Robust Federated Finetuning of Foundation Models via Alternating Minimization of LoRA

Shuangyi Chen [1] **Yue Ju** [2] **Hardik Dalal** [2] **Zhongwen Zhu** [2] **Ashish Khisti** [1]

Abstract

Parameter-Efficient Fine-Tuning (PEFT) has risen as an innovative training strategy that updates only a select few model parameters, significantly lowering both computational and memory demands. PEFT also helps to decrease data transfer in federated learning settings, where communication depends on the size of updates. In this work, we explore the constraints of previous studies that integrate a well-known PEFT method named LoRA with federated fine-tuning, then introduce RoLoRA, a robust federated fine-tuning framework that utilizes an alternating minimization approach for LoRA, providing greater robustness against decreasing fine-tuning parameters and increasing data heterogeneity. Our results indicate that RoLoRA not only presents the communication benefits but also substantially enhances the robustness and effectiveness in multiple federated fine-tuning scenarios.

1. Introduction

The recent emergence of foundation models in various applications significantly changes the field of machine learning. Characterized by their broad adaptability and massive scale, these models require access to vast and diverse datasets to effectively learn across different tasks and domains. However, this presents a significant challenge: foundation models not only require large amounts of data, but also data of high quality. Federated learning provides a promising solution to this issue. It enables the use of data from multiple sources while protecting the privacy of the data. By combining insights from different decentralized sources, federated learning allows for collaborative model training without exposing sensitive information. This method is especially beneficial for foundation models, as it can access a broad range of data while maintaining privacy.

Recently, Parameter-Efficient Fine-Tuning (PEFT) has emerged as an innovative training strategy that updates only a small subset of model parameters, substantially reducing computational and memory demands. A notable method in this category is LoRA (Hu et al., 2021), which utilizes low-rank matrices to approximate weight changes during fine-tuning. These matrices are integrated with pre-trained weights for inference, facilitating reduced data transfer in scenarios such as federated learning, where update size directly impacts communication efficiency. Many works integrate LoRA into federated setting. For example, FedPETuning (Zhang et al., 2023d) compared various PEFT methods in a federated setting. SLoRA (Babakniya et al., 2023), a hybrid approach that combines sparse fine-tuning with LoRA, is introduced to tackle data heterogeneity in federated settings. Furthermore, FS-LLM (Kuang et al., 2023) is presented, which is a framework for fine-tuning LLMs in federated environments. However, these studies typically apply the FedAVG algorithm directly to LoRA modules, overlooking the interference introduced by this aggregation approach. With this consideration, Sun et al. designs a federated finetuning framework named FFA-LoRA (Sun et al., 2024) based on LoRA by freezing down-projection matrix $\mathbf{A}$ for all the clients and only updating up-projection matrix $\mathbf{B}$. Furthermore, they apply DP-SGD to preserve privacy. Using sufficient number of finetuning parameters, FFA-LoRA with a larger learning rate achieves performance comparable to FedAVG for LoRA modules while halving the communication costs. However, we observe that with fewer fine-tuning parameters, FFA-LoRA is less robust than FedAVG for LoRA modules, primarily due to its limited expressiveness stemming from the restricted number of trainable parameters. Another common issue in federated learning is data heterogeneity among clients. To address this, we drew inspiration from the personalized federated framework FedRep (Collins et al., 2021), which alternates between updating clients' representation and head. This approach highlights the importance of learning a robust low-rank representation and demonstrates superior convergence speed compared to simultaneously updating both represen-

[1]ECE Department, University of Toronto, Toronto, Canada [2]Ericsson-GAIA Montréal, Canada. Correspondence to: Shuangyi Chen <shuangyi.chen@mail.utoronto.ca>, Ashish Khisti <akhisti@ece.utoronto.ca>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

tation and head. Therefore, we propose a robust federated fine-tuning framework, RoLoRA, based on alternating minimization of LoRA. Empirical evidence demonstrates that RoLoRA is more robust against **Decreasing Fine-tuning Parameters** and **Increasing Data Heterogeneity**, while still halving communication costs, similar to FFA-LoRA.

Related Work We provide a summary of the literature on PEFT, Variants of LoRA, PEFT in Federated Setting, and FL with data heterogeneity in Appendix A.

2. Preliminaries

2.1. Low-Rank Adaptation: LoRA

Low-Rank Adaptation (LoRA) (Hu et al., 2021) fine-tunes large language models efficiently by maintaining the original model weights fixed and adding small, trainable matrices in each layer. These matrices perform low-rank decompositions of updates, reducing the number of trainable parameters. This approach is based on the finding that updates to model weights during task-specific tuning are usually of low rank, which allows for fewer parameters to be adjusted. For example, for a pre-trained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{d\times d}$, the update is a low-rank product $\mathbf{BA}$, where $\mathbf{A} \in \mathbb{R}^{r\times d}$ and $\mathbf{B} \in \mathbb{R}^{d\times r}$, with $r << d$. Only $\mathbf{A}$ and $\mathbf{B}$ are trainable, allowing $\mathbf{W} = \mathbf{W}_0 + \alpha\mathbf{BA}$, with α adjusting the update's impact. Applying LoRA in a federated setting is a practical choice. By using LoRA adapters, clients can fine-tune foundation models efficiently with limited resources. Since only these specific matrices need to be transmitted to a central server, this approach significantly reduces communication costs. This makes LoRA an advantageous solution for enhancing model performance in collaborative scenario comparing to full parameter finetuning in the federated setting.

2.2. FedAVG of LoRA Introduces Interference

Integrating LoRA within a federated setting presents challenges. In such a setup, each of the N clients is provided with the pretrained model weights $\mathbf{W}_0$, which remain fixed during finetuning. Clients are required only to send the updated matrices $\mathbf{B}_i$ and $\mathbf{A}_i$ to a central server for aggregation. While most current studies, such as SLoRA(Babakniya et al., 2023) and FedPETuning(Zhang et al., 2023d), commonly apply FedAVG directly to these matrices as shown in (2), this approach might not be optimal. The precise update for each client's model, $\Delta\mathbf{W}_i$, should be calculated as the product of the low-rank matrices $\mathbf{A}_i$ and $\mathbf{B}_i$. Consequently, aggregation on the individual matrices introduces interference.

$$\frac{1}{N}\sum_{i=1}^{N}\Delta\mathbf{W}_i = \frac{1}{N}(\mathbf{B}_1\mathbf{A}_1 + \mathbf{B}_2\mathbf{A}_2 + ... + \mathbf{B}_N\mathbf{A}_N) \quad (1)$$

$$\neq \frac{1}{N}(\mathbf{B_1} + \mathbf{B_2} + ... + \mathbf{B_N})\frac{1}{N}(\mathbf{A_1} + \mathbf{A_2} + ... + \mathbf{A_N}) \quad (2)$$

2.3. FedRep: Common Representation via Alternating Minimization

A common challenge in federated learning is data heterogeneity among clients. FedRep (Collins et al., 2021) addresses this by finding a common representation, which is effectively achieved via alternatively updating clients' representation and head, aggregating only representation while keeping the head diverse. The algorithm demonstrates the necessity of learning a robust low-rank representation. Additionally, the alternating optimization has shown superior convergence speed compared to approaches that simultaneously update both representation and head. We observe a structured similarity between the LoRA adapter and the representation-head structure. Specifically, we consider the down-projection matrix $\mathbf{A}$ in each LoRA adapter as the low-rank representation for the features of the intermediate layers. We hypothesize that learning a robust low rank representation (down-projection matrix) is also advantageous for the intermediate features when the clients has heterogeneous inputs. However, since the LoRA adapters are cascaded in the model unlike single representation-head structure in model considered in FedRep, keeping up-projection matrix $\mathbf{B}$ diverse may not be favorable for convergence.

With these considerations, we propose robust federated fine-tuning framework based on alternating minimization of LoRA (RoLoRA).

3. Our Framework

We describe the framework design of RoLoRA and discuss its practical advantages.

Alternating Minimization and Corresponding Aggregation Motivated by the observations discussed in Section 2.2 and 2.3, we propose applying alternating minimization to the local fine-tuning of each client in a setting with N clients. Unlike the approach in FFA-LoRA, where $\mathbf{A}$ is consistently frozen, we suggest a alternating update strategy. There are alternating odd and even communication rounds designated for updating, aggregating $\mathbf{A}$ and $\mathbf{B}$, respectively.

$$\text{In the odd comm. round:} \quad \frac{1}{N}\sum_{i=1}^{N}\Delta\mathbf{W}_i^{2t+1}$$

$$= \frac{1}{N}(\mathbf{B}_1^{t+1}\mathbf{A}_1^{t} + \mathbf{B}_2^{t+1}\mathbf{A}_2^{t} + ... + \mathbf{B}_N^{t+1}\mathbf{A}_N^{t}) \quad (3)$$

$$= \frac{1}{N}(\mathbf{B}_1^{t+1} + \mathbf{B}_2^{t+1} + ... + \mathbf{B}_N^{t+1})\mathbf{A}^{t}$$

$$\text{In the even comm. round:} \quad \frac{1}{N}\sum_{i=1}^{N}\Delta\mathbf{W}_i^{2t+2}$$

$$= \frac{1}{N}(\mathbf{B}_1^{t+1}\mathbf{A}_1^{t+1} + \mathbf{B}_2^{t+1}\mathbf{A}_2^{t+1} + ... + \mathbf{B}_N^{t+1}\mathbf{A}_N^{t+1}) \quad (4)$$

$$= \frac{1}{N}\mathbf{B}^{t+1}(\mathbf{A}_1^{t+1} + \mathbf{A}_2^{t+1} + ... + \mathbf{A}_N^{t+1})$$

In the odd communication round, all clients freeze $\mathbf{A}^t$ and update $\mathbf{B}^t$. The central server then aggregates these updates to compute $\mathbf{B}^{t+1} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{B}_i^{t+1}$ and distributes $\mathbf{B}^{t+1}$ back to the clients. In the subsequent communication round, clients freeze $\mathbf{B}^{t+1}$ and update $\mathbf{A}^t$. The server aggregates these to obtain $\mathbf{A}^{t+1} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{A}_i^{t+1}$ and returns $\mathbf{A}^{t+1}$ to the clients. It is important to note that in round $2t+1$, the frozen $\mathbf{A}_i^t$ are identical across all clients, as they are synchronized with $\mathbf{A}^t$ from the central server at the beginning of the round. This strategy ensures that the update and aggregation method introduces no interference, as demonstrated in (3) and (4).

Computation and Communication Cost The parameter-freezing nature of RoLoRA enhances computational and communication efficiency. In each communication round, the number of trainable parameters in the model is effectively halved compared to FedAVG with LoRA. The only additional cost for RoLoRA compared to FFA-LoRA is the alternating freezing of the corresponding parameters. We remark this additional cost is negligible because it is applied to the clients' models and can be executed concurrently during the server's aggregation.

4. Experiments

We evaluate the performance of RoLoRA in various federated settings. We use NVIDIA GeForce RTX 4090 or NVIDIA A40 for all the experiments.

Baselines Considering cross-silo federated setting where the number of clients is relatively small and all clients will participate in each round, we will explore the following three methods based on FedAVG.

- **LoRA** means LoRA adapter and its finetuning algorithm are directly applied to local finetuning of clients in the federated system. Specifically, in iteration t, the server receives $\mathbf{A}_i^t$ and $\mathbf{B}_i^t$ from client i and aggregates by $\mathbf{A}^t = \mathsf{Avg}(\mathbf{A}_i^t)$ and $\mathbf{B}^t = \mathsf{Avg}(\mathbf{B}_i^t)$.
- **LoRA-FFA** (Sun et al., 2024) is a baseline that enable the clients to finetune $\mathbf{B}$ and keep $\mathbf{A}$ frozen locally. Thus, in iteration t, the server aggregates by $\mathbf{B}^t = \mathsf{Avg}(\mathbf{B}_i^t)$.
- **RoLoRA** enables clients to alternate updating $\mathbf{A}$ and $\mathbf{B}$ as described in Section 3.

Model and Datasets. We take the pre-trained RoBERTa-Large (355M) (Liu et al., 2019) models from the HuggingFace Transformers library. and evaluate the performance of three federated finetuning methods on 5 datasets (SST-2, QNLI, MNLI, QQP, RTE) from the GLUE (Wang et al., 2019). Due to the limitation of the unpublished test set in GLUE, we follow the previous studies (Zhang et al., 2023d) and use the original validation set as the new test set and split a part of the training set as the validation set.

Implementation. We implement all the methods based on FederatedScope-LLM (Kuang et al., 2023). To make a fair comparison, for each dataset, we obtain the best performance on test set and report the average over five seeds. Specifically, the learning rate is chosen from the set $\{5e-4, 1e-3, 2e-3, 5e-3, 1e-2, 2e-2, 5e-2, 1e-1, 2e-1\}$. Other hyper-parameters for experiments are specified in Table 4 in Appendix B.1.

Effect of Number of Finetuning Parameters In Figure 1, we compare three methods across five GLUE datasets. We apply LoRA to every weight matrix of the selected layers, given different budgets of LoRA parameters. For each dataset, we experiment with three budgets, ranging from high to low. The corresponding layer sets, $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$, are detailed in Table 5 in Appendix B.1.

The figures indicates that with sufficient number of finetuning parameters, the three methods can achieve comparable best accuracy; as the number of LoRA parameters is reduced, the performance of the three methods deteriorates to varying degrees. However, RoLoRA, which achieves performance comparable to LoRA, demonstrates greater robustness compared to FFA-LoRA, especially under conditions of limited fine-tuning parameters. It is important to note that with the same finetuning parameters, the communication cost of RoLoRA and FFA-LoRA is always half of that of LoRA due to their parameter freezing nature. This implies that RoLoRA not only sustains its performance but also enhances communication efficiency. We expand the middle set of data of each of Figure 1, corresponding to $\mathcal{P}_2$, and show the details of the performance of three methods in Table 1.

Effect of Data Heterogeneity In this section, we study the effect of data heterogeneity. The layer set with LoRA adapters in Table 2 is $\mathcal{P}_2$ as in Table 1. In Table 2, we increased the number of clients from 3 to 20, and then to 50, ensuring that there is no overlap in the training samples each client can access. Consequently, each client receives a smaller fraction of the total dataset, leading to a rise in data heterogeneity among the clients. We observe that as the data heterogeneity increases, while maintaining the same number of fine-tuning samples, the performance of the LoRA method significantly deteriorates for most datasets. In contrast, RoLoRA maintains its accuracy levels. The performance of FFA-LoRA also declines, attributed to the limited expressiveness of the random initialization of $\mathbf{A}$ for clients' heterogeneous data. Notably, RoLoRA achieves this accuracy while incurring only half the communication

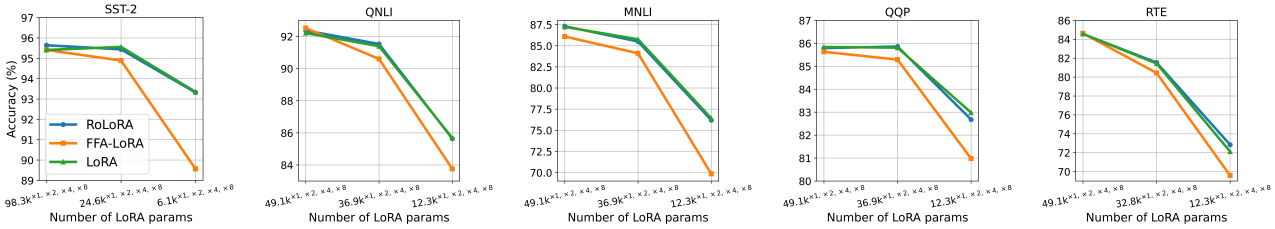

Figure 1. Results with RoBERTa-Large models on GLUE of different budget of finetuning parameters. The accuracy is computed by averaging over different ranks $\{1, 2, 4, 8\}$. The number of clients is 3.

Rank	Methods	Comm. cost	SST-2	QNLI	MNLI	QQP	RTE	Avg.
r=8	LoRA	×16	$95.68_{\pm 0.14}$	$91.46_{\pm 0.30}$	$85.93_{\pm 0.01}$	$85.95_{\pm 0.18}$	$81.35_{\pm 0.74}$	88.07
	FFA-LoRA	×8	$94.99_{\pm 0.10}$	$91.09_{\pm 0.36}$	$85.21_{\pm 0.03}$	$85.76_{\pm 0.08}$	$80.14_{\pm 1.02}$	87.44
	RoLoRA	×8	$95.45_{\pm 0.14}$	$91.84_{\pm 0.09}$	$85.76_{\pm 0.01}$	$85.91_{\pm 0.22}$	$81.32_{\pm 0.78}$	88.06
r=4	LoRA	×8	$95.62_{\pm 0.17}$	$91.59_{\pm 0.21}$	$86.20_{\pm 0.05}$	$86.13_{\pm 0.10}$	$81.46_{\pm 1.22}$	88.20
	FFA-LoRA	×4	$95.18_{\pm 0.09}$	$91.35_{\pm 0.32}$	$84.58_{\pm 0.21}$	$85.50_{\pm 0.25}$	$81.10_{\pm 0.33}$	87.48
	RoLoRA	×4	$95.49_{\pm 0.16}$	$91.64_{\pm 0.30}$	$85.70_{\pm 0.04}$	$86.14_{\pm 0.06}$	$82.43_{\pm 0.84}$	88.28
r=2	LoRA	×4	$95.64_{\pm 0.11}$	$92.04_{\pm 0.11}$	$85.85_{\pm 0.19}$	$86.16_{\pm 0.08}$	$82.19_{\pm 1.03}$	88.38
	FFA-LoRA	×2	$94.91_{\pm 0.16}$	$90.11_{\pm 0.17}$	$84.06_{\pm 0.19}$	$85.48_{\pm 0.01}$	$80.86_{\pm 0.51}$	87.08
	RoLoRA	×2	$95.60_{\pm 0.10}$	$91.62_{\pm 0.32}$	$85.55_{\pm 0.05}$	$86.16_{\pm 0.18}$	$82.19_{\pm 1.03}$	88.22
r=1	LoRA	×2	$95.32_{\pm 0.18}$	$90.48_{\pm 0.56}$	$85.08_{\pm 0.04}$	$85.01_{\pm 0.05}$	$81.10_{\pm 0.95}$	87.40
	FFA-LoRA	×1	$94.49_{\pm 0.22}$	$89.87_{\pm 0.37}$	$82.60_{\pm 0.03}$	$84.42_{\pm 0.50}$	$79.66_{\pm 1.08}$	86.21
	RoLoRA	×1	$95.22_{\pm 0.14}$	$91.01_{\pm 0.23}$	$84.97_{\pm 0.05}$	$85.24_{\pm 0.18}$	$80.23_{\pm 1.02}$	87.33

Table 1. Results with RoBERTa-Large models on GLUE. We report the average and std. over five seeds. The number of clients is 3. Please refer to Table 6 in Appendix B.2.2 for the actual communication cost.

		SST-2	QNLI	MNLI	QQP	RTE
iid.	LoRA	95.64	92.04	85.85	86.16	82.19
	FFA-LoRA	94.91	90.11	84.06	85.48	80.86
	RoloRA	95.60	91.62	85.66	86.16	82.19
mild het.	LoRA	94.27	86.91	81.22	82.07	46.21
	FFA-LoRA	93.92	89.58	80.51	82.62	57.76
	RoloRA	94.84	90.77	85.13	85.10	81.23
severe het.	LoRA	93.23	82.57	58.96	76.96	49.10
	FFA-LoRA	92.32	85.15	62.79	77.78	53.07
	RoloRA	94.61	89.83	85.15	85.55	72.92

Table 2. Results with RoBERTa-Large models with varying client numbers (3, 20, 50) to increase data heterogeneity in federated setting, maintaining a constant sample count during fine-tuning. The rank used is 2.

Model/Rank	Methods	QNLI	MNLI	QQP	RTE
Rob-L$_{r=4}$	LoRA	86.07	76.58	83.77	73.58
	FFA-LoRA	88.54	78.28	84.04	74.64
	RoLoRA	89.13	82.33	84.58	76.51
Rob-L$_{r=2}$	LoRA	85.47	76.26	82.32	69.68
	FFA-LoRA	87.81	77.24	83.81	76.11
	RoLoRA	89.19	82.18	84.24	76.53

Table 3. Results with RoBERTa-Large models on GLUE. Same message size in each round for three methods for each dataset. The number of clients is 3.

costs associated with LoRA. Figure 2 in Appendix B.2.2 illustrates the dynamics during fine-tuning for three methods, highlighting that the convergence speed of RoLoRA is substantially better than that of the other two methods.

Align Communication Cost for Three Methods In Table 3, we conduct a comparison of three methods under the constraint of identical communication costs under the assumption that the number of clients is small. To align the communication costs across these methods, two approaches are considered. The first approach involves doubling the rank of FFA-LoRA and RoLoRA, with the outcomes detailed in Table 1. The second approach requires doubling the number of layers equipped with LoRA adapters. In the results presented in Table 3, the latter strategy is employed. Specifically, for both FFA-LoRA and RoLoRA, we adjust the communication costs by doubling the number of layers equipped with LoRA adapters, compared to the baseline LoRA method, where the layer set $\mathcal{P}_3$ are attached with adapters, thereby standardizing the size of the transmitted messages. Table 3 demonstrates that when operating within a constrained communication cost budget, the performance of RoLoRA consistently surpasses that of the other two methods.

More experimental results with different models and settings are provided in Appendix B.2.

5. Conclusion

In this work, we introduce RoLoRA, a robust federated fine-tuning framework using alternating minimization for LoRA. RoLoRA improves robustness against reduced fine-tuning parameters and increased data heterogeneity. Our results show that RoLoRA enhances communication efficiency, robustness, and effectiveness in various federated fine-tuning settings.

References

Babakniya, S., Elkordy, A., Ezzeldin, Y., Liu, Q., Song, K.-B., EL-Khamy, M., and Avestimehr, S. SLoRA: Federated parameter efficient fine-tuning of language models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*, 2023. URL https://openreview.net/forum?id=06quMTmtRV.

Chen, H.-Y. and Chao, W.-L. On bridging generic and personalized federated learning for image classification, 2022.

Collins, L., Hassani, H., Mokhtari, A., and Shakkottai, S. Exploiting shared representations for personalized federated learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2089–2099. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/collins21a.html.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms, 2023.

Hayou, S., Ghosh, N., and Yu, B. Lora+: Efficient low rank adaptation of large models, 2024.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models, 2021.

Kopiczko, D. J., Blankevoort, T., and Asano, Y. M. VeRA: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NjNfLdxr3A.

Kuang, W., Qian, B., Li, Z., Chen, D., Gao, D., Pan, X., Xie, Y., Li, Y., Ding, B., and Zhou, J. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning, 2023.

Li, Y., Yu, Y., Liang, C., He, P., Karampatziakis, N., Chen, W., and Zhao, T. Loftq: Lora-fine-tuning-aware quantization for large language models, 2023.

Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H. Dora: Weight-decomposed low-rank adaptation, 2024.

Liu, X., Ji, K., Fu, Y., Tam, W. L., Du, Z., Yang, Z., and Tang, J. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks, 2022.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach, 2019.

Oh, J., Kim, S., and Yun, S.-Y. FedBABU: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=HuaYQfggn5u.

Sun, Y., Li, Z., Li, Y., and Ding, B. Improving loRA in privacy-preserving federated learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NLPzL6HWNl.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019.

Xu, J., Tong, X., and Huang, S.-L. Personalized federated learning with feature alignment and classifier collaboration. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SXZr8aDKia.

Zaken, E. B., Ravfogel, S., and Goldberg, Y. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models, 2022.

Zhang, H., Li, C., Dai, W., Zou, J., and Xiong, H. FedCR: Personalized federated learning based on across-client common representation with conditional mutual information regularization. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 41314–41330. PMLR, 23–29 Jul 2023a. URL https://proceedings.mlr.press/v202/zhang23w.html.

Zhang, L., Zhang, L., Shi, S., Chu, X., and Li, B. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning, 2023b.

Zhang, Q., Chen, M., Bukharin, A., Karampatziakis, N., He, P., Cheng, Y., Chen, W., and Zhao, T. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning, 2023c.

Zhang, Z., Yang, Y., Dai, Y., Wang, Q., Yu, Y., Qu, L., and Xu, Z. FedPETuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 9963–9977, Toronto, Canada, July 2023d. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.632. URL https://aclanthology.org/2023.findings-acl.632.

A. Related Works

A.1. Parameter Efficient Fine Tuning (PEFT): LoRA and Its Variants

As the size of large language models (LLMs) continues to increase, it is computationally expensive and time-consuming to finetune to the full model. Parameter efficient finetuning (PEFT) allows for updates to a smaller subset of parameters, significantly reducing the computational and memory requirements. One of the most well-known methods is LoRA(Hu et al., 2021). LoRA uses low-rank matrices to approximate changes in weights during fine-tuning, allowing them to be integrated with pre-trained weights before inference. Based on LoRA, many PEFT methods are developed. For example, Zhang et al. (Zhang et al., 2023c) designs AdaLoRA by using SVD decomposition and pruning less significant singular values for more efficient updates. VeRA (Kopiczko et al., 2024) is proposed to further reduce the number of trainable parameters during finetuning by using a single pair of low-rank matrices shared across all layers and learning small scaling vectors. Zhang et al. (Zhang et al., 2023b) proposes a memory-efficient fine-tuning method named LoRA-FA which keeps the projection-down weight of $\mathbf{A}$ fixed and updates the projection-up weight of $\mathbf{B}$ during finetuning. Hayou et al. (Hayou et al., 2024) enhance LoRA by assigning different learning rates to $\mathbf{A}$ and $\mathbf{B}$, theoretically confirming that the optimal approach requires a higher learning rate for $\mathbf{B}$ than for $\mathbf{A}$. Liu et al. analyze magnitude and directional updates in LoRA versus full parameter fine-tuning and introduce DoRA(Liu et al., 2024), which decomposes pre-trained weights for fine-tuning and applies LoRA for directional updates. A quantized version of LoRA named QLoRA(Dettmers et al., 2023) is introduced. Building upon that, Li et al. develops LoftQ (Li et al., 2023) for a better initialization for quantized training.

A.2. PEFT in Federated Setting

PEFT adjusts only a few lightweight or a small portion of the total parameters for specific tasks, keeping most foundational model parameters unchanged. This feature can help reduce data transfer in federated learning, where communication depends on the size of updates. Zhang et al. (Zhang et al., 2023d) compares multiple PEFT methods in federated setting, including Adapter(Houlsby et al., 2019), LoRA(Hu et al., 2021), Prompt tuning(Liu et al., 2022) and Bit-Fit(Zaken et al., 2022). SLoRA(Babakniya et al., 2023), which combines sparse finetuning and LoRA, is proposed by Babakniya et al. to address the data heterogeneity in federated setting. Sun et al. designs a federated finetuning framework named FFA-LoRA based on LoRA (Sun et al., 2024) by freezing matrix $\mathbf{A}$ for all the clients and only updating matrix $\mathbf{B}$. Furthermore, they apply DP-SGD to preserve privacy. FS-LLM (Kuang et al., 2023), a framework for finetuning LLM, is introduced.

A.3. FL with Data Heterogeneity

FL with a Common Representation FL with a common representation aims to address the challenges of data heterogeneity in FL. Those FL methods learn a shared global representation while allowing each client to have its own personalized partial model. Works include FedRep(Collins et al., 2021), which learns a global low-dimensional representation and personalized head for each client, FedCR (Zhang et al., 2023a), which introduces a regularizer to encourage learning a shared representation, and FedPAC (Xu et al., 2023), which performs class-wise feature alignment. Other methods like FedBABU (Oh et al., 2022) and FedRoD (Chen & Chao, 2022) also aim to learn a shared representation across clients. Although we focus on a common model in the federated setting in this work, we got inspired by training algorithm introduced by FedRep (Collins et al., 2021) to learn a low-rank representation for the intermediate features. We discuss the similarity and difference between the LoRA adapter and representation-head structure.

B. Experiments

B.1. Setup

We show the hyper-parameter configurations for each dataset in Table 4.

	SST-2	QNLI	MNLI	QQP	RTE
Total comm. rounds	500	500	500	500	200
Batch Size	64	32	32	32	32
Local Epochs	20	20	20	20	20

Table 4. Hyper-parameters configurations

	Layer Attributes	SST-2	QNLI	MNLI	QQP	RTE
$\mathcal{P}_1$	Type	W_v, W_q	W_v, W_q	W_v, W_q	W_v, W_q	W_v, W_q
	Index	$\{0, \ldots, 23\}$	$\{12, \ldots, 23\}$	$\{12, \ldots, 23\}$	$\{12, \ldots, 23\}$	$\{12, \ldots, 23\}$
$\mathcal{P}_2$	Type	W_v, W_q	W_v, W_q	W_v, W_q	W_v, W_q	W_v, W_q
	Index	$\{18, \ldots, 23\}$	$\{15, \ldots, 23\}$	$\{15, \ldots, 23\}$	$\{15, \ldots, 23\}$	$\{16, \ldots, 23\}$
$\mathcal{P}_3$	Type	W_v	W_v, W_q	W_v, W_q	W_v, W_q	W_v, W_q
	Index	$\{21, \ldots, 23\}$	$\{21, \ldots, 23\}$	$\{21, \ldots, 23\}$	$\{21, \ldots, 23\}$	$\{21, \ldots, 23\}$

Table 5. The selected layer set attached with LoRA

In Table 5, we include the details about layers attached with LoRA adapters for different budget of finetuning parameters, for each dataset.

B.2. More Results

B.2.1. Communication Cost

In Table 6, we show the uplink communication cost for three methods for the layer set $\mathcal{P}_2$ using rank=1.

Rank	Methods	SST-2	QNLI	MNLI	QQP	RTE
r=1	LoRA	46.9	93.8	93.8	140.6	125
	FFA-LoRA	23.5	46.9	46.9	70.3	62.5
	RoLoRA	23.5	46.9	46.9	70.3	62.5

Table 6. Uplink message size (KB) in each communication round for experiments in Table 1 when rank=1. The message size will proportionally increase if utilizing different ranks.

B.2.2. Finetuning Dynamics of the Setup with Severe Data Heterogeneity

In Figure 2, we show the convergence of three methods under severe data heterogeneity with 50 clients. RoLoRA demonstrates superior convergence speed compared to the other two methods.

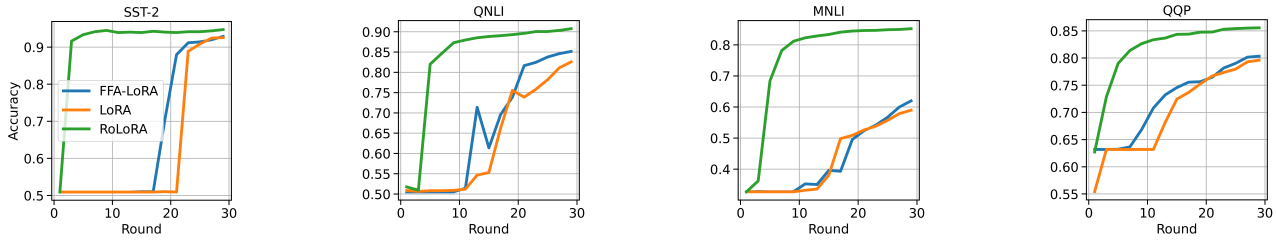

Figure 2. Accuracies over rounds with RoBERTa-Large models on SST-2, QNLI, MNLI, and QQP. The total number of clients is 50.

B.2.3. DeBERTa-XLarge Results

In Table 7, we show the results with DeBERTa-XLarge (900M). For both FFA-LoRA and RoLoRA, we modify the communication costs by equipping twice as many layers with LoRA adapters compared to the standard LoRA method. So the communication costs are aligned for three methods.

Model/Rank	Methods	QNLI	MNLI	QQP	RTE
Deb-XL$_{r=4}$	LoRA	90.16	83.58	85.67	74.97
	FFA-LoRA	90.08	83.31	85.79	78.73
	RoLoRA	91.36	84.63	86.54	81.48
Deb-XL$_{r=2}$	LoRA	90.12	83.25	84.56	72.55
	FFA-LoRA	90.28	83.47	85.59	79.06
	RoLoRA	91.32	84.84	86.50	81.34

Table 7. Results with DeBERTa-XLarge models on GLUE. Same message size in each round for three methods for each dataset. The number of clients is 3.

B.2.4. Performance when QLoRA is Applied

In Table 8, we quantize the frozen pre-trained weights to 8 bit and 4 bit for each client, and apply QLoRA(Dettmers et al., 2023). The relative accuracy is computed as $\frac{\mathsf{Acc}_{\mathsf{FP}}-\mathsf{Avg}(\mathsf{Acc}_{8b}+\mathsf{Acc}_{4b})}{\mathsf{Acc}_{\mathsf{FP}}}$. We use sufficient finetuning parameters and the selected layer set is $\mathcal{P}_1$ to study the effect of the quantized foundation model in the federated setting.

Methods	QQP			QNLI			MNLI			
	FP	8-bit	4-bit	FP	8-bit	4-bit	FP	8-bit	4-bit	Relative Acc.
LoRA	85.86	85.78	85.47	92.87	92.71	92.00	87.69	86.57	86.49	$\downarrow 0.7\%$
FFA-LoRA	85.74	85.59	85.35	92.51	91.20	91.12	85.75	84.86	83.19	$\downarrow 1.3\%$
RoLoRA	85.64	85.51	85.47	92.48	92.51	91.63	87.95	87.27	86.12	$\downarrow 0.7\%$

Table 8. Results of RoBERTa-Large models on QQP, QNLI, MNLI with the quantized base model. The rank used is 2.