Towards Federated Low-Rank Adaptation with Rank Heterogeneity

Yuji Byun
Defense science and Technology
Pohang University of Science and Technology
Pohang, South Korea
yujibyun@postech.ac.kr

Jaeho Lee
Department of Electrical Engineering
Pohang University of Science and Technology
Pohang, South Korea
jaeho.lee@postech.ac.kr

Abstract

Low-rank adaptation (LoRA) is an attractive alternative of adapting full weights for the federated fine-tuning of large pretrained models, which can significantly reduce computational burden. In principle, federated LoRA can provide an effective mean to allocate different resources to each client by tuning ranks for each client. However, we find that the empirical performance of LoRA is highly unstable with respect to such rank-heterogeneity. Our investigation reveals that the root cause of this instability is the zero-padding-based aggregation strategy adopted in conventional federated LoRA frameworks, which causes the information from high rank clients to become diluted during the aggregation process. To address this issue, we propose a new replication-based padding strategy, which allows us to better leverage the information from clients with high-quality datasets. This method ensures that valuable information from high rank clients is retained during the aggregation process, accelerating the convergence speed and enhancing the overall prediction quality of the global model.

1 Introduction

Large-scale pretrained models, or *foundation models*, have demonstrated unprecedentedly strong performance on various tasks (Achiam et al., 2023; Radford et al., 2021; Kirillov et al., 2023), but they also have unprecedentedly large number of parameters. Large language models (LLMs), for instance, often have over trillions of parameters (Achiam et al., 2023). This vast scale is problematic as it incurs much computational burden. In this context, low-rank adaptation (LoRA) has emerged as a promising option for federated fine-tuning from pretrained weights (Hu et al., 2022). Instead of fine-tuning full weight matrices, LoRA keeps the original weight matrices frozen and trains only the updates, which is parametrized as a product of two low-rank matrices. This dramatically reduces the effective number of parameters.

As research on LoRA progresses, its application in federated learning has also gained attention, with some studies exploring the use of LoRA in heterogeneous setting. This aspect, however, is immensely understudied in the literature, leaving much room for improvements (Babakniya et al., 2023; Cho et al., 2024). In this paper, we identify a critical shortcoming of existing frameworks for rank-heterogeneous federated low-rank adaptation: Whenever the client quality varies significantly, existing rank-heterogeneous methods tend to converge slower than the rank-homogeneous federated learning that uses a smaller total bandwidth. Our investigation reveals that this is primarily due to the suboptimal *aggregation* strategy used in conventional federated LoRA frameworks (Cho et al., 2024); to aggregate updates with disparate rank, such works rely on *zero-padding*, i.e., matching the dimensionality by concatenating all-zero rows and columns to the low-rank-decomposed parameter

Submitted to Workshop on Adaptive Foundation Models@NeurIPS 2024. Do not distribute.

updates. Such strategy may be suboptimal, as the high-priority information in the highest rank LoRA update (from the privileged client) can be made less relevant by being averaged with padded zeros.

To address this problem, we develop a very simple yet effective *replication* strategy for aggregating rank-heterogeneous LoRA updates in the setting where clients have high variances in the data quality. To avoid having highly relevant information from being diluted, our strategy pads the lower rank updates with the replicated entries of the higher-priority clients, instead of zeros. Empirically, the proposed method achieves faster convergence to the higher accuracy than existing rank-homogeneous and heterogeneous paradigms.

2 Background and Related work

2.1 Federated Fine-tuning with LoRA

Low-Rank Adaptation, or simply LoRA (Hu et al., 2022), is a parameter-efficient fine-tuning (PEFT) method that keeps the original pretrained weight parameters fixed and only trains newly added parameters. More concretely, consider fine-tuning a pretrained weight matrix $W_{\text{pre}} \in \mathbb{R}^{m\times n}$. LoRA reparametrizes the updated weight matrix $W_{\text{ft}} \in \mathbb{R}^{m\times n}$ as a sum of the original weight matrix and a product of two low-rank matrices:

$$W_{\text{ft}} = W_{\text{pre}} + BA, \qquad A \in \mathbb{R}^{r\times n},\ B \in \mathbb{R}^{m\times r}, \tag{1}$$

where r is the rank of the parameter update. As we keep the original weight matrix W_{pre} frozen, only the matrices A and B are trainable parameters. Thus, the number of (active) parameters becomes $(m+n)r$, which can be smaller than the number of parameters for the original matrix mn whenever the rank r meets the condition $r < mn/(m+n)$. Typically, the rank r is chosen to be much smaller than the dimensionality of the weight matrix m, n. For instance, for fine-tuning LLMs, e.g., Llama (Touvron et al., 2023), it is conventional to use $r = 16$ for the matrices of size $m = n = 4096$. In this case, the number of parameter reduces to the $1/128 \approx 0.78\%$ of the original weight matrix, leading to a proportional decrease in the communication cost for federated fine-tuning.

2.2 Data Heterogeneity and the Federated PEFT

Data heterogeneity, or the discrepancy among the client-wise data distribution, has been studied extensively in federated learning. Such heterogeneity is very common in real world scenarios, and can severely degrade the model performance (Zhu et al., 2021). Many works have focused on resolving this issue, proposing various solutions including that involve data sharing (Zhao et al., 2018) or better calibration of batch normalization (Li et al., 2021).

The dataset heterogeneity has also been discussed in the context of parameter-efficient federated learning. For instance, Kim et al. (2023) study how the negative impacts of dataset heterogeneity can be mitigated the federated learning of adapters (Houlsby et al., 2019). Most closely related to our work, Cho et al. (2024) consider assigning different rank for the clients, as a mean of addressing inter-client heterogeneity.

In contrast to these works, our work primarily focuses on the scenario where the *relative importance* of each client can be dramatically different. Even clients with similar amounts of data can have significantly different importances depending on whether the data is uniformly distributed, and when both clients have a similar degree of imbalance but different majority classes, their importances can also differ. When some clients are notably of better quality than others, we demonstrate that the algorithm of Cho et al. (2024) may not be effective; our work proposes a way to fix this problem.

3 Method

Aggregation with zero-padding. In the federated LoRA, the server receives k different LoRA updates from the clients:

$$\Delta W_1 = B_1 A_1, \quad A_1 \in \mathbb{R}^{r_1\times n}\ B_1 \in \mathbb{R}^{m\times r_1} \tag{2}$$

$$\Delta W_i = B_i A_i, \quad A_i \in \mathbb{R}^{r_2\times n}\ B_i \in \mathbb{R}^{m\times r_2}\ (i \neq 1) \tag{3}$$

where we assume $r_1 > r_2$. To aggregate LoRA updates with mismatched dimension, (Cho et al., 2024) proposes to perform *zero-padding*, i.e., concatenate zero columns and rows to the LoRA

Table 1: Comparison of accuracy before and after aggregation.

	Method	round 1		round 2		round 3	
		before	after	before	after	before	after
Client A (balanced)	Zero-padding	84.34	38.95	71.58	42.92	86.58	50.53
	Replication method	84.34	82.11	88.82	86.16	89.47	86.05
Others (average)	Zero-padding	24.96	23.95	31.07	43.42	45.06	49.11
	Replication method	24.96	23.95	31.07	44.08	44.48	76.63

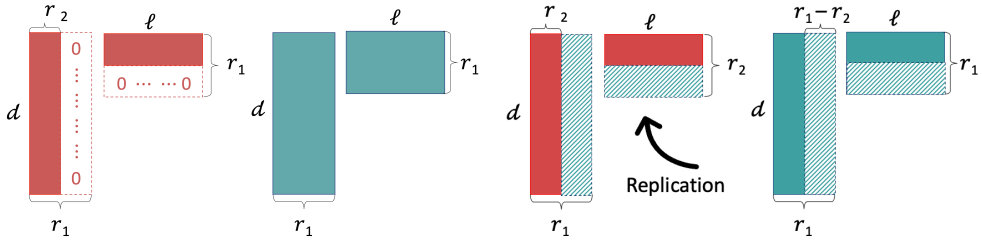

Figure 1: A comparison of two strategies for aggregating rank-heterogeneous LoRA updates. Left: A visual illustration of the conventional zero-padding strategy. Right: A visual illustration of the proposed replication-based strategy.

updates. That is, the matrix B_i is padded with all-zero columns as $\tilde{B}_i = [B_i|0|0|\cdots|0]$, where the number of zero-padded columns is $r_1 - r_2$. By averaging the zero-padded weight updates, the left r_1 columns may retain the same relative scale as the original weight upate. However, the remaining r_2 columns may have the relative scale of $1/k$, having their impact on the overall model much diminished as the number of clients grow. In this sense, the conventional zero-padding strategy can be deemed suboptimal in retaining the information of the high-quality client as shown in table 1.

To resolve the issue, we propose a very simple yet effective solution. In particular, we develop a *replication*-based aggregation strategy, which replicates the parameter-update information from the high-quality clients and applies it to pad the updates for the lower rank clients. This operation can be done rapidly, thus incurring only negligible latency to the overall federated learning pipeline.

Replication algorithm: A simplified description. For simplicity, we first describe the case where we have one high rank client and one low rank client. Concretely, let $\Delta W_1 = B_1 A_1$ be the high rank parameter updates from the high-quality client ($r = r_1$), and let $\Delta W_2 = B_2 A_2$ be the low rank parameter update from another client ($r = r_2 < r_1$). The padded version of the low rank matrix is

$$\tilde{B}_2 = \left[B_2 \middle| b_1^{(r_2+1)} \middle| \cdots \middle| b_1^{(r_1)} \right], \tag{4}$$

where $b_1^{(i)}$ denotes the i-th column vector of the matrix B_1. We process the matrix A_2 in a similar manner (figure 1). After this step, we average the weight matrices as usual in the FedAvg (McMahan et al., 2017). That is, we perform the averaging:

$$B_{\text{new}} = \frac{1}{2}\left(B_1 + \tilde{B}_2\right), \quad A_{\text{new}} = \frac{1}{2}\left(A_1 + \tilde{A}_2\right) \tag{5}$$

The case of multiple clients. Whenever there are multiple high rank clients, we handle this in three steps: (1) Aggregate the high rank clients (2) Replicate the entries of the aggregated high rank clients (3) Take a weighted average of the padded low rank and the aggregated high rank LoRA updates; here, we set the relative weight of the aggregated high rank LoRA updates to be proportional to the number of high rank clients.

Allocating high rank. First, we allocate low rank to all clients. After the first local update phase, the server select top-k clients with the highest validation accuracy. The clients are then allocated of a high rank. We test this algorithm in Experiments section, where we find that the proposed algorithm often performs even better than collecting high rank updates from every clients.

4 Experiments

4.1 Experimental Setup

Datasets. We focus on the task of text classification. We use two datasets: AG's News (Zhang et al., 2015), and DBpedia (Auer et al., 2007). We use 10% of the test set for validation, and use the rest for testing.

Models. Considering the fact that the computational resources of edge clients are often limited, we use two lightweight BERT-style language models: DistilBERT (Sanh et al., 2019), and ALBERT (Lan et al., 2020). For classification, we add an initialized classification layer to these models; the layer is frozen to its initial values without further training, as in (Sun et al., 2024). We apply LoRA only to the self-attention layer.

Clients. We employ total 100 clients, and the training dataset is distributed over these clients without overlap. We model two different types of clients: (1) *High-quality* (HQ) clients have more balanced local data, i.e., having similar number of samples from each class. (2) *Low-quality* (LQ) clients have datasets with more class imbalance, i.e., minority classes can have very few samples. We randomly select 10% of all clients to be HQ clients, and the remaining 90% to be LQ clients. To implement the clients, we follow prior studies (Lin et al., 2021; Babakniya et al., 2023) to apply Dirichlet distribution for generating non-i.i.d. datasets. The distribution is parametrized by a hyperparameter α; the smaller α indicates more heterogeneity, and the larger α implies more uniformity. We use $\alpha = 5.0$ and $\alpha = 1.0$ for HQ and LQ data, respectively. The average number of samples for both HQ and LQ data have been set to be equal. At the initial round, we apply $r = 5$ to all clients. After initial round, we assign $r = 20$ to the top 10% clients that achieve highest validation accuracy.

Training. We follow FedAvg (McMahan et al., 2017) to conduct one local epoch training per a global round. We randomly select 10% of clients to participate global round, ensuring that the proportion of clients with high and low rank remains consistent with the overall distribution. We use Adam with the learning rate 5e-4, without any further scheduling.

Baselines. We compare the proposed replication-based aggregation strategy with three baselines. (1) *Homogeneous.* All clients have a same rank; there is no need to aggregate or truncate. We evaluate $r \in \{5, 7, 20\}$, where $r = 7$ has a similar total communication cost with the rank-heterogeneous LoRA; see table 2. (2) *Naïve zero-padding.* The strategy where one pads all-zero rows and all-zero columns to match the dimensionality of rank-heterogeneous weight updates (proposed in Cho et al. (2024)). (3) *Frobenius zero-padding.* One performs the same zero-padding, but applies a weighted sum instead of averaging, with weight proportional to the frobenius norm of the product matrix $\|\Delta W_i\|_F$ (proposed in (Cho et al., 2024)).

4.2 Experimental Results

The leftmost data point denotes the accuracy at initialization (thus can be ignored when comparing baselines), and the subsequent data points denote the test accuracies after each communication round.

Result in DistilBERT. (Figure 2, top row) Our first observation is that the proposed replication-based aggregation strategy (red) achieves the fastest convergence over all compared methods in both cases. In particular, the proposed strategy successfully achieves the near-peak test accuracy in two communication rounds. In terms of the converged test accuracy, the proposed strategy is also among one of the best methods, with the communication-heavy option (homogeneous rank 20; orange) only slightly outperforming on AG's News. Zero-padding-based strategies (dashed lines with circles) converge slower than rank-homogeneous options, with Frobenius padding converging slightly faster than naïve. Among rank-homogeneous models, the one with a higher rank tends to converge faster, to a higher final accuracy than the one that uses a lower rank.

Result in ALBERT. (Figure 2, bottom row) Similarly, our method achieves a the fastest convergence to the high accuracy, only slightly worse than the communication-heavy case (homogeneous rank 20). We note that, in AG's News, the homogeneous LoRA tend to perform slightly better than the replication-based padding after the very first round; this is because the quality of the high rank client selected in the step by our method happened to be worse than other high rank clients. However, our method quickly starts to outperform the baselines in the subsequent rounds; this suggests that our method performs robust w.r.t. the suboptimalities in the high rank client selection.

Figure 2: Test accuracy of DistilBERT (top) and ALBERT (bottom) on the AG's News (left) and DBPedia (right) datasets.

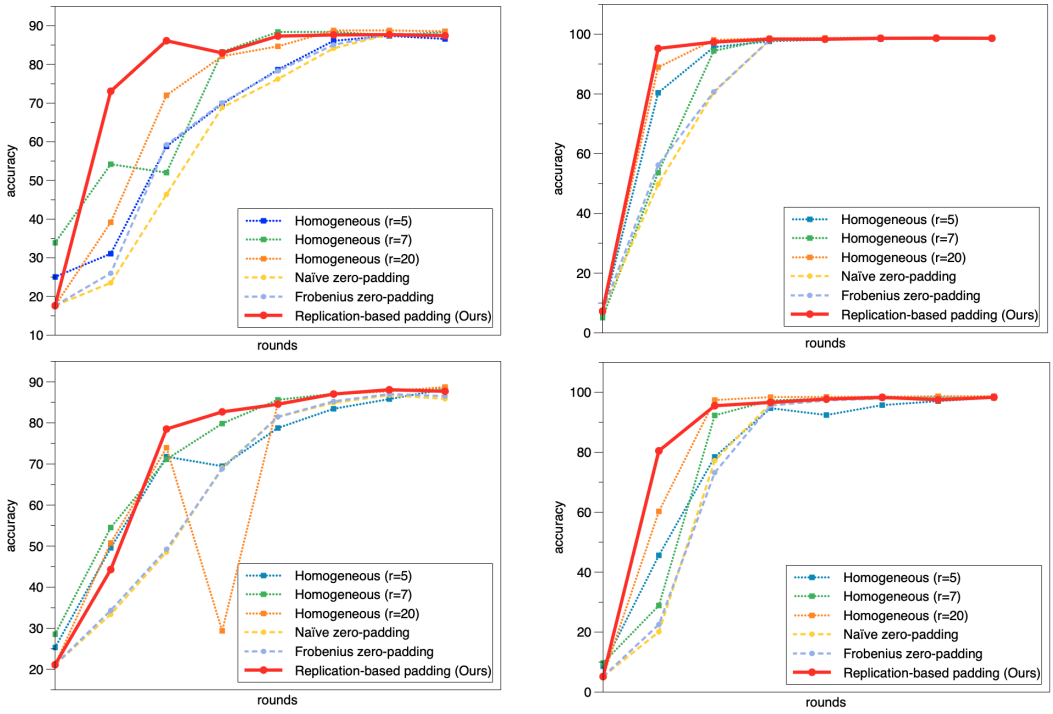

Table 2: Comparison of Communication Costs by Rank with DistilBERT

Method	***LoRA (r=20)***	***LoRA (r=7)***	***Ours***	***LoRA (r=5)***
num of parameters	552,960	193,536	179,715	138,240
communication cost	2.11MB	0.74MB	0.69MB	0.53MB
percent of total model	0.83%	0.30%	0.27%	0.21%

Comparison of communication cost. In this part, we make a quantitative comparison of the communication cost used by each rank-homogeneous and hetero-geneous methods (table 2). In particular, we compare the uplink communication, i.e., sent to server, which is the main communication bottleneck. We compare the communication cost used per client (in average) for the transmission of LoRA updates. We use DistilBERT for comparison. *Percent of total model* indicates the percentage of parameters used when applying LoRA compared to fully fine-tuning the original DistilBERT.

From the table, we confirm that our method uses less communication bandwidth than rank-homogeneous options with $r = 7$. As our method requires smaller number of rounds for convergence, our method is communication-efficient than rank-homogeneous options in both ends.

5 Conclusion

We have analyzed the negative impacts that the zero-padding method during the aggregation process when using heterogeneous LoRA in federated learning, and proposed a replication-based padding method to address these issues. We have experimentally demonstrated that this method not only outperforms heterogeneous LoRA using zero-padding but also achieves faster convergence compared to homogeneous LoRA with high ranks. This suggests that in situations with extremely limited bandwidth, a strategy of assigning higher ranks to only a limited set of clients—while leaving others with low rank—can reduce the overall communication cost burden. We believe that our research opens up new challenges and opportunities in federated fine-tuning, and we are confident that this study will contribute to more efficient federated learning in terms of communication costs.

Acknowledgment

This work has been supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00453301, No. RS-202300213710).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *IEEE International Conference on Computer Vision*, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Sara Babakniya, Ahmed Roushdy Elkordy, Yahya H Ezzeldin, Qingfeng Liu, Kee-Bong Song, Mostafa El-Khamy, and Salman Avestimehr. SLoRA: Federated parameter efficient fine-tuning of language models. In *Workshop on Federated Learning in the Age of Foundation Models @ NeurIPS*, 2023.

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, Matt Barnes, and Gauri Joshi. Heterogeneous LoRA for federated fine-tuning of on-device foundation models. In *Conference on Empirical Methods in Natural Language Processing*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint 2302.13971*, 2023.

Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 2021.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

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

Yeachan Kim, Junho Kim, Wing-Lam Mok, Jun-Hyung Park, and SangKeun Lee. Client-customized adaptation for parameter-efficient federated learning. In *Findings of the Association for Computational Linguistics*, 2023.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, 2019.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Conference on Artificial Intelligence and Statistics*, 2017.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 2015.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In *International Semantic Web Conference*, 2007.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *International Conference on Learning Representations*, 2020.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving LoRA in privacy-preserving federated learning. *International Conference on Learning Representations*, 2024.

Bill Yuchen Lin, Chaoyang He, Zihang Zeng, Hulin Wang, Yufen Huang, Christophe Dupuy, Rahul Gupta, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr. FedNLP: Benchmarking federated learning methods for natural language processing tasks. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2021.