Towards Federated Low-Rank Adaptation of Language Models with Rank Heterogeneity

Yuji Byun[1,2] and **Jaeho Lee**[1]
[1]Pohang University of Science and Technology (POSTECH), [2]ROK Marine Corps
{yujibyun,jaeho.lee}@postech.ac.kr

Abstract

Low-rank adaptation (LoRA) offers an efficient alternative to full-weight adaptation in federated fine-tuning of language models, significantly reducing computational costs. By adjusting ranks for each client, federated LoRA enables flexible resource allocation. However, we observe that heterogeneous ranks among clients lead to unstable performance. Our analysis attributes this instability to the conventional zero-padding aggregation strategy, which dilutes information from high-rank clients during model aggregation. To address this issue, we propose a replication-based padding strategy that better retains valuable information from clients with high-quality data. Empirically, this approach accelerates convergence and enhances the global model's predictive performance.

1 Introduction

Modern language models have shown unprecedentedly strong performance on many tasks (Achiam et al., 2023), but they also have unprecedentedly many parameters. Their gigantic sizes become especially problematic in *federated fine-tuning* of language models, where the cost to compute and communicate local gradients grows proportionally to the number of parameters (Yao et al., 2024).

To address this, recent works adopt low-rank adaptation (LoRA; Hu et al. (2022)) for federated fine-tuning of language models. Instead of tuning all weights, LoRA freezes original weights and trains only the update parametrized as a product of two low-rank matrices. This reduces the number of parameters, thus reducing the computation and communication needed (Babakniya et al., 2023).

A key promise of federated LoRA is its potential to improve the resource-accuracy tradeoff by adjusting client-wise ranks (Cho et al., 2024). Such rank-heterogeneity provides not only a handy way to tune client-wise computation and communication budgets, but also a mean to bias the global update toward certain clients that are considered giving higher-quality gradient estimates.

In this work, we identify a critical shortcoming of existing rank-heterogeneous federated LoRA methods for language models. Whenever the *quality* of clients varies significantly, conventional rank-heterogeneous LoRA struggles to converge faster than naïve rank-homogeneous LoRA. Our analysis suggests that such underperformance might be due to suboptimal *aggregation* strategy; to aggregate LoRA updates with disparate ranks, typical works adopt *zero-padding* strategy, i.e., matching the dimensionality by appending all-zero rows and columns to the low-rank-decomposed parameter updates (Cho et al., 2024). This strategy may not be optimal whenever there exists some clients which provides much higher-quality information, as the information from such clients can be made less relevant by being averaged with padded zeros.

To tackle this problem, we develop a simple yet effective fix, called *replication* strategy. To avoid having highly relevant information from being diluted, we pad lower rank updates with rows and columns replicated from high-priority clients, instead of zeros. Empirically, the proposed method achieves faster convergence to the higher accuracy than existing rank-homogeneous and heterogeneous paradigms. In short, our contributions are:

- We identify the shortcomings of existing rank-heterogeneous federated LoRA frameworks for language models, *i.e.*, unexpected slow convergence under high client quality disparity.
- We diagnose the problems in zero-padding-based aggregation, *i.e.,* failing to preserve information from high-quality clients.
- We propose a new replication-based aggregation strategy designed to preserve the important information in high-priority clients better, and empirically demonstrate that the proposed method outperforms baseline methods.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 356–362

2 Background

LoRA. LoRA is a parameter-efficient fine-tuning (PEFT) method that keeps the pretrained weights fixed and only trains newly added parameters (Hu et al., 2022). 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:

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

where r is the rank of the parameter update. As we keep W_{pre} frozen, only 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 r is sufficiently small. For fine-tuning language models, e.g., LLaMA (Touvron et al., 2023), it is typical 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 matrix, leading to a proportional decrease in the communication cost for federated fine-tuning.

Federated LoRA. In federated LoRA with k clients, the server receives k different LoRA updates from the clients. That is, the server receives

$$\Delta W_i = B_i A_i, \quad A_i \in \mathbb{R}^{r_i\times n}, B_i \in \mathbb{R}^{m\times r_i} \quad (2)$$

In rank-homogeneous LoRA (*i.e.*, $r_i = r$), a basic way to aggregate the updates from the clients may aggregated by taking an average for both A and B (McMahan et al., 2017). Concretely, one performs

$$\bar{A} = \frac{1}{k}\sum_{i=1}^{k} A_i, \quad \bar{B} = \frac{1}{k}\sum_{i=1}^{k} B_i \quad (3)$$

The aggregated LoRA weights are then distributed to each client, which is updated further locally until the next communication round.

Zero-padding. With heterogeneous rank, *i.e.*, whenever $r_i \neq r_j$ does not hold in general, a conventional strategy is to pad the missing dimensions with zero (Cho et al., 2024). Concretely, one can consider the zero-padded weight matrices

$$\begin{aligned} \tilde{A}_i^\top &= \left[A_i^\top|\mathbf{0}|\mathbf{0}|\cdots|\mathbf{0}\right] \in \mathbb{R}^{n\times r_{\max}} \\ \tilde{B}_i &= \left[B_i|\mathbf{0}|\mathbf{0}|\cdots|\mathbf{0}\right] \in \mathbb{R}^{m\times r_{\max}} \end{aligned} \quad (4)$$

where $r_{\max}$ denotes the maximum rank among all clients. This operation preserves the matrix product $\tilde{B}\tilde{A} = BA$, and thus can be deemed 'harmless.' After matching the dimensionality, one can proceed to aggregate the weight updates as in typical rank-homegeneous federated LoRA (eq. (3)).

High-rank	Round 1		Round 2		Round 3	
	Before	After	Before	After	Before	After
Zero-padding	84.34	38.95	71.58	42.92	86.58	50.53
Replication	84.34	82.11	88.82	86.16	89.47	86.05
Low-rank (avg.)	**Round 1**		**Round 2**		**Round 3**	
	Before	After	Before	After	Before	After
Zero-padding	24.96	23.95	31.07	43.42	45.06	49.11
Replication	24.96	23.95	31.07	44.08	44.48	76.63

Table 1: Comparison of accuracy before and after aggregation, for the high-rank client with a high quality local dataset (top) and the low-rank clients that have low quality local datasets (bottom).

3 Shortcomings of the zero-padding

Our first observation is that the rank-heterogeneous federated LoRA with zero-padding tends to perform worse than rank-homogeneous LoRA, whenever the dataset quality varies significantly over the clients (will be shown later in Section 6, Figure 2). Here, we have varied the dataset quality of each clients by drawing local data from Dirichlet distribution, as in Lin et al. (2021). Here, the client with larger and more balanced datasets are considered of higher quality, as they achieve higher local accuracy during the early training. We have assigned higher ranks to the higher-quality clients.

Why can zero-paddings hurt? We hypothesize that such unexpected underperformance of zero-padding is due to the fact that padded zeros tend to dilute useful information captured by high-quality clients. To see this, consider averaging k weight matrices $A_1, \tilde{A}_2, \ldots, \tilde{A}_k$ where A_1 is of rank r_1 and $\tilde{A}_i$ are of rank $r_2 < r_1$, which is zero-padded with $r_1 - r_2$ all-zero rows. By averaging, the top r_2 rows may retain the same relative scale as the original weight. However, the remaining $r_1 - r_2$ rows may have the relative scale of $1/k$, having their impact on the overall model much diminished as the number of clients grow.

Indeed, our empirical analysis supports this hypothesis; Table 1 compares the accuracy achieved by high-rank clients before and after aggregating the information from low-rank clients. We observe that the accuracy degrades severely after aggregation, suggesting that useful information of the high-rank clients has been lost during aggregation (see Appendix B for more detailed setup).

4 Method: Replication strategy

To address this shortcoming, we develop a very simple yet effective method, called *replication* strategy. Instead of padding all-zero vectors, we replicate the rows and columns from the high-rank clients and append them to low-rank clients (Figure 1).

Concretely, we first consider a simple case where we have one high-rank client and one low-rank client; let $\Delta W_1 = B_1 A_1$ be the high-rank parameter updates from the first client with some rank r_1, and let $\Delta W_2 = B_2 A_2$ be the low rank parameter update from the second client with rank $r_2 < r_1$. Then, the 'row/column-replicated' version of the low rank matrix is given by

$$\begin{aligned} \tilde{A}_2^\top &= \left[A_2^\top \middle| \mathbf{a}_{1,r_2+1}^\top \middle| \cdots \middle| \mathbf{a}_{1,r_1}^\top\right], \\ \tilde{B}_2 &= \left[B_2 \middle| \mathbf{b}_{1,r_2+1} \middle| \cdots \middle| \mathbf{b}_{1,r_1}\right], \end{aligned} \quad (5)$$

where $\mathbf{a}_{1,i}$ and $\mathbf{b}_{1,i}$ denotes the ith row and column vectors of A_1 and B_1, respectively. Then, we can proceed to aggregating the matrices, as in eq. (3). Note that the operations can be done rapidly, thus incurring negligible latency to the overall pipeline.

Whenever there are multiple high rank clients, we handle this in three steps: (1) Aggregate 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.

We emphasize that the overall communication cost remains unchanged. Since the replication process is performed exclusively on the server, we can enjoy the advantages of our method without any additional communication overhead.

Mechanism for allocating high-rank. Instead of manually inspecting local datasets to see which client has a high quality dataset (and thus high rank should be allocated), we adopt a simple loss-based criterion to assign 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, and allocate a high rank.

5 Experimental setup

Datasets. We focus on the text classification, using AG's News (Zhang et al., 2015) and DBpedia (Auer et al., 2007) datasets; we preprocess the DBpedia dataset as in Zhang et al. (2015). We use 10% of the test set for validation, and the rest for testing.

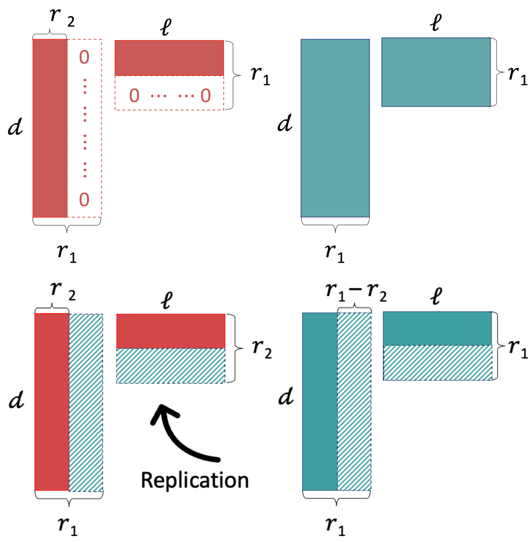

Figure 1: A visual comparison of two strategies for aggregating rank-heterogeneous LoRA updates. Top: Zero-padding. Bottom: Replication (proposed).

Models. We experiment on lightweight BERT-style language models, which are appropriate to be deployed on edge clients: DistilBERT (Sanh et al., 2019), and ALBERT (Lan et al., 2020). For classification, we add an initialized-and-frozen classification layer to these models, as in Sun et al. (2024). We apply LoRA only on self-attention layers, following Hu et al. (2022).

Clients. We employ total 100 clients, and the training dataset is partitioned over these clients without overlap. We model two types of clients: (1) *High-quality* (HQ) clients have balanced local data, i.e., have similar number of samples for each class. (2) *Low-quality* (LQ) clients have datasets with more class imbalance, i.e., have very few samples from certain classes. We randomly select 10% of all clients to be HQ, and the remaining 90% to be LQ. 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; we use the hyperparameter $\alpha = \{5.0, 1.0\}$ for HQ and LQ, respectively. The average number of samples for both HQ and LQ have been set to be equal. At the initial round, we apply $r = 5$ to all clients. After the initial round, we assign $r = 20$ to the top 10% clients that achieve highest validation accuracy.

Training. Following McMahan et al. (2017), we conduct one local epoch training per global round. We randomly select 10% of clients to participate in each global round, ensuring the proportion of high-

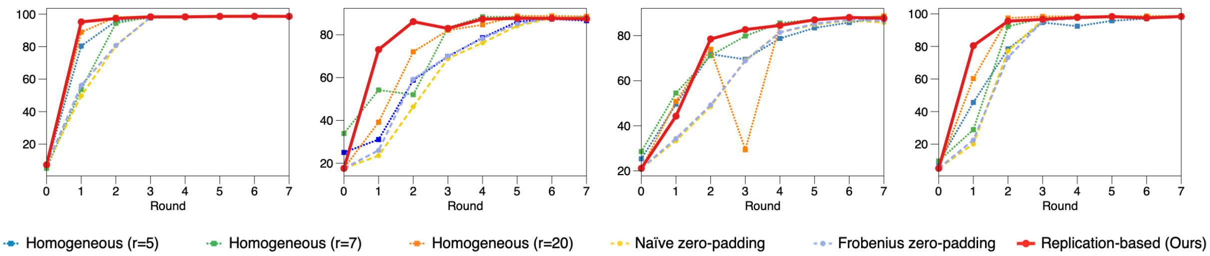

Figure 2: Test accuracy of DistilBERT (left two panels) and ALBERT (right two panels) on the AG's News (first and third) and DBPedia (second and fourth) datasets.

rank clients remains consistent with the overall distribution. We use Adam with the learning rate 5e-4, without any learning rate scheduling.

Baselines. We compare the performance of the proposed replication strategy with three baselines. (1) *Homogeneous*: All clients have a same rank, and thus there is no need for an additional aggregation strategy. We evaluate $r \in \{5, 7, 20\}$, where $r = 7$ has a similar total communication cost with the rank-heterogeneous LoRA; see Table 2 for an explicit communication cost comparison. (2) *Naïve zero-padding* (Cho et al., 2024): Pads zeros to low-rank updates, as described in eq. (4). (3) *Frobenius zero-padding* (Cho et al., 2024): Same as naïve padding, but applies a weighted sum instead of averaging, with weight proportional to the Frobenius norm of the product matrix $\|\Delta W_i\|_F$.

6 Results

The experimental results are given in Figure 2. 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.

DistilBERT. (Left two) We first observe that the proposed replication strategy (red) achieves the fastest convergence over all methods in both cases. In particular, the strategy closely achieves the peak test accuracy in two communication rounds. In terms of the final accuracy, the proposed strategy is also among one of the best, together with the communication-heavy option (homogeneous rank 20; orange) which only slightly outperforms on AG's News. Zero-padding strategies (dotted 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.

	LoRA (r=20)	*LoRA (r=7)*	*Ours*
number of parameters	552,960	193,536	179,715
communication cost	2.11MB	0.74MB	0.69MB
fraction of total model	0.83%	0.30%	0.27%

Table 2: Communication cost comparison on DistilBERT. We compare the communication cost used per client (in average) for transmitting LoRA updates.

ALBERT. (Right two) Similarly, our method achieves a the fastest convergence to the high accuracy, only slightly worse than the communication-heavy case (homogeneous rank 20). 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.

7 Conclusion

We have identified and analyzed the drawbacks of the zero-padding method during the aggregation process when using heterogeneous LoRA in federated fine-tuning of language models and proposed a replication-based padding method to address these issues. We have experimentally demonstrated that this method achieves faster convergence with lower resource usage compared to homogeneous LoRA with high ranks. This suggests that assigning higher ranks to only a limited set of clients—while leaving others with lower ranks—can better align with client resources and data, optimizing overall performance. Additionally, this study focuses on a single high rank and a single low rank, allowing for exploration of multiple ranks to better manage resource and data heterogeneity. 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 learning.

Acknowledgments

This work has been supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00213710, No. RS-2024-00453301).

Limitations

Our approach is based on the assumption that at least one client possesses high-quality data in the federated learning setting. In cases where all clients have data of similarly high quality, the performance gains of our method may be limited. In addition, we have only explored a binary categorization of clients (high-quality, and low-quality), while in practice the client quality can be quite diverse.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Yuhang Yao, Jianyi Zhang, Junda Wu, Chengkai Huang, Yu Xia, Tong Yu, Ruiyi Zhang, Sungchul Kim, Ryan Rossi, Ang Li, et al. 2024. Federated large language models: Current progress and future directions. *arXiv preprint arXiv:2409.15723*.

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

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

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

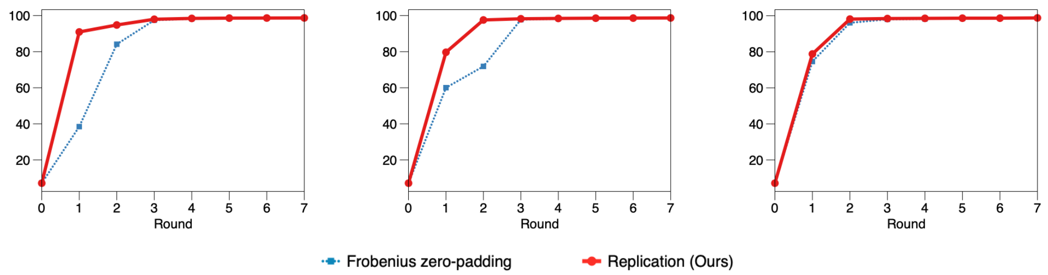

Figure 3: Test accuracy based on the proportion of high-rank clients, with the results shown for 10%, 20%, 50% of high-rank clients from left to right.

A Related work

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 as well. For instance, Kim et al. (2023) studies 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) considers 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. Clients with similar data distribution can have very different importances whenever the amount of data significantly differs, and vice versa when both clients have similar degree of imbalance with different majority classes. 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.

B Experimental setup for Table 1

To establish a simple experimental setup, We conduct the experiments using DistilBERT and AG's News dataset and considered 15 clients. One client ha d a perfectly uniform data distribution, while the remaining clients followed a Dirichlet distribution with $\alpha = 0.6$, the average number of data points from these clients has been kept equal to the number of data points of the client with uniform distribution.

C Additional experiments

For additional discussion, We conduct the experiments using the DistilBERT model and the DBPedia dataset. These experiments focus on examining the effects of rank allocation and varying the proportion of high-rank clients.

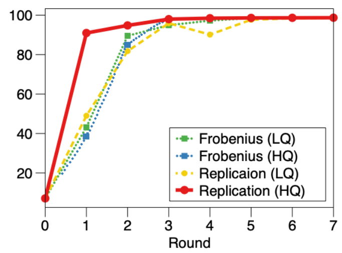

Figure 4: Comparison of model performance based on rank allocation

C.1 Rank allocation

To demonstrate the advantages of assigning high ranks to clients with high-quality data, we conduct an experiment where high ranks were assigned to clients with low-quality data, specifically those in the bottom 10% in the initial round. As shown in Figure 4, We observe that assigning high ranks to low-quality clients did not result in better performance than even simple Frobenius zero-padding. This suggests that copying the weights of models trained on imbalanced data offers limited benefits.

C.2 Proportion of high rank clients

To compare results based on the high-rank client ratio, we conduct experiments with high-rank client ratios set at 10%, 20%, and 50%. The results can be seen in Figure 3. As the high-rank client ratio increases, the performance gap with Frobenius zero-padding diminishes. This trend can be interpreted as the disadvantage of diluting high-rank information being offset by the reduction in replicated weights. However, it is important to note that as the proportion of high-rank clients increases, more resources are required.

D Other experimental details

All experiments were executed on a single NVIDIA RTX A6000 GPU without distributed training. The graphs within the figure were generated using a single fixed random seed for consistency.