

# Towards Efficient Communication and Secure Federated Recommendation System via Low-rank Training

Ngoc-Hieu Nguyen  
ngochieutb13@gmail.com  
College of Engineering & Computer  
Science, VinUniversity  
Hanoi, Vietnam

Tuan-Anh Nguyen  
21anh.nt@vinuni.edu.vn  
College of Engineering & Computer  
Science, VinUniversity  
Hanoi, Vietnam

Tuan Nguyen  
tuan.nm@vinuni.edu.vn  
College of Engineering & Computer  
Science, VinUniversity  
Hanoi, Vietnam

Vu Tien Hoang  
vu.ht@vinuni.edu.vn  
College of Engineering & Computer  
Science, VinUniversity  
Hanoi, Vietnam

Dung D. Le\*  
dung.ld@vinuni.edu.vn  
College of Engineering & Computer  
Science, VinUniversity  
Hanoi, Vietnam

Kok-Seng Wong\*  
wong.ks@vinuni.edu.vn  
College of Engineering & Computer  
Science, VinUniversity  
Hanoi, Vietnam

## ABSTRACT

Federated Recommendation (FedRec) systems have emerged as a solution to safeguard users' data in response to growing regulatory concerns. However, one of the major challenges in these systems lies in the communication costs that arise from the need to transmit neural network models between user devices and a central server. Prior approaches to these challenges often lead to issues such as computational overheads, model specificity constraints, and compatibility issues with secure aggregation protocols. In response, we propose a novel framework, called Correlated Low-rank Structure (CoLR), which leverages the concept of adjusting lightweight trainable parameters while keeping most parameters frozen. Our approach substantially reduces communication overheads without introducing additional computational burdens. Critically, our framework remains fully compatible with secure aggregation protocols, including the robust use of Homomorphic Encryption. The approach resulted in a reduction of up to 93.75% in payload size, with only an approximate 8% decrease in recommendation performance across datasets. Code for reproducing our experiments can be found at <https://github.com/NNHieu/CoLR-FedRec>.

## CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; • **Security and privacy** → **Privacy protections**.

## KEYWORDS

Recommendation System, Federated Learning, Communication efficiency

\*Co-last author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

WWW '24, May 13–17, 2024, Singapore, Singapore

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0171-9/24/05

<https://doi.org/10.1145/3589334.3645702>

## ACM Reference Format:

Ngoc-Hieu Nguyen, Tuan-Anh Nguyen, Tuan Nguyen, Vu Tien Hoang, Dung D. Le, and Kok-Seng Wong. 2024. Towards Efficient Communication and Secure Federated Recommendation System via Low-rank Training. In *Proceedings of the ACM Web Conference 2024 (WWW '24)*, May 13–17, 2024, Singapore, Singapore. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3589334.3645702>

## 1 INTRODUCTION

In a centralized recommendation system, all user behavior data is collected on a central server for training. However, this method can potentially expose private information that users may be hesitant to share with others. As a result, various regulations such as the General Data Protection Regulation (GDPR)[35] and the California Consumer Privacy Act (CCPA)[31] have been implemented to limit the centralized collection of users' personal data. In response to this challenge, and in light of the increasing prevalence of edge devices, federated recommendation (FedRec) systems have gained significant attention for their ability to uphold user privacy [2, 5, 13, 23–25, 33, 37–39, 41].

The training of FedRec systems is often in a cross-device setting which involves transferring recommendation models between a central server and numerous edge devices, such as mobile phones, laptops, and PCs. It is increasingly challenging to transfer these models due to the growing model complexity and parameters in modern recommendation systems [22, 29, 42]. In addition, clients participating in FedRec systems often exhibit differences in their computational processing speeds and communication bandwidth capabilities, primarily stemming from variations in their hardware and infrastructure [19]. These discrepancies can give rise to stragglers and decrease the number of participants involved in training, potentially leading to diminished system performance.

Practical FedRec systems require the implementation of mechanisms that reduce the amount of communication costs. Three commonly used approaches to reduce communication costs include (i) reducing the frequency of communication by allowing local updates, (ii) minimizing the size of the message through message compression, and (iii) reducing the server-side communication traffic by restricting the number of participating clients per round

[36]. Importantly, these three methods are independent and can be combined for enhanced efficiency.

In this study, we address the challenge of communication efficiency in federated recommendations by introducing an alternative to compression methods. Many existing compression methods involve encoding and decoding steps that can introduce significant delays, potentially outweighing the gains achieved in per-bit communication time [34]. Another crucial consideration is the compatibility with aggregation protocols. For example, compression techniques that do not align with all-reduce aggregation may yield reduced communication efficiency in systems employing these aggregation techniques [34]. This is also necessary for many secure aggregation protocols such as Homomorphic Encryption (HE) [1, 4]. Moreover, many algorithms assume that clients have the same computational power, but this may induce stragglers due to computational heterogeneity and can increase the runtime of algorithms.

Based on our observation that the update transferred between clients and the central server in FedRec systems has a low-rank bias (Section 4.1), we propose Correlated Low-rank Structure update (CoLR). CoLR increases communication efficiency by adjusting lightweight trainable parameters while keeping most parameters frozen. Under this training scheme, only a small amount of trainable parameters will be shared between the server and clients. Compared with other compression techniques, our methods offer the following benefits. (i) **Reduce both up-link and down-link communication cost:** CoLR avoid the need of unrolling the low-rank message in the aggregation step by using a correlated projection, (ii) **Low computational overheads:** Our method enforces a low-rank structure in the local update during the local optimization stage so eliminates the need to perform a compression step. Moreover, CoLR can be integrated into common aggregation methods such as FedAvg and does not require additional computation. (iii) **Compatible with secure aggregation protocols:** the aggregation step on CoLR can be carried by simple additive operations, this simplicity makes it compatible with strong secure aggregation methods such as HE, (iv) **Bandwidth heterogeneity awareness:** Allowing adaptive rank for clients based on computational/communication budget. Our framework demonstrates a capability to provide a strong foundation for building a secure and practical recommendation system.

Our contributions can be summarized as following:

- We propose a novel framework, CoLR, designed to tackle the communication challenge in training FedRec systems.
- We conducted experiments to showcase the effectiveness of CoLR. Notably, even with an update size equates to 6.25% of the baseline model, CoLR demonstrates remarkable efficiency by retaining 93.65% accuracy (in terms of HR) compared to the much larger baseline.
- We show that CoLR is compatible with HE-based FedRec systems and, hence, reinforces the security of the overall recommendation systems.

## 2 RELATED WORK

*Federated Recommendation (FedRec) Systems.* In recent years, FedRec systems have risen to prominence as a key area of research in both machine learning and recommendation systems. FCF [2]

and FedRec [23] are the pioneering FL-based methods for collaborative filtering based on matrix factorization. The former is designed for implicit feedback, while the latter is for explicit feedback. To enhance user privacy, FedMF [5] applies distributed matrix factorization within the FL framework and introduces the HE technique for securing gradients before they are transmitted to the server. MetaMF [24] is a distributed matrix factorization framework using a meta-network to generate rating prediction models and private item embedding. [39] presents FedPerGNN, where each user maintains a GNN model to incorporate high-order user-item information. FedNCF [33] adapts Neural Collaborative Filtering (NCF) [12] to the federated setting, incorporating neural networks to learn user-item interaction functions and thus enhancing the model's learning capabilities.

*Communication Efficient Federated Recommendation.* Communication efficiency is of the utmost importance in FL [17]. JointRec [7] reduces uplink costs in FedRS by using low-rank matrix factorization and 8-bit probabilistic quantization to compress weight updates. Some works explore reducing the entire item latent matrix payload by meta-learning techniques [24, 38]. LightFR [44] proposes a framework to reduce communication costs by exploiting the learning-to-hash technique under federated settings and enjoys both fast online inference and reduced memory consumption. Another solution is proposed by Khan et al. [16], which is a multi-arm bandit algorithm to address item-dependent payloads.

*Low-rank Structured Update.* Konečný et al. [17] propose to enforce every update to local model  $\Delta_u$  to have a low rank structure by express  $\Delta_u = A_u^{(t)} B_u^{(t)}$  where  $A_u^{(t)} \in \mathbb{R}^{d_1 \times k}$  and  $B_u^{(t)} \in \mathbb{R}^{k \times d_2}$ . In subsequent computation,  $A_u^{(t)}$  is generated independently for each client and frozen during local training procedures. This approach saves a factor of  $d_1/k$ . Hyeon-Woo et al. [14] proposes a method that re-parameterizes weight parameters of layers using low-rank weights followed by the Hadamard product. The authors show that FedPara can achieve comparable performance to the original model with 3 to 10 times lower communication costs on various tasks, such as image classification, and natural language processing.

*Secure FedRec.* Sending updates directly to the server without implementing privacy-preserving mechanisms can lead to security vulnerabilities. Chai et al. [5] demonstrated that in the case of the Matrix Factorization (MF) model using the FedAvg learning algorithm, if adversaries gain access to a user's gradients in two consecutive steps, they can deduce the user's rating information. One approach involves leveraging HE to encrypt intermediate parameters before transmitting them to the server [5, 32]. This method effectively safeguards user ratings while maintaining recommendation accuracy. However, it introduces significant computational overhead, including encryption and decryption steps on the client side, as well as aggregation on the server side. Approximately 95% of the time consumed by the system is dedicated to operations carried out on the ciphertext [5]. Liang et al. [21] aim to enhance the performance of FedRec systems using denoising clients. Liu et al. [26] discuss the development of secure recommendation systems in cross-domain settings. Recent studies [40, 43, 45] show that FedRecs are susceptible to poisoning attacks of malicious clients.

### 3 PRELIMINARIES

In this section, we present the preliminaries and the setting that the paper is working with. Also, this part will discuss the challenges in applying compression methods.

#### 3.1 Federated Learning for Recommendation

In the typical settings of item-based FedRec systems [23], there are  $M$  users and  $N$  items where each user  $u$  has a private interaction set denoted as  $O_u = \{(i, r_{ui})\} \subset [N] \times \mathbb{R}$ . These users want to jointly build a recommendation system based on local computations without violating participants' privacy. This scenario naturally aligns with the horizontal federated setting [28], as it allows us to treat each user as an active participant. In this work, we also use the terms user and client interchangeably. The primary goal of such a system is to generate a ranked list of top-K items that a given user has not interacted with and are relevant to the user's preferences. Mathematically, we can formalize the problem as finding a global model parameterized by  $\theta$  that minimizes the following global loss function  $\mathcal{L}(\cdot)$ :

$$\mathcal{L}(\theta) \triangleq \sum_{u=1}^M w_u \mathcal{L}_u(\theta) \quad (1)$$

where  $\theta$  is the global parameter,  $w_u$  is the relative weight of user  $u$ . And  $\mathcal{L}_u(\theta) := \sum_{(i, r_{ui}) \in O_u} \ell_u(\theta, (i, r_{ui}))$  is the local loss function at user  $u$ 's device. Here  $(i, r_{ui})$  represents a data sample from the user's private dataset, and  $\ell_u$  is the loss function defined by the learning algorithm. Setting  $w_u = N_u/N$  where  $N_u = |O_u|$  and  $N = \sum_{u=1}^M N_u$  makes the objective function  $\mathcal{L}(\theta)$  equivalent to the empirical risk minimization objective function of the union of all the users' dataset. Once the global model is learned, it can be used for user prediction tasks.

In terms of learning algorithms, Federated Averaging (FedAvg) [28] is one of the most popular algorithms in FL. FedAvg divides the training process into rounds. At the beginning of the  $t$ -th round ( $t \geq 0$ ), the server broadcasts the current global model  $\theta^{(t)}$  to a subset of users  $\mathcal{S}^{(t)}$  which is often uniformly sampled without replacement in simulation [23, 36]. Then each sampled client in the round's cohort performs  $\tau_u$  local SGD updates on its local dataset and sends the local model changes  $\Delta_u^{(t)} = \theta_u^{(t, \tau_u)} - \theta^{(t)}$  to the server. Finally, the server performs an aggregation step to update the global model:

$$\theta^{(t+1)} = \theta^{(t)} + \frac{\sum_{u \in \mathcal{S}^{(t)}} w_u \Delta_u^{(t)}}{\sum_{u \in \mathcal{S}^{(t)}} w_u} \quad (2)$$

The above procedure will repeat until the algorithm converges.

#### 3.2 Limitation of current compression methods

Communication is one of the main bottlenecks in FedRec systems and can be a serious constraint for both servers and clients. Although diverse optimization techniques exist to enhance communication efficiency, such methods may not preserve privacy. Moreover, tackling privacy and communication efficiency as separate concerns can result in suboptimal solutions.

*Top-K compression.* This method is based on sparsification, which represents updates as sparse matrices to reduce the transfer size.

However, the process of allocating memory for copying the gradient (which can grow to a large size, often in the millions) and then sorting this copied data to identify the top-K threshold during each iteration is costly enough that it negates any potential enhancements in overall training time when applied to real-world systems. As a result, employing these gradient compression methods in their simplest form does not yield the expected improvements in training efficiency. As observed in Gupta et al. [10], employing the Top-K compression for training large-scale recommendation models takes 11% more time than the baseline with no compression.

*SVD compression.* This method returns a compressed update with a low-rank structure, which is based on singular value decomposition. After obtaining factorization results  $U_u$  and  $V_u$ , the aggregation step requires performing decompression and computing  $\sum_{u \in \mathcal{S}} \frac{N_u}{N} U_u V_u$  and this sum is not necessarily low-rank so there is no readily reducing cost in the downlink communication without additional compression-decompression step. The need to perform matrix multiplication makes this method incompatible with HE. Moreover, performing SVD decomposition on an encrypted matrix by known schemes remains an open problem.

### 4 PROPOSED METHOD

#### 4.1 Motivation

Our method is motivated by analyzing the optimization process at each user's local device. We consider an effective federated matrix factorization (FedMF) as the backbone model. This model represents each item and user by a vector with the size of  $d$  denoted  $\mathbf{q}_i$  and  $\mathbf{p}_u$  respectively. And the predicted ratings  $r_{ui}$  are given by  $\hat{r}_{ui} = \mathbf{q}_i^\top \mathbf{p}_u$ . Then the user-wise local parameter  $\theta_u$  consists of the user  $u$ 's embedding  $\mathbf{p}_u$  and the item embedding matrix  $Q$ , where  $\mathbf{q}_i$  is the  $i$ th column of  $Q$ . The loss function  $\mathcal{L}_u$  at user  $u$ 's device is given in the following.

$$\mathcal{L}_u(\mathbf{p}_u, Q) = \sum_{(i, r_{ui}) \in O_u} \ell(r_{ui}, (Q^\top \mathbf{p}_u)_i) + \frac{\lambda}{2} \|\mathbf{p}_u\|_2^2 + \frac{\lambda}{2} \|Q\|_2^2$$

Let  $\eta$  be the learning rate, the update on the user embedding  $\mathbf{p}_u$  at each local optimization step is given by:

$$\mathbf{p}_u^{(t+1)} = \mathbf{p}_u^{(t)} (1 - \eta\lambda) + \eta Q^{(t)\top} (\mathbf{r} - \hat{\mathbf{r}}^{(t)}). \quad (3)$$

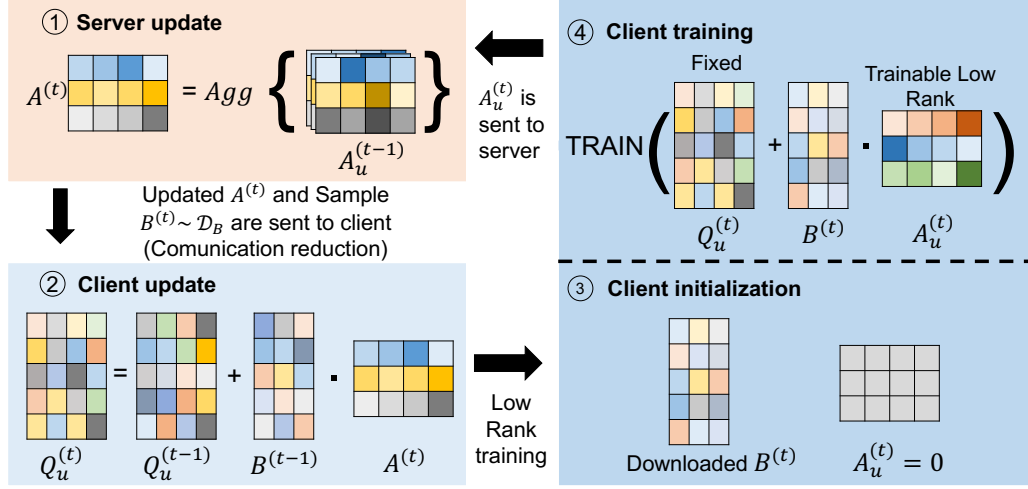
Let  $\mathbf{m} \in \mathbb{R}^N$  be a binary vector where  $\mathbf{m}_i = 1$  if  $i \in O_u$ , then the item embedding matrix  $Q$  is updated as follows:

$$Q^{(t+1)} = Q^{(t)} - \eta(\lambda Q^{(t)} - (\mathbf{m} * (\mathbf{r}_u - \hat{\mathbf{r}}_u)) \mathbf{p}_u^{(t)\top}) \quad (4)$$

The update that is sent to the central server has the following formula,

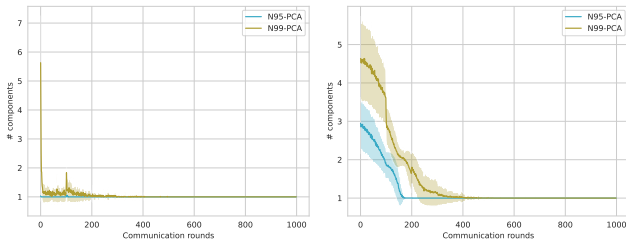
$$\Delta_Q^{(t)} = Q^{(t+1)} - Q^{(t)} = \eta \left[ (\mathbf{m} * (\mathbf{r}_u - \hat{\mathbf{r}}_u)) \mathbf{p}_u^{(t)\top} - \lambda Q^{(t)} \right] \quad (5)$$

As we can see from equation 4, since each client only stores the presentation of only one user  $\mathbf{p}_u$ , the update on the item embedding matrix at each local step is the sum of a rank-1 matrix and a regularization component. Given that  $\lambda$  is typically small, the low-rank component contributes most to the update  $\Delta_Q^{(t)}$ . And if the direction of  $\mathbf{p}_u$  does not change much during the local optimization phase, the update  $\Delta_Q^{(t)}$  can stay low-rank. From this observation, we first



**Figure 1: Illustration of CoLR at training round  $t$ .** At first, the server conducts aggregation over the local model  $A_u^{(t-1, \tau_u)}$  to obtain the global model update  $A^{(t)}$ . Subsequently,  $A^{(t)}$  are transmitted to the clients. The client will update their  $Q_u^{(t)}$  using this  $A^{(t)}$ , then initializes a new matrix  $A_u^{(t,0)}$  and download the matrix  $B^{(t)}$  which is sampled at the server and shared between clients. Finally, the client carries out local training and then sends the local model update  $A_u^{(t, \tau_u)}$  to the server for the next training round.

assume that the update of the item embedding matrix in training FedRec systems  $\Delta_Q^{(t)}$  can be well approximated by a low-rank matrix. We empirically verify this assumption by monitoring the effective rank of  $\Delta_Q^{(t)}$  at each training round for different datasets. The result is plotted in figure 2 where we plot the mean and standard deviation averaged over a set of participants in each round.



**Figure 2: PCA components progression.** The figures show the number of components that account for 99% (N99-PCA in green) and 95% (N95-PCA in blue) explained variance of all transfer item embedding matrix across communication rounds on the MovieLens-1M (left) and Pinterest (right) datasets.

This analysis suggests that reducing the communication by restricting the update to be low-rank might not sacrifice performance significantly. In the next section, we propose an efficient communication framework based on this motivation. Since most of the transferred parameters in recommendation models are from the item embedding layers, we will focus on applying the proposed method for embedding layers in this work.

## 4.2 Low-rank Structure

We propose explicitly enforcing a low-rank structure on the local update of the item embedding matrix  $Q$ . In particular, the local update  $(\Delta_Q)_u^{(t)}$  is parameterized by a matrix product

$$(\Delta_Q)_u^{(t)} = B_u^{(t)} A_u^{(t)}$$

where  $B_u^{(t)} \in \mathbb{R}^{d \times r}$  and  $A_u^{(t)} \in \mathbb{R}^{r \times N}$ . Given this parameterization, the embedding  $\mathbf{q}_i$  of an item with index  $i$  is given by

$$\mathbf{q}_i^{(t)} = \left( Q^{(t)} + B_u^{(t)} A_u^{(t)} \right) \mathbf{e}_i$$

where  $\mathbf{e}_i \in \mathbb{R}^N$  is a one-hot vector whose value at  $i$ -th is 1. This approach effectively saves a factor of  $\frac{N \times d}{N \times r + d \times r}$  in communication since clients only need to send the much smaller matrices  $A_u$  and  $B_u$  to the central server.

## 4.3 Correlated Low-rank Structure Update

Even though enforcing a low-rank structure on the update can greatly reduce the uplink communication size, doing aggregation and performing privacy-preserving is not trivial and faces the following three challenges: (1) the server needs to multiply out all the pairs  $A_u^{(t)}$  and  $B_u^{(t)}$  before performing the aggregation step; (2) the sum of low-rank solutions would typically leads to a larger rank update so there is no reducing footprint in the downlink communication; (3) secure aggregation method such as HE cannot directly apply to  $A_u^{(t)}$  and  $B_u^{(t)}$  since it will require to perform the multiplication between two encrypted matrices, which is much more costly than simple additive operation.

To reduce the downlink communication cost, we observe that if either  $A_u^{(t)}$  or  $B_u^{(t)}$  is identical between users and is fixed during the local training process, then the result of the aggregation step can be

represented by a low-rank matrix with the following formulation:

$$\Delta_Q^{(t)} = B^{(t)} \left( \sum_{u \in S} A_u^{(t)} \right).$$

Notice that this aggregation is also compatible with HE since it only requires additive operations on a set of  $A_u^{(t)}$  and clients can decrypt this result and then compute the global update  $\Delta_Q^{(t)}$  at their local device.

Based on the above observation, we propose the Correlated Low-rank Structure Update (CoLR) framework. In this framework, the server randomly initializes a matrix  $B^{(t)}$  at the beginning of each training round and shares it among all participants. Participants then set  $B_u^{(t)} = B^{(t)}$  and freeze this matrix during the local training phase and only optimize for  $A_u^{(t)}$ . The framework is presented in Algorithm 1 and illustrated in Figure 1. Note that the communication cost can be further reduced by sending only the random seed of the matrix  $B^{(t)}$ . A concurrent work [3] proposes FFA-LoRA which also fixes the randomly initialized non-zero matrices and only finetunes the zero-initialized matrices. They study FFA-LoRA in the context of federated fine-tuning LLMs and using differential privacy [8] to provide privacy guarantees.

---

**Algorithm 1:** Correlated Low-rank Structure Update Matrix Factorization

---

**Input:** Initial model  $Q^{(0)}$ ; update rank  $r$ , a distribution  $\mathcal{D}_B$  for initializing  $B$ ; CLIENTOPT, SERVEROPT with learning rates  $\eta, \eta_s$ ;

```

1 for  $t \in \{0, 1, 2, \dots, T\}$  do
2   Sample a subset  $S^{(t)}$  of clients
3   Sample  $B^{(t)} \sim \mathcal{D}_B$ 
4   for client  $u \in S^{(t)}$  in parallel do
5     if  $t > 0$  then
6       Download  $A^{(t)}$ 
7       Merge  $Q_u^{(t)} = Q^{(t-1)} + B^{(t-1)} A^{(t)}$ 
8     end
9     Initialize  $Q_u^{(t,0)} = Q^{(t)}$ 
10    Download  $B^{(t)}$  and Initialize  $A_u^{(t,0)} = \mathbf{0}$ 
11    Set trainable parameters  $\theta_u^{(t,0)} = \{A_u^{(t,0)}, \mathbf{p}_u^{(t,0)}\}$ 
12    for  $k = 0, \dots, \tau_u - 1$  do
13      Perform local update  $\theta_u^{(t,k+1)} =$ 
14        CLIENTOPT  $\left( \theta_u^{(t,k)}, \nabla \mathcal{L}_u \left( \theta_u^{(t,k)} \right), \eta \right)$ 
15    end
16     $\mathbf{p}_u^{(t+1)} = \mathbf{p}_u^{(t, \tau_u)}$ 
17    Upload  $\{A_u^{(t, \tau_u)}\}$  to the central server
18  end
19 end

```

---

$$A^{(t+1)} = \sum_{u \in S^{(t)}} \frac{N_u}{N} A_u^{(t, \tau_u)};$$

*Differences w.r.t. SVD compression.* We compare our method with SVD since it also uses a low-rank structure. The difference is that in CoLR, participants directly optimize these models on the low-rank parameterization, while SVD only compresses the result from the local training step.

#### 4.4 Subsampling Correlated Low-rank Structure update (SCoLR)

In this section, we consider scenarios where edge devices establish communication with a central server using network connections that vary in quality. We propose a variant of CoLR termed Subsampling Correlated Low-rank Structure update (SCoLR) which allows each device to choose a unique local rank, denoted as  $r_u$ , aligning with their specific computational capacities and individual preferences throughout the training process.

Let us denote  $r_g$  as the rank of global update, which is sent from the server to participants through downlink connections, and  $r_u$  as the rank of local update, which is sent from clients to the central server for aggregation through uplink connections. In implementation, we set  $r_g$  to be larger than  $r_u$ , reflecting that downlink bandwidth is often higher than uplink. Given these rank parameters, at the start of each training round, the central server first initializes a matrix  $B$  with the shape of  $\mathbb{R}^{d \times r_g}$ . Then, participants in that round will download this matrix to their local devices and select a subset of columns of  $B$  to perform the local optimization step. In particular, we demonstrate this process through the following formulation:

$$(\Delta_Q)_u^{(t)} = B S_u A_u, \quad (6)$$

where  $B$  is a matrix with the shape of  $\mathbb{R}^{d \times r_g}$  and  $A_u$  is a matrix with the shape of  $\mathbb{R}^{r_u \times N}$ . Specifically,  $S_u$  is a binary matrix with  $r_u$  rows and  $r_g$  columns, where each row has exactly one non-zero element. The non-zero element in the  $i$ -th row is at the  $j$ -th column, where  $j$  is the  $i$ -th element of a randomly shuffled array of integers from 1 to  $r_g$ . The detail is presented in Algorithm 2. Importantly, sharing the matrix  $S_u$  does not divulge sensitive user information. Multiplying this matrix with  $S_u$  is essentially a row reordering operation on the matrix  $A_u$ . As a result, we can effectively perform additive HE between pairs of rows from  $A_{u_1}$  and  $A_{u_2}$ . This approach ensures privacy while accommodating varying network connections' quality among clients.

## 5 EXPERIMENTS

### 5.1 Experimental Setup

**Table 1: Statistics of the datasets used in evaluation.**

Datasets	# Users	# Items	# Ratings	Data Density
MovieLens-1M [11]	6,040	3,706	1,000,209	4.47%
Pinterest [9]	55,187	9,916	1,500,809	0.27%

*Datasets.* We experiment with two publicly available datasets, which are MovieLens-1M [11] and Pinterest [9]. The statistics of these datasets are summarized in Table 1. We follow common practice in recommendation systems for preprocessing by retaining

users with at least 20 interactions and converting numerical ratings into implicit feedback [2, 12].

**Evaluation Protocols.** We employ the standard leave-one-out evaluation to set up our test set [12]. For each user, we use all their interactions for training while holding out their last interaction for testing. During the testing phase, we randomly sampled 99 non-interacted items for each user and ranked the test item amongst these sampled items.

To evaluate the performance and verify the effectiveness of our model, we utilize two evaluation metrics, i.e., Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG), which are widely adopted for item ranking tasks. The above two metrics are usually truncated at a particular rank level (e.g. the first  $k$  ranked items) to emphasize the importance of the first retrieved items. Intuitively, the HR metric measures whether the test item is present on the top- $k$  ranked list, and the NDCG metric measures the ranking quality, which comprehensively considers both the positions of ratings and the ranking precision.

**Models and Optimization.** For the base models, we adopt Matrix Factorization with the FedAvg learning algorithm, also used in Chai et al. [5]. In our experiments, the dimension of user and item embedding  $d$  is set to 64 for the MovieLens-1M dataset and 16 for the Pinterest dataset. This is based on our observation that increasing the embedding size on the Pinterest dataset leads to overfitting and decreased performance on the test set. This observation is also consistent with He et al. [12]. We use the simple SGD optimizer for local training at edge devices.

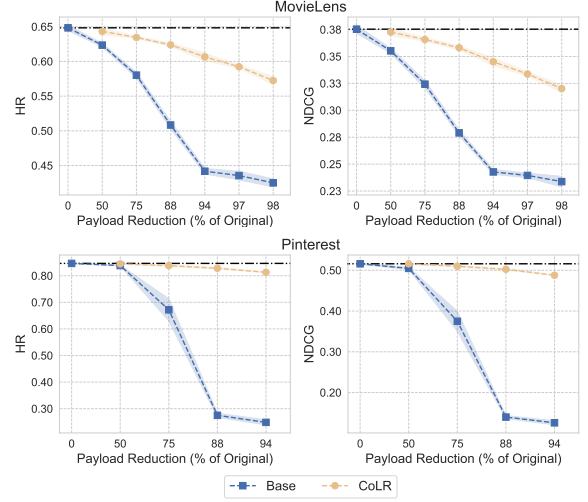
**Federated settings.** In each round, we sample  $M$  clients uniformly randomly, without replacement in a given round and across rounds. Instead of performing  $\tau_i$  steps of ClientOpt, we perform  $E$  epochs of training over each client's dataset. This is done because, in practical settings, clients have heterogeneous datasets of varying sizes. Thus, specifying a fixed number of steps can cause some clients to repeatedly train on the same examples, while certain clients only see a small fraction of their data.

**Baselines.** We have conducted a comparison between our framework and the basic FedMF models, along with two compression methods: SVD and Top-K compression. The first method, which is SVD-based, returns a compressed update with a low-rank structure. The second method is based on sparsification, representing updates as sparse matrices to reduce the transfer size.

**Machine.** The experiments were conducted on a machine equipped with an Intel(R) Xeon(R) W-1250 CPU @ 3.30GHz and a Quadro RTX 4000 GPU.

## 5.2 Experimental Results

**(1) CoLR can achieve comparable performances with the base models.** Given our primary focus is on recommendation performance within communication-limited environments, we commence our investigation by comparing the recommendation performance between CoLR and the base model FedMF given the same communication budget. On the ML-1M dataset, we adjust the dimensions of user and item embeddings across the set [1, 2, 4, 8, 16, 32, 64] for FedMF while fixing the embedding size of CoLR to 64,



**Figure 3: Performance on the MovieLens-1M dataset (Top) and the Pinterest dataset (Bottom).** We plot the utilities (HR and NDCG) versus the payload reduction and compare CoLR with the base model with the same transfer size. Each point represents the average recommendation performance on the test set across five random seeds. The shaded areas denote the standard deviation over the mean. The dashed black line presents the largest base model's performance.

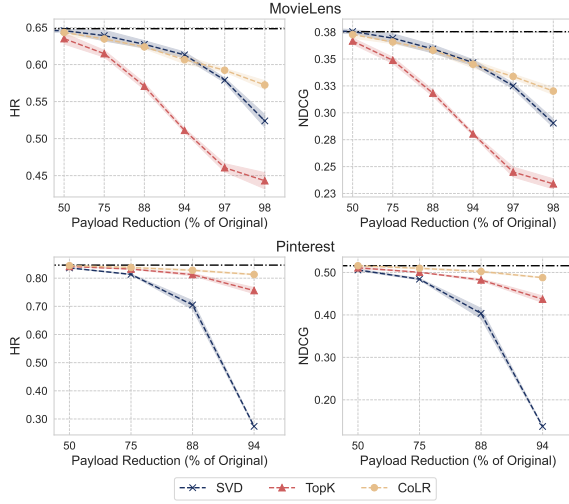
with different rank settings within [1, 2, 4, 8, 16, 32]. Similarly, for Pinterest, the embedding range for FedMF is [1, 2, 4, 8, 16], while CoLR has an embedding size of 16 and ranks in the range of [1, 2, 4, 8]. Our settings lead to approximately equivalent transfer sizes for both methods in each dataset.

In Figure 3, we present the HR and NDCG metrics across different transfer sizes. With equal communication budgets, CoLR consistently outperforms the base models on both datasets. To illustrate, on the Pinterest dataset, even with an update size equates to 6.25% of the largest model, CoLR achieves a notable performance (81.03% HR and 48.50% NDCG) compared to the base model (84.74% HR and 51.79% NDCG) while attaining a much larger reduction in terms of communication cost (16x). In contrast, the FedMF models with corresponding embedding sizes achieve much lower accuracies. On the MovieLens-1M dataset, we also observe a similar pattern where CoLR consistently demonstrates higher recommendation performance when compared to their counterparts.

The result from this experiment highlights that CoLR can achieve competitive performance when compared to the fully-trained model, FedMF while greatly reducing the cost of communication.

**(2) Comparison between CoLR and other compression-based methods.** We conducted the above experiment employing two compression methods, SVD and top-K compression, with compression ratios matched to those of CoLR. To ensure a fair evaluation, we applied the same compression ratio to both upload and download messages. The outcomes, depicted in Figure 4, reveal that CoLR consistently achieves favorable performance while outperforming other methods in scenarios with limited communication





**Figure 4: HR and NDCG on MovieLens-1M dataset (Top) and Pinterest Dataset (Bottom). We plot the utilities versus the payload reduction and compare CoLR with other methods with the same payload reduction. The dashed black line presents the base model’s performance.**

budgets. Notably, the performance of SVD and top-K compression varies across datasets. While SVD demonstrates favorable results with the MovieLens dataset, its performance substantially deteriorates with the Pinterest dataset.

**Table 2: Communication and training times for MovieLens-1M dataset, measured in minutes.**

Method	Communication time (mins)	Computation time (mins)	Total Training Time (mins)
MF-64	80.43	169.07	249.50
CoLR@1	1.26	169.18	170.43
CoLR@2	2.51	169.21	171.72
CoLR@4	5.03	169.27	174.30
CoLR@8	10.05	169.29	179.34
CoLR@16	20.11	169.30	189.41
CoLR@32	40.21	169.38	209.60
SVD@1	1.26	169.49	170.75
SVD@2	2.51	169.50	172.02
SVD@4	5.03	169.53	174.55
SVD@8	10.05	169.59	179.65
SVD@16	20.11	169.64	189.74
SVD@32	40.21	169.60	209.82
Top-K@1	2.51	169.76	172.28
Top-K@2	5.03	169.79	174.81
Top-K@4	10.05	169.82	179.87
Top-K@8	20.11	169.92	190.03
Top-K@16	40.21	170.14	210.35

In the previous results, the evaluation of techniques focuses on the overall number of transmitted bits. Although this serves

as a broad indicator, it fails to consider the time consumed by encoding/decoding processes and the fixed network latency within the system. When these time delays significantly exceed the per-bit communication time, employing compression techniques may offer limited or minimal benefits. In the following, we do an analysis to understand the effects of using CoLR and compression methods in training FedRec models.

We follow the model from [36] to estimate the communication efficiency of deploying methods to real-world systems. The execution time per round when deploying an optimization algorithm  $\mathcal{A}$  in a cross-device FL system is estimated as follows,

$$\begin{aligned}
 T_{\text{round}}(\mathcal{A}) &= T_{\text{comm}}(\mathcal{A}) + T_{\text{comp}}(\mathcal{A}), \\
 T_{\text{comm}}(\mathcal{A}) &= \frac{S_{\text{down}}(\mathcal{A})}{B_{\text{down}}} + \frac{S_{\text{up}}(\mathcal{A})}{B_{\text{up}}} \\
 T_{\text{comp}}(\mathcal{A}) &= \max_{j \in \mathcal{D}_{\text{round}}} T_{\text{client}}^j + T_{\text{server}}(\mathcal{A}), \\
 T_{\text{client}}^j(\mathcal{A}) &= R_{\text{comp}} T_{\text{sim}}^j(\mathcal{A}) + C_{\text{comp}}
 \end{aligned}$$

where client download size  $S_{\text{down}}(\mathcal{A})$ , upload size  $S_{\text{up}}(\mathcal{A})$ , server computation time  $T_{\text{server}}$ , and client computation time  $T_{\text{client}}^j$  depend on model and algorithm  $\mathcal{A}$ . Simulation time  $T_{\text{server}}$  and  $T_{\text{client}}^j$  can be estimated from FL simulation in our machine. We get the estimation of parameters  $(B_{\text{down}}, B_{\text{up}})$ ,  $R_{\text{comp}}$ ,  $C_{\text{comp}}$  from Wang et al. [36].

$$\begin{aligned}
 B_{\text{down}} &\sim 0.75\text{MB/secs}, B_{\text{up}} \sim 0.25\text{MB/secs}, \\
 R_{\text{comp}} &\sim 7, \text{ and } C_{\text{comp}} \sim 10 \text{ secs.}
 \end{aligned}$$

Table 2 presents our estimation in terms of communication times and computation time. Notice that CoLR adds smaller overheads to the computation time while still greatly reducing the communication cost.

**(3) CoLR is compatible with HE.** In this section, we argue that tackling privacy and communication efficiency as separate concerns can result in suboptimal solutions and point out the limitation in applying SVD and Top-K compression on HE-based FedRec systems.

Since performing SVD decomposition on an encrypted matrix remains an open problem, we conduct tests using two communication efficient methods: CoLR and Top-K. These tests are carried out under identical configurations, encompassing local updates and the number of clients involved in training rounds. The setup entails the utilization of the CKKS Cryptosystem [6] for our CoLR method, while the Top-K method employs the Paillier cryptosystem [30] for encryption, decryption, and aggregation in place of the top-K vector. The detailed implementation is described in Appendix C. Table 3 displays the client and server overheads in seconds, as well as the size of the ciphertext and plaintext.

Table 3 shows that CoLR can reduce client and server overheads by up to 3-10×. For Top-K compression, when the value of  $k$  doubles (i.e., doubling the top-K vector’s size), the operation time for both client-side and server-side operations also doubles, as it mandates operations on each value within the vector. Throughout the experiment, CoLR consistently outperforms the Top-K method across compression ratios, exhibiting lower time overheads on both the

**Table 3: Overheads, and Communication ratios for MovieLens-1M dataset; Comm Ratio is calculated by file sizes of Ciphertext over file sizes of Plaintext.**

Method	Client overheads	Server overheads	Ciphertext size	Plaintext size	Comm Ratio
FedMF	0.93 s	2.39 s	24,587 KB	927 KB	26.52
FedMF w/ Top-K@1/64	88.20 s	88.06 s	3,028 KB	29 KB	103.09
FedMF w/ Top-K@2/64	182.02 s	185.59 s	6,056 KB	58 KB	103.83
FedMF w/ Top-K@4/64	353.25 s	364.67 s	12,112 KB	116 KB	104.20
FedMF w/ Top-K@8/64	723.45 s	750.98 s	24,225 KB	232 KB	104.40
FedMF w/ Top-K@16/64	1449.90 s	1483.91 s	48,448 KB	464 KB	104.49
FedMF w/ CoLR@1	0.07 s	0.24 s	3,073 KB	15 KB	206.31
FedMF w/ CoLR@2	0.07 s	0.25 s	3,073 KB	29 KB	104.63
FedMF w/ CoLR@4	0.07 s	0.25 s	3,073 KB	58 KB	52.69
FedMF w/ CoLR@8	0.08 s	0.25 s	3,073 KB	116 KB	26.44
FedMF w/ CoLR@16	0.15 s	0.51 s	6,147 KB	232 KB	26.49
FedMF w/ CoLR@32	0.30 s	1.03 s	12,293 KB	464 KB	26.51

client and server sides. In terms of ciphertext sizes, the Top-K compression method with Paillier encryption demands encryption for each value within the top-K vector. Consequently, whenever the size of the top-K vector doubles, the ciphertext size also doubles. In contrast, as previously explained, our scheme produced at most  $\lceil \frac{n}{8096} \rceil$  blocks of ciphertext, with the ciphertext size not doubling each time  $k$  doubles. This phenomenon illustrates why, in several cases, the ciphertext size remains consistent even as the plaintext size increases. With lower payload reductions aimed at achieving greater recommendation performance, our scheme demonstrates smaller ciphertext sizes, offering a reduction in bandwidth consumption.

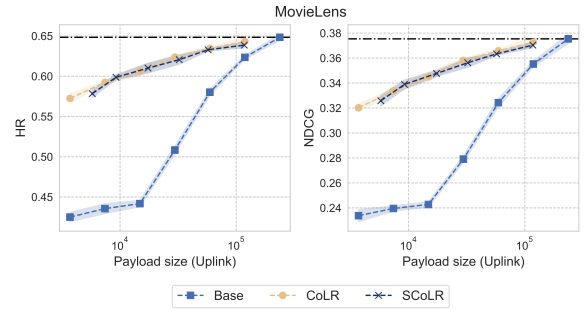
### 5.3 Heterogeneous network bandwidth

In this section, we evaluate our proposed method SCoLR and explore the scenario where each client can dynamically select  $r_u$  value during each training round  $t$ . This scenario reflects real-world federated learning, where clients often showcase differences in communication capacities, as exemplified in [18, 20]. It becomes inefficient to impose a uniform communication budget on all clients within this heterogeneous context, as some devices may not be able to harness their network connections fully.

For this experiment, we set the global rank  $r_g$  of SCoLR in the list of  $\{2, 4, 8, 16, 32, 64\}$  and uniformly sample the local rank  $r_u$  such that  $1 \leq r_u \leq r_g$ . It's crucial to emphasize that  $r_u$  is independently sampled for each user and may differ from one round to the next. This configuration mirrors a practical scenario where the available resources of a specific user may undergo substantial variations at different time points during the training phase. We present the result on the MovieLens-1M dataset in Table 4. We compare SCoLR with the base models and CoLR in Figure 5. This result demonstrates that SCoLR is effective under the device heterogeneity setting since it can match the performance of CoLR under the same uplink communication budget.

## 6 CONCLUSION

In this work, we propose Correlated Low-rank Structure update (CoLR), a framework that enhances communication efficiency and

**Figure 5: Performance of SCoLR on MovieLens-1M dataset. We plot the utilities versus the payload size. The dashed black line is the base model's performance.**

privacy preservation in FedRec by leveraging the inherent low-rank structure in updating transfers, our method reduces communication overheads. CoLR also benefits from the CKKS cryptosystem, which allows the implementation of a secured aggregation strategy within FedRec. With minimal computational overheads and bandwidth-heterogeneity awareness, it offers a flexible and efficient means to address the challenges of federated learning. For future research, we see several exciting directions. First, our framework still involves a central server, we would like to test how our methods can be effectively adapted to a fully decentralized, peer-2-peer communication setting [27, 46]. Secondly, investigating methods to handle dynamic network conditions and straggler mitigation in real-world settings will be crucial. Lastly, expanding our approach to accommodate more advanced secure aggregation techniques for reduced server-side computational costs and extending its compatibility with various encryption protocols can further enhance its utility in privacy-sensitive applications.

## ACKNOWLEDGMENTS

This research was funded by Vingroup Innovation Foundation (VINIF) under project code VINIF.2022.DA00087



## REFERENCES

- [1] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *ACM Comput. Surv.* 51, 4, Article 79 (jul 2018), 35 pages. <https://doi.org/10.1145/3214303>
- [2] Muhammad Ammad-Ud-Din, Elena Ivannikova, Suleiman A Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. 2019. Federated collaborative filtering for privacy-preserving personalized recommendation system. *ArXiv preprint abs/1901.09888* (2019), 4274–4282. <https://arxiv.org/abs/1901.09888>
- [3] Anonymous. 2023. Improving LoRA in Privacy-preserving Federated Learning. In *Submitted to The Twelfth International Conference on Learning Representations*. <https://openreview.net/forum?id=NLPzL6HWNl> under review.
- [4] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Dallas, Texas, USA) (CCS '17). Association for Computing Machinery, New York, NY, USA, 1175–1191. <https://doi.org/10.1145/3133956.3133982>
- [5] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2020. Secure federated matrix factorization. *IEEE Intelligent Systems* 36, 5 (2020), 11–20.
- [6] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. 2017. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology – ASIACRYPT 2017*, Tsuyoshi Takagi and Thomas Peyrin (Eds.). Springer International Publishing, Cham, 409–437.
- [7] Sijing Duan, Deyu Zhang, Yanbo Wang, Lingxiang Li, and Yaoxue Zhang. 2020. JointRec: A Deep-Learning-Based Joint Cloud Video Recommendation Framework for Mobile IoT. *IEEE Internet of Things Journal* 7, 3 (2020), 1655–1666. <https://doi.org/10.1109/JIOT.2019.2944889>
- [8] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*, Shai Halevi and Tal Rabin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 265–284.
- [9] Xue Geng, Hanwang Zhang, Jingwen Bian, and Tat-Seng Chua. 2015. Learning Image and User Features for Recommendation in Social Networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 4274–4282. <https://doi.org/10.1109/ICCV.2015.486>
- [10] Vipul Gupta, Dhruv Choudhary, Peter Tang, Xiaohan Wei, Xing Wang, Yuzhen Huang, Arun Kejariwal, Kannan Ramchandran, and Michael W. Mahoney. 2021. Training Recommender Systems at Scale: Communication-Efficient Model and Data Parallelism. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Virtual Event, Singapore) (KDD '21). Association for Computing Machinery, New York, NY, USA, 2928–2936. <https://doi.org/10.1145/3447548.3467080>
- [11] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2015), 1–19.
- [12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) (WWW '17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. <https://doi.org/10.1145/3038912.3052569>
- [13] István Hegedűs, Gábor Danner, and Márk Jelasity. 2020. Decentralized Recommendation Based on Matrix Factorization: A Comparison of Gossip and Federated Learning. In *Machine Learning and Knowledge Discovery in Databases*, Peggy Celier and Kurt Driessens (Eds.). Springer International Publishing, Cham, 317–332.
- [14] Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. 2022. FedPara: Low-rank Hadamard Product for Communication-Efficient Federated Learning. In *International Conference on Learning Representations*. OpenReview.net. <https://openreview.net/forum?id=d71n4ftoCBY>
- [15] Xiaoqian Jiang, Miran Kim, Kristin Lauter, and Yongsoo Song. 2018. Secure Outsourced Matrix Computation and Application to Neural Networks. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) (CCS '18). Association for Computing Machinery, New York, NY, USA, 1209–1222. <https://doi.org/10.1145/3243734.3243837>
- [16] Farwa K. Khan, Adrian Flanagan, Kuan Eeik Tan, Zareen Alamgir, and Muhammad Ammad-ud din. 2021. A Payload Optimization Method for Federated Recommender Systems. In *Proceedings of the 15th ACM Conference on Recommender Systems* (Amsterdam, Netherlands) (RecSys '21). Association for Computing Machinery, New York, NY, USA, 432–442. <https://doi.org/10.1145/3460231.3474257>
- [17] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*. <https://arxiv.org/abs/1610.05492>
- [18] Fan Lai, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. 2021. Oort: Efficient Federated Learning via Guided Participant Selection. In *15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21)*. USENIX Association, 19–35. <https://www.usenix.org/conference/osdi21/presentation/lai>
- [19] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60. <https://doi.org/10.1109/MSP.2020.2975749>
- [20] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60. <https://doi.org/10.1109/MSP.2020.2975749>
- [21] Feng Liang, Weike Pan, and Zhong Ming. 2021. FedRec++: Lossless Federated Recommendation with Explicit Feedback. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 5 (May 2021), 4224–4231. <https://doi.org/10.1609/aaai.v35i5.16546>
- [22] Chih-Lun Liao and Shie-Jue Lee. 2016. A clustering based approach to improving the efficiency of collaborative filtering recommendation. *Electron. Commer. Rec. Appl.* 18, C (jul 2016), 1–9. <https://doi.org/10.1016/j.elerap.2016.05.001>
- [23] Guanyu Lin, Feng Liang, Weike Pan, and Zhong Ming. 2020. Fedrec: Federated recommendation with explicit feedback. *IEEE Intell. Syst.* 36, 5 (2020), 21–30.
- [24] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Dongxiao Yu, Jun Ma, Maarten de Rijke, and Xiuzhen Cheng. 2020. Meta Matrix Factorization for Federated Rating Predictions. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) (SIGIR '20). Association for Computing Machinery, New York, NY, USA, 981–990. <https://doi.org/10.1145/3397271.3401081>
- [25] Shuchang Liu, Yingqiang Ge, Shuyuan Xu, Yongfeng Zhang, and Amelie Marian. 2022. Fairness-Aware Federated Matrix Factorization. In *Proceedings of the 16th ACM Conference on Recommender Systems* (Seattle, WA, USA) (RecSys '22). Association for Computing Machinery, New York, NY, USA, 168–178. <https://doi.org/10.1145/3523227.3546771>
- [26] Shuchang Liu, Shuyuan Xu, Wenhui Yu, Zuohui Fu, Yongfeng Zhang, and Amelie Marian. 2021. FedCT: Federated Collaborative Transfer for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (<conf-loc>, <city>Virtual Event</city>, <country>Canada</country>, </conf-loc>) (SIGIR '21). Association for Computing Machinery, New York, NY, USA, 716–725. <https://doi.org/10.1145/3404835.3462825>
- [27] L. Lyu, Y. Li, K. Nandakumar, J. Yu, and X. Ma. 2022. How to Democratise and Protect AI: Fair and Differentially Private Decentralised Deep Learning. *IEEE Transactions on Dependable and Secure Computing* 19, 02 (mar 2022), 1003–1017. <https://doi.org/10.1109/TDSC.2020.3006287>
- [28] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (Proceedings of Machine Learning Research, Vol. 54), Aarti Singh and Jerry Zhu (Eds.). PMLR, 1273–1282. <https://proceedings.mlr.press/v54/mcmahan17a.html>
- [29] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G. Azzolini, Dmytro Dzhalagov, Andrey Mallevich, Iliia Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, Wenlin Chen, Vijay Rao, Bill Jia, Liang Xiong, and Misha Smelyanskiy. 2019. Deep Learning Recommendation Model for Personalization and Recommendation Systems. arXiv:1906.00091 [cs.IR]
- [30] Pascal Paillier. 1999. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology – EUROCRYPT '99*, Jacques Stern (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 223–238.
- [31] Stuart L Paddau. 2018. The California consumer privacy act: Towards a European-style privacy regime in the United States. *J. Tech. L. & Pol'y* 23 (2018), 68.
- [32] Vasileios Perifanis, George Drosatos, Giorgos Stamatelatos, and Pavlos S. Efraimidis. 2023. FedPOIRec: Privacy-preserving federated poi recommendation with social influence. *Information Sciences* 623, C (apr 2023), 767–790. <https://doi.org/10.1016/j.ins.2022.12.024>
- [33] Vasileios Perifanis and Pavlos S Efraimidis. 2022. Federated Neural Collaborative Filtering. *Knowledge-Based Systems* 242 (2022), 108441.
- [34] Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. 2019. PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. [https://proceedings.neurips.cc/paper\\_files/paper/2019/file/d9fbed9da256e344c1fa46bb46c34c5f-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/d9fbed9da256e344c1fa46bb46c34c5f-Paper.pdf)
- [35] Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed.*, Cham: Springer International Publishing 10, 3152676 (2017), 10–5555.
- [36] Jianyu Wang, Zachary Burr Charles, Zheng Xu, Gauri Joshi, Brendan McMahan, Blaise Hilary Agüera-Arcas, Maruan Al-Shedivat, Galen Andrew, A. Salman Avestimehr, Katharine Daly, Deepesh Data, Suhas Diggavi, Hubert Eichner, Advait Gadhihar, Zachary Garrett, Antonios M. Girgis, Filip Hanzely, Andrew Hard, Chaoyang He, Samuel Horvath, Zhouyuan Huo, Alex Ingerman, Martin Jaggi, Tara Javidi, Peter Kairouz, Satyen Chandrakant Kale, Sai Praneeth Karimireddy, Jakub Konečný, Sanmi Koyejo, Tian Li, Luyang Liu, Mehryar Mohri, Hang Qi, Sashank Reddi, Peter Richtarik, Karan Singhal, Virginia Smith, Mahdi Soltanolkotabi, Weikang Song, Ananda Theertha Suresh, Sebastian Stich, Ameet Talwalkar, Hongyi Wang, Blake Woodworth, Shanshan Wu, Felix Yu,

- Honglin Yuan, Manzil Zaheer, Mi Zhang, Tong Zhang, Chunxiang (Jake) Zheng, Chen Zhu, and Wennan Zhu. 2021. A Field Guide to Federated Optimization. <https://arxiv.org/abs/2107.06917>
- [37] Li-e Wang, Yihui Wang, Yan Bai, Peng Liu, and Xianxian Li. 2021. POI Recommendation with Federated Learning and Privacy Preserving in Cross Domain Recommendation. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 1–6. <https://doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484510>
- [38] Qinyong Wang, Hongzhi Yin, Tong Chen, Junliang Yu, Alexander Zhou, and Xiangliang Zhang. 2021. Fast-adapting and privacy-preserving federated recommender system. *The VLDB Journal* (2021), 1–20.
- [39] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Tao Qi, Yongfeng Huang, and Xing Xie. 2022. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications* 13, 1 (2022), 1–10.
- [40] Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2022. FedAttack: Effective and Covert Poisoning Attack on Federated Recommendation via Hard Sampling. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) (KDD '22). Association for Computing Machinery, New York, NY, USA, 4164–4172. <https://doi.org/10.1145/3534678.3539119>
- [41] Liu Yang, Ben Tan, Vincent W. Zheng, Kai Chen, and Qiang Yang. 2020. *Federated Recommendation Systems*. Springer International Publishing, Cham, 225–239. [https://doi.org/10.1007/978-3-030-63076-8\\_16](https://doi.org/10.1007/978-3-030-63076-8_16)
- [42] Jingwei Yi, Fangzhao Wu, Chuhan Wu, Ruixuan Liu, Guangzhong Sun, and Xing Xie. 2021. Efficient-FedRec: Efficient Federated Learning Framework for Privacy-Preserving News Recommendation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2814–2824. <https://doi.org/10.18653/v1/2021.emnlp-main.223>
- [43] Wei Yuan, Quoc Viet Hung Nguyen, Tiek He, Liang Chen, and Hongzhi Yin. 2023. Manipulating Federated Recommender Systems: Poisoning with Synthetic Users and Its Countermeasures. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (<conf-loc>, <city>Taipei</city>, <country>Taiwan</country>, </conf-loc>) (SIGIR '23). Association for Computing Machinery, New York, NY, USA, 1690–1699. <https://doi.org/10.1145/3539618.3591722>
- [44] Honglei Zhang, Fangyuan Luo, Jun Wu, Xiangnan He, and Yidong Li. 2023. LightFR: Lightweight Federated Recommendation with Privacy-Preserving Matrix Factorization. *ACM Trans. Inf. Syst.* 41, 4, Article 90 (mar 2023), 28 pages. <https://doi.org/10.1145/3578361>
- [45] Shijie Zhang, Hongzhi Yin, Tong Chen, Zi Huang, Quoc Viet Hung Nguyen, and Lizhen Cui. 2022. PipAttack: Poisoning Federated Recommender Systems for Manipulating Item Promotion. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) (WSDM '22). Association for Computing Machinery, New York, NY, USA, 1415–1423. <https://doi.org/10.1145/3488560.3498386>
- [46] Zhizhao Zhang, Tianzhi Yang, and Yuan Liu. 2020. SABlockFL: a blockchain-based smart agent system architecture and its application in federated learning. *International Journal of Crowd Science* 4, 2 (2020), 133–147. <https://doi.org/10.1108/IJCS-12-2019-0037>

## A ALGORITHM DETAILS

In Section 4.4, we presented SCoLR to address the bandwidth heterogeneity problem. We provide the detail of this method in Algorithm 2 and the detail experimental results in Table 4.

**Algorithm 2:** Subsampling Correlated Low-rank Structure update (SCoLR)

---

**Input:** Initial model  $Q^{(0)}$ ; global update rank  $r_g$ , local update rank  $\{r_u\}$ , a distribution  $\mathcal{D}_B$  for initializing  $B$ ; CLIENTOPT, SERVEROPT with learning rates  $\eta, \eta_s$ ;

```

1 for  $t \in \{0, 1, 2, \dots, T\}$  do
2   Sample a subset  $S^{(t)}$  of clients and  $B^{(t)} \sim \mathcal{D}_B$ 
3   for client  $u \in S^{(t)}$  in parallel do
4     if  $t > 0$  then
5       Download  $A^{(t)}$  and merge
6        $Q_u^{(t,0)} = Q^{(t-1)} + B^{(t-1)}A^{(t)}$ 
7     end
8     Initialize  $Q_u^{(t,0)} = Q^{(t)}$ 
9     Download  $B^{(t)}$ , Initialize  $A_u^{(t,0)} = \mathbf{0}$ , and sample  $S_u^{(t)}$ 
10    Set trainable parameters  $\theta_u^{(t,0)} = \{A_u^{(t,0)}, \mathbf{p}_u^{(t,0)}\}$ 
11    for  $k = 0, \dots, \tau_u - 1$  do
12      Perform local update  $\theta_u^{(t,k+1)} =$ 
13      CLIENTOPT  $(\theta_u^{(t,k)}, \nabla_{\theta_u} \mathcal{L}_u(\theta_u^{(t,k)}), \eta)$ 
14    end
15     $\mathbf{p}_u^{(t+1)} = \mathbf{p}_u^{(t, \tau_u)}$ 
16    Upload  $\{S_u^{(t)}, A_u^{(t, \tau_u)}\}$  to the central server
17  end

```

---


$$A^{(t+1)} = \sum_{u \in S^{(t)}} \frac{N_u}{N} S_u^{(t)} A_u^{(t, \tau_u)};$$


---

**Table 4:** HR, NDCG of SCoLR algorithm on the MovieLens-1M dataset under computation/device heterogeneity settings.

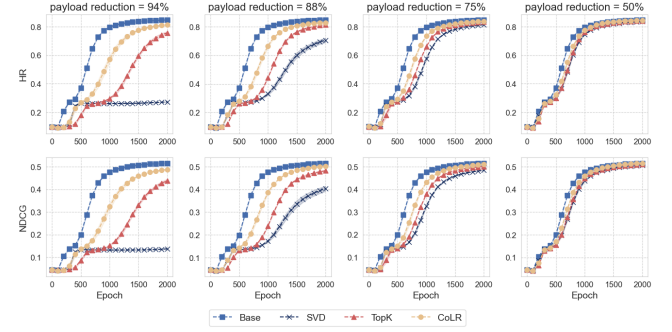
Global rank	Local rank	HR	NDCG
64	1 – 64	63.86	37.04
32	1 – 32	63.29	36.34
16	1 – 16	62.06	35.63
8	1 – 8	61.02	34.77
4	1 – 4	59.87	33.89
2	1 – 2	57.84	32.58

## B EXPERIMENTAL DETAILS

*Hyper-parameter settings.* To determine hyper-parameters, we create a validation set from the training set by extracting the second last interaction of each user and tuning hyper-parameters on it. We tested the batch size of [32, 64, 128, 256], the learning rate of [0.5, 0.1, 0.05, 0.01], and weight decay in  $[5e-4, 1e-4]$ . For each dataset,

we set the number of clients participating in each round to be equal to 1% of the number of all users in that dataset. We also vary the local epochs in [1, 2, 4]. The number of aggregation epochs is set at 1000 for MovieLens-1M and 2000 for Pinterest as the training process is converged at these epochs.

We plot the convergence speed of four methods on the Pinterest dataset in Figure 6.



**Figure 6:** Average HR and NDCG on Pinterest dataset varies as communication rounds.

## C HOMOMORPHIC ENCRYPTION WITH COMPRESSORS AND COLR

*Limitation of applying HE with SVD Compression:* The SVD method requires matrix multiplication on encrypted matrices  $U$ ,  $S$ , and  $V$  derived from local clients' updates. There are several research endeavors aimed at providing efficient algorithms for applying homomorphic encryption in this context, specifically using the CKKS cryptosystem [15]. However, a limitation of this method is that the dimension of the matrix must be in the form of  $2^n$ , often necessitating additional padding on the original matrices to achieve this form, particularly in the case of larger dimension matrices. Additionally, to reduce the size of global updates sent from the server to clients, additional SVD decompositions are required. Performing SVD decomposition on an encrypted matrix by known schemes remains an open problem, resulting in high downlink bandwidth consumption.

*Limitation of applying HE with TopK Compression:* The TopK compression method necessitates in-place homomorphic operations, a characteristic not compatible with the CKKS scheme, designed to perform homomorphic encryption on tensors. As an alternative, we have employed the Paillier cryptosystem [30], a partially homomorphic encryption scheme capable of encrypting individual numbers. While Paillier allows for the implementation of the FedAvg aggregation strategy, it requires the secured aggregation process on the server must be executed on each element in the TopK vector. Consequently, increasing the value of  $K$  results in higher operational costs for encryption, decryption, and secured aggregation.

*Implementation of HE with CoLR:* Our CoLR method leverages the inherent efficiency of the CKKS cryptosystem, which can execute operations on multiple values as a vector. For a flattened vector of size  $n$ , both clients and the server need to perform operations on at most  $\lceil \frac{n}{8096} \rceil$  blocks.

## D AN ANALYSIS ON THE INITIALIZATION OF THE MATRIX B

If each client performs only one GD step locally then  $B$  can be seen as the projection matrix and  $Ba_i$  is the projection of the update of item  $i$  on the subspace spanned by columns of  $B$ . We denote the error of the update on each item embedding  $i$  by  $\epsilon_i$  which has the following formulation:

$$\epsilon_i = \mathbb{E}_B \left[ \left\| \bar{\Delta}_Q - \frac{1}{|S|} \left( \sum_{u \in S} B_u a_u \right) \right\|_2^2 \right]. \quad (7)$$

We analyze the effect of different initialization of  $B$  on this error. First, we state the proposition D.1 which gives an upper bound on the error  $\epsilon_i$ .

**PROPOSITION D.1 (UPPER BOUND THE ERROR).** *If  $B_u$  is independently generated between users and are chosen from a distribution  $\mathcal{B}$  that satisfies:*

- (1) *Bounded operator norm:*  $\mathbb{E} [\|B\|^2] \leq L_{\mathcal{B}}$
- (2) *Bounded bias:*  $\|\mathbb{E} B_u B_u^\top \bar{\mathbf{p}}_u - \bar{\mathbf{p}}_u\|_2 \leq \sqrt{\delta_{\mathcal{B}}}$

Then,

$$\epsilon_i = \mathbb{E}_B \left[ \left\| \bar{\Delta}_Q - \frac{1}{|S|} \left( \sum_{u \in S} B_u a_u \right) \right\|_2^2 \right] \quad (8)$$

$$\leq \frac{1}{|S|} C_{\mathcal{B}}^2 \delta_{\mathcal{B}} + \frac{1}{|S|} \max_{u \in S} \alpha_u \|\mathbf{p}_u\|_2^2 (L_{\mathcal{B}}^2 + 1). \quad (9)$$

**PROOF.** Assume  $B_u$  is independently generated between users, we have

$$\begin{aligned} \epsilon_i &= \frac{1}{|S|^2} \mathbb{E}_B \left[ \left\| \sum_{u \in S} (r_{ui} - \hat{r}_{ui}) (B_u B_u^\top \mathbf{p}_u - \mathbf{p}_u) \right\|_2^2 \right] \\ &= \frac{1}{|S|^2} \mathbb{E}_B \left[ \left\| \sum_{u \in S} \alpha_u (B_u B_u^\top \mathbf{p}_u - \mathbf{p}_u) \right\|_2^2 \right] \\ &= \frac{1}{|S|^2} \sum_{u_1 \in S} \sum_{u_2 \neq u_1} \alpha_{u_1} \alpha_{u_2} \mathbb{E}_B \langle B_{u_1} B_{u_1}^\top \mathbf{p}_{u_1} - \mathbf{p}_{u_1}, B_{u_2} B_{u_2}^\top \mathbf{p}_{u_2} - \mathbf{p}_{u_2} \rangle \\ &\quad + \frac{1}{|S|^2} \sum_{u \in S} \alpha_u^2 \mathbb{E}_B \left[ \|B_u B_u^\top \mathbf{p}_u - \mathbf{p}_u\|_2^2 \right] \end{aligned}$$

If  $B_u$  are independently chosen from a distribution  $\mathcal{B}$  that satisfies:

- (1) *Bounded operator norm:*  $\mathbb{E} [\|B\|^2] \leq L_{\mathcal{B}}$
- (2) *Bounded bias:*  $\|\mathbb{E} B_u B_u^\top \bar{\mathbf{p}}_u - \bar{\mathbf{p}}_u\|_2 \leq \sqrt{\delta_{\mathcal{B}}}$

We have

$$\begin{aligned} &\mathbb{E}_B \langle B_{u_1} B_{u_1}^\top \mathbf{p}_{u_1} - \mathbf{p}_{u_1}, B_{u_2} B_{u_2}^\top \mathbf{p}_{u_2} - \mathbf{p}_{u_2} \rangle \\ &= \|\mathbf{p}_{u_1}\| \|\mathbf{p}_{u_2}\| \mathbb{E}_B \langle B_{u_1} B_{u_1}^\top \bar{\mathbf{p}}_{u_1} - \bar{\mathbf{p}}_{u_1}, B_{u_2} B_{u_2}^\top \bar{\mathbf{p}}_{u_2} - \bar{\mathbf{p}}_{u_2} \rangle \\ &= \|\mathbf{p}_{u_1}\| \|\mathbf{p}_{u_2}\| \langle \mathbb{E} B_{u_1} B_{u_1}^\top \bar{\mathbf{p}}_{u_1} - \bar{\mathbf{p}}_{u_1}, \mathbb{E} B_{u_2} B_{u_2}^\top \bar{\mathbf{p}}_{u_2} - \bar{\mathbf{p}}_{u_2} \rangle \quad (10) \\ &\leq \|\mathbf{p}_{u_1}\| \|\mathbf{p}_{u_2}\| \|\mathbb{E} B_{u_1} B_{u_1}^\top \bar{\mathbf{p}}_{u_1} - \bar{\mathbf{p}}_{u_1}\|_2 \|\mathbb{E} B_{u_2} B_{u_2}^\top \bar{\mathbf{p}}_{u_2} - \bar{\mathbf{p}}_{u_2}\|_2 \\ &\leq C_{\mathcal{B}}^2 \delta_{\mathcal{B}} \quad (11) \end{aligned}$$

where (11) follows since  $B_u$  are independently sampled between users. The second term is

$$\begin{aligned} &\frac{1}{|S|^2} \sum_{u \in S} \alpha_u^2 \mathbb{E}_B \left[ \|B_u B_u^\top \mathbf{p}_u - \mathbf{p}_u\|_2^2 \right] \\ &= \frac{1}{|S|^2} \sum_{u \in S} \alpha_u^2 \|\mathbf{p}_u\|_2^2 \mathbb{E}_B \left[ \|B_u B_u^\top \bar{\mathbf{p}}_u - \bar{\mathbf{p}}_u\|_2^2 \right] \\ &= \frac{1}{|S|^2} \sum_{u \in S} \alpha_u \|\mathbf{p}_u\|_2^2 \mathbb{E}_B \left[ \|B_u B_u^\top \bar{\mathbf{p}}_u\|_2^2 + \|\bar{\mathbf{p}}_u\|_2^2 - 2 \bar{\mathbf{p}}_u^\top B_u B_u^\top \bar{\mathbf{p}}_u \right] \\ &= \frac{1}{|S|^2} \sum_{u \in S} \alpha_u \|\mathbf{p}_u\|_2^2 \mathbb{E}_B \left[ \|B_u B_u^\top \bar{\mathbf{p}}\|_2^2 + 1 - 2 \bar{\mathbf{p}}_u^\top B_u B_u^\top \bar{\mathbf{p}} \right] \\ &\leq \frac{1}{|S|} \max_{u \in S} \alpha_u \|\mathbf{p}_u\|_2^2 \left( \mathbb{E}_B \left[ \|B_u B_u^\top\|_2^2 \right] + 1 \right) \\ &\leq \frac{1}{|S|} \max_{u \in S} \alpha_u \|\mathbf{p}_u\|_2^2 (L_{\mathcal{B}}^2 + 1) \end{aligned}$$

□

Next, we bound the bias and the operator norm of  $B_u$  if it is sampled from a Gaussian distribution in the lemma D.2.

**LEMMA D.2 (GAUSSIAN INITIALIZATION).** *Let  $r < d$ . Consider  $B \in \mathbb{R}^{d \times r}$  be sampled from the Gaussian distribution where  $B$  has i.i.d.  $\mathcal{N}(0, 1/k)$  entries and a fixed unit vector  $\mathbf{v} \in \mathbb{R}^d$ . Then*

- (1) *Bounded operator norm:*

$$\mathbb{E} \|B\|^2 \leq \frac{d}{r} \left( 1 + O \left( \sqrt{\frac{r}{d}} \right) \right)$$

- (2) *Unbias: for every unit vector  $\mathbf{v} \in \mathbb{R}^d$*

$$\|\mathbb{E} B B^\top \mathbf{v} - \mathbf{v}\| = 0$$

**PROOF.** Let  $B' = PB$  where  $P \in \mathbb{R}^{d \times d}$  is the rotation matrix such that  $P\mathbf{v} = \mathbf{e}_1$ . Due to the rotational symmetry of the normal distribution,  $B'$  is a random matrix with i.i.d.  $\mathcal{N}(0, 1/r)$  entries. Note that  $B = P^\top B'$ .

$$\begin{aligned} \mathbb{E}_B [B B^\top \mathbf{v}] &= \mathbb{E}_B [P^\top P B B^\top P^\top P \mathbf{v}] \\ &= P^\top \mathbb{E}_B [B' B'^\top \mathbf{e}_1] \end{aligned}$$

Let  $\mathbf{z} = B' B'^\top \mathbf{e}_1$ . Notice that  $\mathbf{z}_j = \langle B'^\top \mathbf{e}_j, B'^\top \mathbf{e}_1 \rangle$ . Because  $B'$  has i.i.d.  $\mathcal{N}(0, 1/r)$  entries,  $\mathbf{z}_1 = \|B'^\top \mathbf{e}_1\|_2^2 = \sum_{k=1}^r (B'_{1k})^2$  is  $1/r$  times a Chi-square random variable with  $r$  degrees of freedom. So  $\mathbb{E}[\mathbf{z}_1] = \frac{1}{r} r = 1$  and  $\mathbb{E}[\mathbf{z}_j] = 0 \forall j > 1$ . Thus,  $\mathbb{E}[\mathbf{z}] = \mathbf{e}_1$ . Therefore,  $\|\mathbb{E}_B [B B^\top \mathbf{v}]\| = \|P^\top \mathbf{e}_1 - \mathbf{v}\| = 0$ . □

From Proposition D.1 and Lemma D.2, we can directly get the following theorem which bound the error of restricting the local update in a low-rank subspace which is randomly sampled from a normal distribution.

**THEOREM D.3.** *Assume  $B_u$  is independently generated between users and are chosen from the normal distribution  $\mathcal{N}(0, 1/r)$ . Then,*

$$\begin{aligned} \epsilon_i &= \mathbb{E}_B \left[ \left\| \bar{\Delta}_Q - \frac{1}{|S|} \left( \sum_{u \in S} B_u a_u \right) \right\|_2^2 \right] \\ &\leq \frac{1}{|S|} \max_{u \in S} \alpha_u \|\mathbf{p}_u\|_2^2 O \left( \frac{d}{r} \right). \end{aligned}$$

This result demonstrates that the square error can increase for lower values of the local rank  $r$ . Building on this insight, we suggest scaling the learning rate of the low-rank components by  $\sqrt{\frac{r}{d}}$  to counter the error.