$ by Young's inequality, we have

$$\begin{aligned}&\mathbb{E}_{\zeta[t]}\Big[F(\widetilde{\mathbf{A}}[t+1])\Big|\mathcal{F}[t],\xi[t]\Big]\\ &\leq F(\widetilde{\mathbf{A}}[t]) - \gamma[t]\Big(1-\frac{L_{d_\xi[t]}\gamma[t]+\rho}{2}\Big)\|\nabla_{\mathbf{A}_{(d_\xi[t])}}F(\mathbf{A}[t])\|_F^2\\ &+\frac{L_{d_\xi[t]}(\gamma[t])^2}{2K}\sigma^2_{d_\xi[t]}+\frac{L^2_{d_\xi[t]}\gamma[t]}{2\rho}\|\frac{1}{K}\sum_{k=1}^K \mathbf{E}^k_{(d_{\xi[t]})}[t]\|_F^2.\end{aligned}$$

Taking expectation with respect to $\xi[t]$ conditioned on $\mathcal{F}[t]$ and substituting $L=\max\{L_1,...,L_D\}$, $\sigma^2=\sum_{d=1}^D \sigma_d^2$ in, we have

$$\begin{aligned}&\mathbb{E}_{\xi[t]}\Big[F(\widetilde{\mathbf{A}}[t+1])\Big|\mathcal{F}[t]\Big]\\ &\leq F(\widetilde{\mathbf{A}}[t]) - \gamma[t]\Big(1-\frac{L\gamma[t]+\rho}{2}\Big)\frac{1}{D}\sum_{d=1}^D\|\nabla_{\mathbf{A}_{(d)}}F(\mathbf{A}[t])\|_F^2\\ &+\frac{L(\gamma[t]\sigma)^2}{2KD}+\frac{L^2\gamma[t]}{2\rho}\frac{1}{D}\sum_{d=1}^D\|\frac{1}{K}\sum_{k=1}^K\mathbf{E}^k_{(d)}[t]\|_F^2.\end{aligned}$$

By Lemma B.1 and let $\gamma[t]=t$, we have

$$\begin{aligned}&\mathbb{E}_{\xi[t]}\Big[F(\widetilde{\mathbf{A}}[t+1])\Big|\mathcal{F}[t]\Big]\\ &\leq F(\widetilde{\mathbf{A}}[t]) - \gamma\Big(1-\frac{L\gamma+\rho}{2}\Big)\frac{1}{D}\sum_{d=1}^D\|\nabla_{\mathbf{A}_{(d)}}F(\mathbf{A}[t])\|_F^2\\ &+\frac{L(\gamma\sigma)^2}{2KD}+\frac{2L^2\gamma^3(1-\delta)(\sigma^2+\omega^2)}{\rho D\delta^2}.\end{aligned} \quad (8)$$

Taking total expectation with respect to all the random variables in $\mathcal{F}[t]$, and averaging the above from $t=0$ to T and letting $\rho<2-L\gamma$, F^* the optimal value, we have we have

$$\begin{aligned}&\frac{1}{T+1}\sum_{t=0}^T\mathbb{E}\Big[\frac{1}{D}\sum_{d=1}^D\|\nabla_{\mathbf{A}_{(d)}}F(\mathbf{A}[t])\|_F^2\Big]\\ &\leq\frac{1}{(T+1)\gamma(1-\frac{L\gamma+\rho}{2})}\Big[F(\mathbf{A}[0])-F^*\Big]\\ &+\frac{1}{(1-\frac{L\gamma+\rho}{2})}\Big[\frac{L\gamma\sigma^2}{2KD}+\frac{2L^2\gamma^2(1-\delta)(\sigma^2+\omega^2)}{\rho D\delta^2}\Big].\end{aligned} \quad (9)$$

By setting $\rho=1$ and using $\mathbb{E}\Big[\frac{1}{D}\sum_{d=1}^D\|\nabla_{\mathbf{A}_{(d)}}F(\mathbf{A}[\mathtt{Output}])\|_F^2\Big]\leq\sum_{t=0}^T\frac{1}{T+1}\mathbb{E}\Big[\frac{1}{D}\sum_{d=1}^D\|\nabla_{\mathbf{A}_{(d)}}F(\mathbf{A}[t])\|_F^2\Big]$, letting $\gamma=\min\{\frac{1}{2L},\frac{\varrho}{\sqrt{T+1}/\sqrt{K}+\frac{(1-\delta)^{1/3}}{\delta^{2/3}}T^{1/3}}\}$ for some $\varrho>0$, we complete the proof of Theorem 4.1:

$$\begin{aligned}&\mathbb{E}[\frac{1}{D}\sum_{d=1}^D\|\nabla_{\mathbf{A}_{(d)}}F(\mathbf{A}[\mathtt{Output}])\|_F^2]\\ &\leq\frac{8L}{T+1}(F(\mathbf{A}[0])-F^*)+\Big[\frac{4}{\varrho}(F(\mathbf{A}[0])-F^*)+\frac{2L\sigma^2\varrho}{D}\Big]\frac{1}{\sqrt{M(T+1)}}\\ &+\Big[\frac{4}{\varrho}(F(\mathbf{A}[0])-F^*)+\frac{8L^2\varrho^2(\sigma^2+\omega^2)}{D}\Big]\frac{(1-\delta)^{1/3}}{\delta^{2/3}(T+1)^{2/3}}.\end{aligned}$$

B.2 Proof Sketch of Theorem 4.2

B.2.1 Auxiliary variables for the proof and iterative relation. We derive the convergence by regarding the iteration as using inexact gradient, which is different from the approach used for the smooth case which is regarded as using delayed variable:

$$\begin{aligned}&\mathbf{A}_{(d_\xi[t])}[t+1]=\mathtt{Prox}\Big(\mathbf{A}_{(d_\xi[t])}[t]-\frac{1}{K}\sum_{k=1}^K\Delta^k_{(d_\xi[t])}[t]\Big)=\\ &\mathtt{Prox}\Big(\mathbf{A}_{(d^k)}[t]-\gamma[t]\frac{1}{K}\sum_{k=1}^K(\mathbf{G}^k_{(d_\xi[t])}[t]\\ &+\frac{1}{\gamma[t]}(\mathbf{E}^k_{(d_\xi[t])}[t+1]-\mathbf{E}^k_{(d_\xi[t])}[t]))\Big).\end{aligned}$$

We define the generalized gradient $\mathbf{Z}[t]=(\mathbf{Z}_{(1)}[t],...,(\mathbf{Z}_{(D)}[t])$, where $\mathbf{Z}_{(d)}[t]=\frac{1}{\gamma[t]}\Big(\mathbf{A}_{(d)}[t]-\mathtt{Prox}_{r_{(d)}}(\mathbf{A}_{(d)}[t]-\gamma[t]\nabla_{\mathbf{A}_{(d)}}F(\mathbf{A}[t]))\Big)$ If $d=d_\xi[t]$, $\overline{\mathbf{A}}_{(d)}[t+1]=\mathtt{Prox}_{r_{(d)}}(\mathbf{A}_{(d)}[t]-\gamma[t]\nabla_{\mathbf{A}_{(d)}}F(\mathbf{A}[t]))$, else if $d\neq d_\xi[t]$ $\overline{\mathbf{A}}_{(d)}[t+1]=\mathbf{A}_{(d)}[t]$.

let $\Phi(\mathbf{A}[t])=F(\mathbf{A}[t])+r(\mathbf{A}[t])$.

B.2.2 Additional Lemma. We need Lemma 1 from [30].

Lemma B.2. *Let* $\mathbf{y}=\mathtt{Prox}_{\gamma r}(\mathbf{x}-\gamma\mathbf{g})$, *for some* $\mathbf{g}$. *Then for* $\mathbf{y}$, *the following inequality holds for any* $\mathbf{z}$,

$$r(\mathbf{y})+\langle\mathbf{y}-\mathbf{z},\mathbf{g}\rangle\leq r(\mathbf{z})+\frac{1}{2\gamma}[\|\mathbf{z}-\mathbf{x}\|_2^2-\|\mathbf{y}-\mathbf{x}\|_2^2-\|\mathbf{y}-\mathbf{z}\|_2^2]. \quad (10)$$

B.2.3 Main Proof sketch of Theorem 4.2. By the block-wise smoothness of F, the convexity of $r_{(d)}(\cdot)$, and the optimality of $\overline{\mathbf{A}}_{(d_\xi[t])}[t+1]$ for $\mathtt{Prox}_{r_{(d)}}(\mathbf{A}_{(d)}[t]-\gamma\nabla_{\mathbf{A}_{(d)}}F(\mathbf{A}[t]))$, we have

$$\Phi(\overline{\mathbf{A}}[t+1])\leq\Phi(\mathbf{A}[t])+(\frac{L_{(d_\xi[t])}}{2}-\frac{1}{\gamma[t]})\|\overline{\mathbf{A}}_{(d_\xi[t])}[t+1]-\mathbf{A}_{(d_\xi[t])}[t]\|_F^2. \quad (11)$$

By Lemma B.2, we have

$$\begin{aligned}&F(\mathbf{A}_{(d_\xi[t])}[t+1],\mathbf{A}_{(-d_\xi[t])}[t])+r_{(d_\xi[t])}(\mathbf{A}_{(d_\xi[t])}[t+1])\\ &\leq F(\overline{\mathbf{A}}_{(d_\xi[t])}[t+1],\mathbf{A}_{(-d_\xi[t])}[t])+r_{(d_\xi[t])}(\overline{\mathbf{A}}_{(d_\xi[t])}[t+1])\\ &+\langle\mathbf{A}_{(d_\xi[t])}[t+1]-\overline{\mathbf{A}}_{(d_\xi[t])}[t+1],\nabla_{(d_\xi[t])}F(\mathbf{A}_{(d_\xi[t])}[t])\\ &-\frac{1}{K}\sum_{k=1}^K(\mathbf{G}^k_{(d_\xi[t])}[t]+\frac{1}{\gamma[t]}(\mathbf{E}^k_{(d_\xi[t])}[t+1]-\mathbf{E}_{(d^k_\xi[t])}[t]))\rangle\\ &+(\frac{L_{(d_\xi[t])}}{2}-\frac{1}{2\gamma[t]})\|\mathbf{A}_{(d_\xi[t])}[t+1]-\mathbf{A}_{(d_\xi[t])}[t]\|_F^2\\ &+(\frac{L_{(d_\xi[t])}}{2}+\frac{1}{2\gamma[t]})\|\overline{\mathbf{A}}_{(d_\xi[t])}[t+1]-\mathbf{A}_{(d_\xi[t])}[t]\|_F^2\\ &-\frac{1}{2\gamma[t]}\|\overline{\mathbf{A}}_{(d_\xi[t])}[t+1]-\mathbf{A}_{(d_\xi[t])}[t+1]\|_F^2.\end{aligned}$$

By bounding the third row of the above equation, choosing $\rho_1=2\gamma[t]$ and $\rho_2=2$, with eq.(11), and letting $\gamma[t]\leq\frac{1}{2L_{(d_\xi[t])}}$, we have

$$\begin{aligned}&\Phi(\mathbf{A}[t+1])\leq\Phi(\mathbf{A}[t])+(L_{(d_\xi[t])}-\frac{1}{2\gamma[t]})\|\overline{\mathbf{A}}_{(d_\xi[t])}[t+1]-\mathbf{A}_{(d_\xi[t])}[t]\|_F^2\\ &+\gamma[t]\frac{1}{K}\sum_{k=1^K}\|\nabla_{(d_\xi[t])}F(\mathbf{A}_{(d_\xi[t])}[t])-\mathbf{G}^k_{(d_\xi[t])}[t]\|_F^2\\ &+\frac{1}{\gamma[t]}\frac{1}{K}\sum_{k=1}^K\|\mathbf{E}^k_{(d_\xi[t])}[t+1]-\mathbf{E}^k_{(d_\xi[t])}[t]\|_F^2.\end{aligned}$$

Taking conditional expectation with respect to $\xi[t]$ conditioned on filtration $\mathcal{F}[t]$, by Lemma B.1 and letting $\gamma[t] = t$, we have

$$\Phi(\mathbf{A}[t+1]) \leq \Phi(\mathbf{A}[t]) + (L - \frac{1}{2\gamma})\frac{1}{D}\sum_{d=1}^{D} \||\|\overline{\mathbf{A}}_{(d)}[t+1] - \mathbf{A}_{(d)}[t]\|_F^2$$

$$+ \frac{\gamma\sigma^2}{D} + \frac{1}{D}\frac{8(1-\delta)}{\delta^2}\gamma(\sigma^2 + \omega^2).$$

Taking total expectation (i.e. with respect to all random variables in $\mathcal{F}[t]$), averaging from $t = 0$ to T and using $\mathbb{E}\Big[\sum_{d=1}^{D}\frac{1}{D}\|\widetilde{\mathbf{G}}_{(d)}[\mathtt{Output}]\|_F^2\Big] \leq \frac{1}{T+1}\mathbb{E}[\sum_{d=1}^{D}\frac{1}{D}\|\widetilde{\mathbf{G}}_{(d)}[t]\|_F^2]$ $= \frac{1}{T+1}\mathbb{E}[\sum_{d=1}^{D}\frac{1}{D}\|(\overline{\mathbf{A}}_{(d)}[t+1] - \mathbf{A}_{(d)}[t])/\gamma\|_F^2]$, by setting $\gamma = \frac{1}{4L}$, we complete our proof:

$$\mathbb{E}\Big[\sum_{d=1}^{D}\frac{1}{D}\|\widetilde{\mathbf{G}}_{(d)}[\mathtt{Output}]\|_F^2\Big]$$

$$\leq \frac{16L}{T+1}(\Phi(\mathbf{A}[0]) - \Phi^*) + \frac{4\sigma^2}{DK} + \frac{32(1-\delta)}{D\delta^2}(\sigma^2 + \omega^2).$$

REFERENCES

[1] [n.d.]. https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/DE_Syn_PUF.

[2] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. 2018. cpSGD: Communication-efficient and differentially-private distributed SGD. In *Advances in Neural Information Processing Systems.*

[3] Brett W. Bader, Tamara G. Kolda, et al. 2017. MATLAB Tensor Toolbox Version 3.0-dev. Available online. https://gitlab.com/tensors/tensor_toolbox

[4] Debraj Basu, Deepesh Data, Can Karakus, and Suhas Diggavi. 2019. Qsparse-local-SGD: Distributed SGD with Quantization, Sparsification, and Local Computations. In *Advances in Neural Information Processing Systems.*

[5] Casey Battaglino, Grey Ballard, and Tamara G Kolda. 2018. A practical randomized CP tensor decomposition. *SIAM J. Matrix Anal. Appl.* 39, 2 (2018), 876–901.

[6] Alex Beutel, Partha Pratim Talukdar, Abhimanu Kumar, Christos Faloutsos, Evangelos E Papalexakis, and Eric P Xing. 2014. Flexifact: Scalable flexible factorization of coupled tensors on hadoop. In *Proceedings of the 2014 SIAM International Conference on Data Mining.* SIAM, 109–117.

[7] J Douglas Carroll and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika* 35, 3 (1970), 283–319.

[8] Tianyi Chen, Xiao Jin, Yuejiao Sun, and Wotao Yin. 2020. Vafl: a method of vertical asynchronous federated learning. *arXiv preprint arXiv:2007.06081* (2020).

[9] Joon Hee Choi and S Vishwanathan. 2014. DFacTo: Distributed factorization of tensors. In *Advances in Neural Information Processing Systems.* 1296–1304.

[10] Xiao Fu, Shahana Ibrahim, Hoi-To Wai, Cheng Gao, and Kejun Huang. 2020. Block-randomized stochastic proximal gradient for low-rank tensor factorization. *IEEE Transactions on Signal Processing* 68 (2020), 2170–2185.

[11] Richard A Harshman et al. 1970. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multimodal factor analysis. (1970).

[12] Huan He, Jette Henderson, and Joyce C Ho. 2019. Distributed Tensor Decomposition for Large Scale Health Analytics. In *The World Wide Web Conference.*

[13] Joyce C Ho, Joydeep Ghosh, Steve R Steinhubl, Walter F Stewart, Joshua C Denny, Bradley A Malin, and Jimeng Sun. 2014. Limestone: High-throughput candidate phenotype generation via tensor factorization. *Journal of biomedical informatics* 52 (2014), 199–211.

[14] Joyce C Ho, Joydeep Ghosh, and Jimeng Sun. 2014. Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* 115–124.

[15] David Hong, Tamara G Kolda, and Jed A Duersch. 2018. Generalized canonical polyadic tensor decomposition. *arXiv preprint arXiv:1808.07452* (2018).

[16] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3 (2016), 160035.

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

[18] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems.* 79–86.

[19] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. 2019. Error Feedback Fixes SignSGD and other Gradient Compression Schemes. In *International Conference on Machine Learning.* 3252–3261.

[20] Yejin Kim, Jimeng Sun, Hwanjo Yu, and Xiaoqian Jiang. 2017. Federated tensor factorization for computational phenotyping. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

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

[22] Tamara G Kolda and David Hong. 2019. Stochastic Gradients for Large-Scale Tensor Decomposition. *arXiv preprint arXiv:1906.01687* (2019).

[23] Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. 2018. Don't Use Large Mini-Batches, Use Local SGD. *arXiv preprint arXiv:1808.07217* (2018).

[24] Jian Lou and Yiu-ming Cheung. 2018. Uplink communication efficient differentially private sparse optimization with feature-wise distributed data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[25] Jian Lou and Yiu-ming Cheung. 2020. An Uplink Communication-Efficient Approach to Featurewise Distributed Sparse Optimization With Differential Privacy. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[26] Jing Ma, Qiuchen Zhang, Joyce C. Ho, and Li Xiong. 2020. Spatio-Temporal Tensor Sketching via Adaptive Sampling. *CoRR* abs/2006.11943 (2020). arXiv:2006.11943 https://arxiv.org/abs/2006.11943

[27] Jing Ma, Qiuchen Zhang, Jian Lou, Joyce C Ho, Li Xiong, and Xiaoqian Jiang. 2019. Privacy-preserving tensor factorization for collaborative health data analysis. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.* 1291–1300.

[28] Ioakeim Perros, Evangelos E Papalexakis, Fei Wang, Richard Vuduc, Elizabeth Searles, Michael Thompson, and Jimeng Sun. 2017. SPARTan: Scalable PARAFAC2 for large & sparse data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 375–384.

[29] Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. 2013. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization* 23, 2 (2013), 1126–1153.

[30] Sashank J Reddi, Suvrit Sra, Barnabas Poczos, and Alexander J Smola. 2016. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. In *Advances in Neural Information Processing Systems.* 1145–1153.

[31] Kijung Shin, Lee Sael, and U Kang. 2016. Fully scalable methods for distributed tensor factorization. *IEEE Transactions on Knowledge and Data Engineering* 29, 1 (2016), 100–113.

[32] Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos. 2017. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing* 65, 13 (2017), 3551–3582.

[33] Sebastian U Stich. 2018. Local SGD Converges Fast and Communicates Little. In *International Conference on Learning Representations.*

[34] Sebastian U Stich and Sai Praneeth Karimireddy. 2019. The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350* (2019).

[35] M Alex O Vasilescu and Demetri Terzopoulos. 2002. Multilinear analysis of image ensembles: Tensorfaces. In *European conference on computer vision.* Springer.

[36] Yichen Wang, Robert Chen, Joydeep Ghosh, Joshua C Denny, Abel Kho, You Chen, Bradley A Malin, and Jimeng Sun. 2015. Rubik: Knowledge guided tensor factorization and completion for health data analytics. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

[37] Chen Xu, Jianqiang Yao, Zhouchen Lin, Wenwu Ou, Yuanbin Cao, Zhirong Wang, and Hongbin Zha. 2018. Alternating multi-bit quantization for recurrent neural networks. *ICLR-2018, arXiv preprint arXiv:1802.00150* (2018).

[38] Yangyang Xu and Wotao Yin. 2015. Block stochastic gradient iteration for convex and nonconvex optimization. *SIAM Journal on Optimization* (2015).

[39] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[40] Jinshan Zeng, Tim Tsz-Kit Lau, Shaobo Lin, and Yuan Yao. 2019. Global Convergence of Block Coordinate Descent in Deep Learning. In *International Conference on Machine Learning.* 7313–7323.

[41] Shandian Zhe, Kai Zhang, Pengyuan Wang, Kuang-chih Lee, Zenglin Xu, Yuan Qi, and Zoubin Ghahramani. 2016. Distributed flexible nonlinear tensor factorization. In *Advances in neural information processing systems.* 928–936.

[42] Shuai Zheng, Ziyue Huang, and James T Kwok. 2019. Communication-Efficient Distributed Blockwise Momentum SGD with Error-Feedback. In *Advances in Neural Information Processing Systems.*