Fed-SB: A *Silver Bullet* for Extreme Communication Efficiency and Performance in (Private) Federated LoRA Fine-Tuning

Raghav Singhal [*1] **Kaustubh Ponkshe** [*1] **Rohit Vartak** [2] **Lav R. Varshney** [3] **Praneeth Vepakomma** [1 4]

Abstract

Low-Rank Adaptation (LoRA) is widely used for efficient fine-tuning, but federated settings pose challenges due to suboptimal adapter averaging. We propose **Federated Silver Bullet (Fed-SB)**, a scalable and communication-efficient method for federated fine-tuning based on LoRA-SB, which introduces a small learnable matrix R between frozen adapters. By directly averaging R, Fed-SB enables exact aggregation and decouples communication cost from the number of clients. It achieves **state-of-the-art performance** across commonsense reasoning, arithmetic reasoning, and language inference tasks while reducing communication costs by up to **230x**. Fed-SB is especially well-suited for private settings, reducing trainable parameters and avoiding noise amplification. Our code is available at: https://github.com/CERT-Lab/fed-sb.

1. Introduction

Large language models (LLMs) excel across diverse tasks (Achiam et al., 2023; Touvron et al., 2023; Team et al., 2023). Fine-tuning (FT) is the most effective way to align LLMs to specific data, but full FT is computationally expensive at scale. Parameter-efficient fine-tuning (PEFT) methods like LoRA (Hu et al., 2021) address this by offering a balance between efficiency and performance.

Federated learning (FL) enables model training on siloed data without sharing raw information (Konečný et al., 2017; Kairouz et al., 2021; Bonawitz et al., 2019). Federated fine-tuning (FT) adapts large pre-trained models to private, distributed datasets. Most approaches use LoRA-based client adaptations (Zhang et al., 2024b), but must balance model performance (Sun et al., 2024) and communication efficiency (Wang et al., 2024; Singhal et al., 2025), requiring careful tradeoff management.

LoRA-SB (Ponkshe et al., 2024) simulates full fine-tuning in low-rank space by learning an $r \times r$ matrix between fixed adapters $\mathbf{A}$ and $\mathbf{B}$, reducing parameters and improving updates. It achieves 2–4× higher effective rank than LoRA with **45–90×** fewer parameters. We introduce **Fed-SB**, a federated variant of LoRA-SB that enables exact aggregation via simple averaging of the matrix $\mathbf{R}$, offering a highly efficient solution for (private) federated fine-tuning.

Differential privacy (DP) is essential in federated settings (Dwork, 2006; Dwork et al., 2014). While DP-SGD (Abadi et al., 2016) is widely used, its noise can amplify model divergence in federated fine-tuning (Sun et al., 2024). Fed-SB improves performance by reducing learnable parameters, thereby limiting noise injection. Additionally, it avoids the noise amplification seen in other methods, enhancing privacy-preserving learning.

Fed-SB pushes the performance vs communication cost Pareto frontier, offering an extremely efficient and scalable solution for both private and non-private federated FT, as shown in Figure 1. It consistently has superior performance while substantially reducing communication overhead than other methods. Our key contributions are:

- We propose **Fed-SB**, a federated fine-tuning method that achieves exact and optimal aggregation in low-rank adaptation without incurring prohibitive communication costs or performance degradation.
- Fed-SB consistently achieves **state-of-the-art** results while significantly reducing communication cost, by up to **230x**, by requiring only an $r \times r$ matrix to be transmitted.
- We demonstrate that Fed-SB is particularly well-suited for privacy-preserving (federated) fine-tuning, as it minimizes noise by reducing the number of learnable parameters and leveraging linearity in the aggregate update.
- Extensive experiments show that Fed-SB consistently outperforms existing methods while drastically reducing communication overhead in both private and non-private federated settings, establishing a new Pareto frontier in

[*]Equal contribution [1]Mohamed bin Zayed University of Artificial Intelligence, UAE [2]Duke University, USA [3]University of Illinois Urbana-Champaign, USA [4]Massachusetts Institute of Technology, USA. Correspondence to: Raghav Singhal <10raghavsinghal@gmail.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

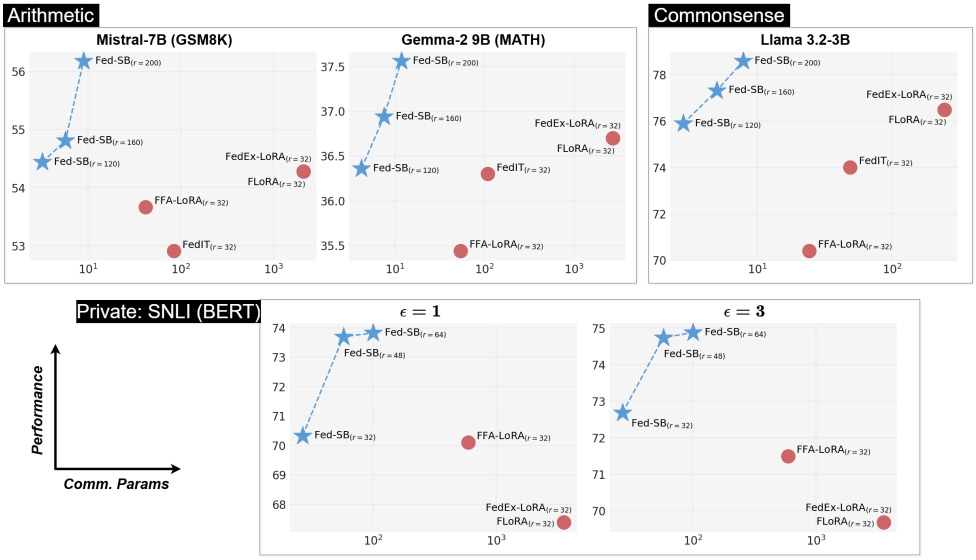


Figure 1: Performance vs. communicated parameter cost for Fed-SB and other federated FT methods in non-private and private federated settings. Fed-SB advances the performance-communication cost Pareto frontier across all models and tasks, achieving **state-of-the-art** accuracy while significantly reducing communication cost.

Table 1: Advantages of Fed-SB over SOTA federated FT methods (c clients). Denote $\kappa_1 := \mathcal{O}((m+n)r)$, $\kappa_2 := \mathcal{O}(\min(c(m+n)r, mn))$, $\kappa_3 := \mathcal{O}(mr)$ and $\kappa_4 := \mathcal{O}(r^2)$.

Property	FedIT	FLoRA	FedEx-LoRA	FFA-LoRA	Fed-SB
Exact aggregation	✗	✓	✓	✓	✓
Learnable params.	κ_1	κ_1	κ_1	κ_3	κ_4
Comm. cost	κ_1	κ_2	κ_2	κ_3	κ_4
No noise ampl.	✗	✗	✗	✓	✓
Privacy (fewer params.)	✗	✗	✗	✗	✓
Optimal expressivity	✓	✓	✓	✗	✓

federated fine-tuning.

2. Method

Federated Fine-Tuning. Given a pretrained weight matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$, the objective in FT is to learn an update $\Delta\mathbf{W}$ for a given dataset. LoRA (Hu et al., 2021) remains the preferred method, where low-rank adapter matrices $\mathbf{A} \in \mathbb{R}^{r\times n}$ and $\mathbf{B} \in \mathbb{R}^{m\times r}$ are learned such that $\Delta\mathbf{W} = \mathbf{BA}$. In federated learning, the dataset is distributed across c clients, and the goal is to learn $\Delta\mathbf{W}$ without sharing local data with a central server. To achieve this, each client learns its own adapter matrices $\mathbf{A}_i$ and $\mathbf{B}_i$. The server aggregates these updates to refine $\mathbf{W}$, along with globally beneficial representations of $\mathbf{A}$ and $\mathbf{B}$, ultimately producing a shared aggregate model $\mathbf{W}^{\text{agg}}$. Next, each client continues the local FT process, followed by aggregation at the end of each round. This cycle repeats over multiple rounds.

Fed-SB: A Silver bullet for (Private) Federated Fine-Tuning. We propose **Fed-SB**, an extremely communication-efficient and high-performing federated adaptation of LoRA-SB. Instead of reparameterizing updates as a low-rank decomposition with learnable adapters, the server distributes frozen adapters $\mathbf{B}$ and $\mathbf{A}$, while clients train only a small

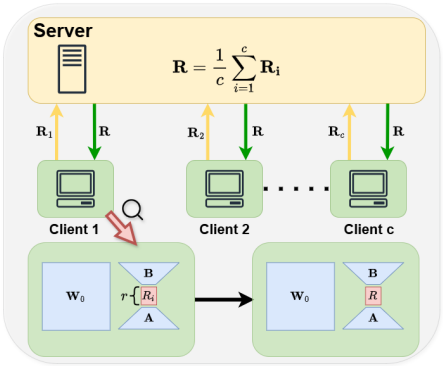


Figure 2: **Fed-SB**: Our method achieves optimal exact aggregation by averaging only the $r \times r$ matrices $\mathbf{R}_i$.

matrix $\mathbf{R}$ (Figure 2). This enables exact aggregation, as the global update is simply the average of $\mathbf{R}$ across clients. Formally, given a pre-trained weight $\mathbf{W}_0$ and data distributed across c clients, each client learns updates of the form:

$$\Delta\mathbf{W}_i = \mathbf{BR}_i\mathbf{A}. \tag{1}$$

The server then aggregates the updates by computing the global $\mathbf{R}$ matrix:

$$\mathbf{R}^{\text{agg}} = \frac{1}{c}\sum_{i=1}^{c}\mathbf{R}_i, \Delta\mathbf{W}^{\text{agg}} = \mathbf{B}\left(\frac{1}{c}\sum_{i=1}^{c}\mathbf{R}_i\right)\mathbf{A}. \tag{2}$$

We show that **Fed-SB** effectively resolves all challenges in (private) federated FT while achieving state-of-the-art communication efficiency and performance. Table 1 highlights the advantages of Fed-SB over other methods.

Fed-SB: Exact Aggregation. Since only $\mathbf{R}$ is trainable, simple averaging of $\mathbf{R}$ across clients ensures exact aggregation without any updates to any other matrix. Further, the linearity of the global update with respect to the client-specific

matrices $\mathbf{R}_i$ guarantees that exact aggregation occurs within rank r, preventing communication costs from scaling with number of clients. This is because the server only needs to aggregate and transmit $\mathbf{R}$, which can be proven by computing the global update $\Delta\mathbf{W}^{\text{agg}}$:

$$\Delta\mathbf{W}^{\text{agg}} = \mathbf{B}\left(\frac{1}{c}\sum_{i=1}^{c}\mathbf{R}_i\right)\mathbf{A}, \tag{3}$$

$$\Delta\mathbf{W}^{\text{agg}} = \frac{1}{c}\sum_{i=1}^{c}\mathbf{B}\mathbf{R}_i\mathbf{A} = \frac{1}{c}\sum_{i=1}^{c}\Delta\mathbf{W}_i. \tag{4}$$

Since the global update is simply the average of the individual updates, the aggregation is exact. The key advantage here is that this exact aggregation does not incur additional communication overhead like FedEx-LoRA, nor does it compromise individual update quality like FFA-LoRA.

Fed-SB: Privacy. Privacy-preserving FT with Fed-SB has two key advantages: 1) Fed-SB avoids noise amplification, a common issue in LoRA-based methods. 2) Since Fed-SB requires fewer learnable parameters, the amount of noise added to enforce DP guarantees is significantly lower.

Avoids Noise Amplification. DP-SGD training in Fed-SB avoids second-order noise terms, as only $\mathbf{R}$ is trainable. This prevents the introduction of cross terms, thereby eliminating noise amplification. The difference between the updates with and without private training is given by:

$$\Delta\mathbf{W}_{DP} - \Delta\mathbf{W} = \mathbf{B}\left(\mathbf{R} + \boldsymbol{\xi}_B\right)\mathbf{A} - \mathbf{BRA} \tag{5}$$

$$\implies \Delta\mathbf{W}_{DP} - \Delta\mathbf{W} = \mathbf{B}\boldsymbol{\xi}_B\mathbf{A}. \tag{6}$$

Since the private update remains linear in $\mathbf{R}$, Fed-SB achieves significant benefits in private settings.

Fewer Learnable Parameters. The noise added for DP enforcement increases with the number of trainable parameters (Bassily et al., 2014; Abadi et al., 2016; Bun et al., 2014), potentially distorting learning and degrading performance. Reducing trainable parameters improves DP performance, provided the model retains sufficient expressivity.

Lemma 2.1. *Consider a model with d learnable parameters trained using DP-SGD. The privacy parameter ϵ for δ-approximate differential privacy, given T training steps and a batch size of q, is:*

$$\epsilon = O(q\sqrt{Td\log(1/\delta)}) = O(\sqrt{d}). \tag{7}$$

Proof. See Appendix B. □

Lemma 2.1 establishes that reducing the number of learnable parameters enhances privacy guarantees under the same training setup. Specifically, achieving an equivalent level of privacy requires injecting less noise per parameter when fewer parameters are trained. Since LoRA-SB optimally approximates full fine-tuning gradients, its updates remain as effective as those in LoRA while benefiting from lower noise per update, resulting in a superior privacy-utility trade-off. More generally, any reparameterization that reduces trainable parameters leads to a smaller accumulated privacy parameter ϵ, thereby improving performance, provided the reduction does not compromise learning.

Table 2: Federated fine-tuning of Llama-3.2 3B across eight commonsense reasoning datasets. # C. denotes the number of parameters communicated per round (in M).

Method	# C.	Accuracy (↑)								
		BoolQ	PIQA	SIQA	Hella.	Wino.	ARC-e	ARC-c	OBQA	Avg.
FedIT$_{r=32}$	48.6	62.99	81.50	73.13	76.83	71.51	84.89	70.65	70.62	74.02
FFA$_{r=32}$	24.3	62.87	80.03	68.53	70.02	65.56	82.95	66.38	66.85	70.40
FedEx$_{r=32}$	243.1	65.05	82.81	74.67	81.84	76.01	86.32	71.42	73.81	76.49
FLoRA$_{r=32}$	243.1	65.05	82.81	74.67	81.84	76.01	86.32	71.42	73.81	76.49
Fed-SB$_{r=120}$	2.8	64.86	81.66	74.87	81.67	75.22	86.03	70.56	72.25	75.89
Fed-SB$_{r=160}$	5.0	65.57	82.37	76.15	84.10	77.98	86.62	72.10	73.63	77.32
Fed-SB$_{r=200}$	7.8	**66.66**	**83.79**	**77.22**	**85.42**	**79.56**	**87.46**	**72.53**	**76.02**	**78.58**

Table 3: Federated fine-tuning of LLaMA-3.2 3B on commonsense reasoning in a **highly data-heterogeneous** setting, where each client trains on a distinct dataset. Communication cost (# C.) is in millions of parameters per round.

Method	# C.	Accuracy (↑)								
		BoolQ	PIQA	SIQA	Hella.	Wino.	ARC-e	ARC-c	OBQA	Avg.
FedIT$_{r=32}$	48.6	60.89	78.22	69.92	73.18	67.88	81.21	67.04	66.91	70.80
FFA$_{r=32}$	24.3	60.73	76.91	65.37	65.18	61.89	79.41	62.92	63.12	67.17
FedEx$_{r=32}$	243.1	62.55	79.36	71.41	78.12	72.45	82.89	67.88	70.25	73.13
FLoRA$_{r=32}$	243.1	62.55	79.36	71.41	78.12	72.45	82.89	67.88	70.25	73.13
Fed-SB$_{r=120}$	2.8	61.41	78.13	71.02	78.24	71.78	82.45	67.12	68.83	72.65
Fed-SB$_{r=160}$	5.0	62.34	79.05	72.39	80.52	74.67	83.18	68.64	70.12	73.98
Fed-SB$_{r=200}$	7.8	**63.28**	**80.34**	**73.56**	**82.07**	**76.01**	**84.01**	**69.02**	**72.46**	**75.21**

Fed-SB: Pushing the Pareto Frontier. Fed-SB has significantly less communication costs than other federated FT methods. This is due to two key reasons: 1) LoRA-SB achieves performance comparable to or better than LoRA while requiring 45-90x fewer trainable parameters. 2) Fed-SB aggregates only the $r \times r$ trainable matrix $\mathbf{R}$, ensuring exact aggregation without additional communication overhead. This allows Fed-SB to leverage higher-rank updates without increasing communication costs. LoRA-SB typically operates at ranks 2–4x higher than LoRA, enabling Fed-SB to capture richer updates. Retaining high-rank information is crucial in FL (Mahla et al., 2025) and a key factor in achieving improved performance.

3. Experiments & Results

Overview. We fine-tune Mistral-7B (Jiang et al., 2023), Gemma-2 9B (Team et al., 2024), Llama-3.2 3B (Dubey et al., 2024), and BERT-base (Devlin, 2018) across three diverse benchmarks. Detailed experimental and dataset specifications are provided in Appendix E and F, respectively.

Baselines. We evaluate Fed-SB against several SOTA federated FT approaches, considering private and non-private

Table 4: Federated fine-tuning of Mistral-7B and Gemma-2 9B on GSM8K and MATH. Communication cost (# Comm.) is in millions of parameters per round.

Model	Method	Rank	# Comm.	Accuracy (↑)	
				GSM8K	MATH
Mistral-7B	FedIT	32	83.88	52.91	12.26
	FFA-LoRA	32	41.94	53.67	12.46
	FedEx-LoRA	32	2097.34	54.28	12.92
	FLoRA	32	2097.34	54.28	12.92
	Fed-SB	120	3.22	54.44	**14.06**
	Fed-SB	160	5.73	54.81	13.74
	Fed-SB	200	8.96	**56.18**	13.76
Gemma-2 9B	FedIT	32	108.04	74.22	36.30
	FFA-LoRA	32	54.02	75.06	35.44
	FedEx-LoRA	32	2701.12	74.68	36.70
	FLoRA	32	2701.12	74.68	36.70
	Fed-SB	120	4.23	74.75	36.36
	Fed-SB	160	7.53	76.88	36.94
	Fed-SB	200	11.76	**77.03**	**37.56**

settings. Specifically, we compare it with **FedIT** (Zhang et al., 2024b), **FedEx-LoRA** (Singhal et al., 2025), **FLoRA** (Wang et al., 2024), and **FFA-LoRA** (Wang et al., 2024).

3.1. Instruction Tuning

Details. We conduct experiments in the **federated non-private** setting across two reasoning tasks: commonsense reasoning and arithmetic reasoning. For **commonsense reasoning**, we fine-tune Llama-3.2 3B on CommonSense170K, a dataset aggregating eight commonsense reasoning corpora (Hu et al., 2023), and evaluate across all constituent datasets. The experiments are performed in a cross-silo federated learning setup involving 5 clients.

We also evaluate Fed-SB **under extreme data heterogeneity**. Instead of randomly sampling examples for each client, we assign each constituent dataset to a distinct client, resulting in a **highly non-IID** 8-client setup. Each client trains on a distinct distribution, with varying dataset sizes.

For **arithmetic reasoning**, we fine-tune Mistral-7B and Gemma-2 9B on 20K samples from MetaMathQA (Yu et al., 2024) and test on the GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) benchmarks. In this setup, we distribute the federated training across 25 clients. In both cases, we apply LoRA modules to the key, query, value, attention output, and all fully connected weights.

Results (Tables 2, 3, 4). Our method achieves **state-of-the-art performance**, outperforming all previous baselines in both accuracy and communication efficiency **across all models and benchmarks**. Figure 3 further illustrates this significant improvement.

Commonsense Reasoning (Table 2). Fed-SB ($r = 200$) achieves an average improvement of 4.56% over FedIT while requiring **6×** lower communication cost. Additionally, Fed-SB ($r = 200$) surpasses the previous SOTA performance methods FedEx-LoRA/FLoRA by 2.09%, while reducing communication cost by an impressive **31×**. Notably, while the communication cost of FedEx-LoRA/FLoRA scales linearly with the number of clients, our method maintains a constant, client-independent communication cost.

Table 5: Centralized (Cent.) private fine-tuning of BERT-base on SNLI under varying privacy budgets ϵ. Trainable parameters (# Params.) are in thousands.

Method	Rank	# Params.	Accuracy (↑)				
			ϵ=1	ϵ=3	ϵ=5	ϵ=7.5	ϵ=10
Cent. LoRA	32	1181.96	66.49	67.79	68.17	70.78	70.81
Cent. FFA-LoRA	32	592.13	74.40	75.02	75.02	76.14	76.60
Cent. Fed-SB	32	26.88	73.99	75.09	74.45	77.01	76.24
Cent. Fed-SB	48	57.59	**75.98**	75.70	76.58	76.77	77.96
Cent. Fed-SB	64	100.61	75.81	**77.07**	**77.59**	**78.75**	**78.08**

Table 6: Federated private fine-tuning of BERT-base on SNLI under varying privacy budgets ϵ. Communication cost (# Comm.) is in thousands of parameters per round.

Method	Rank	# Comm.	Accuracy (↑)				
			ϵ=1	ϵ=3	ϵ=5	ϵ=7.5	ϵ=10
FedIT	32	1181.96	49.57	51.29	48.53	55.63	60.96
FFA-LoRA	32	592.13	70.11	71.49	72.69	73.27	74.02
FedEx-LoRA	32	3541.26	67.38	69.68	72.92	71.89	74.33
FLoRA	32	3541.26	67.38	69.68	72.92	71.89	74.33
Fed-SB	32	26.88	70.33	72.68	73.57	73.62	73.85
Fed-SB	48	57.59	73.70	74.74	73.66	74.75	75.02
Fed-SB	64	100.61	**73.83**	**74.88**	**76.27**	**75.75**	**75.86**

Highly Data-Heterogenous Setting (Table 3). Fed-SB significantly outperforms all other methods even in this highly non-IID setting. Specifically, Fed-SB ($r = 200$) surpasses the previous state-of-the-art methods, FedEx-LoRA and FLoRA, by 2.08% in accuracy while achieving a remarkable **31×** reduction in communication cost.

Arithmetic Reasoning (Table 4). For Mistral-7B, Fed-SB ($r = 200$) outperforms FedEx-LoRA/FLoRA on GSM8K by 1.90%, while achieving a **234×** reduction in communication cost. Additionally, Fed-SB ($r = 200$) surpasses FFA-LoRA on GSM8K by 2.51%, with approximately **5×** lower communication cost. For Gemma-2 9B, Fed-SB ($r = 200$) outperforms FedEx-LoRA/FLoRA on MATH by 0.86%, while reducing communication cost by **230×**.

3.2. (Federated) Private Fine-Tuning

Details. We fine-tune BERT-base on SNLI (Bowman et al., 2015), a standard language inference benchmark. Following LoRA(Hu et al., 2021), we apply LoRA modules only to the self-attention layers. Our evaluation considers two DP settings: a **centralized private** setup and a **federated private** setup. To enforce DP guarantees, we use the Opacus library (Yousefpour et al., 2021) with the DP-SGD optimizer (Abadi et al., 2016). In the federated setting, training is conducted in a cross-silo setup with 3 clients.

Results (Tables 5, 6). Fed-SB consistently outperforms all prior baselines in **both accuracy and communication/parameter efficiency** across **all privacy budgets** in both settings. Figures 4, 5, and 6 further illustrate this.

Centralized Private (Table 5). Fed-SB showcases significant improvement over other methods while using only a fraction of the parameters, across all ϵ values. For instance, at $\epsilon = 3$, Fed-SB ($r = 64$) surpasses centralized LoRA and centralized FFA-LoRA by 9.28% and 2.05%, respectively, while using $\approx$ **12x** and **6x** fewer parameters.

Federated Private (Table 6). Fed-SB consistently outperforms all methods across all values of ϵ, while significantly reducing communication costs. For instance, at $\epsilon = 1$, Fed-SB ($r = 64$) outperforms FedIT, FedEx-LoRA/FLoRA, and FFA-LoRA by 24.26%, 6.48%, and 2.72%, respectively, while reducing communication cost by approximately **12x**, **35x**, and **6x**. FedIT performs significantly worse in the federated private setting compared to the federated non-private setting. We hypothesize that this is due to increased deviation in updates under DP constraints and added noise, leading to greater divergence from the ideal.

4. Conclusion

Fed-SB is a communication-efficient federated adaptation of LoRA-SB that ensures exact aggregation by training only a small $r \times r$ matrix and using direct averaging. This eliminates high-rank update costs and keeps communication overhead independent of the number of clients. Its linearity avoids noise amplification, and its compact parameterization reduces the noise needed for differential privacy. Fed-SB achieves up to **230**$\times$ lower communication costs and sets a **new state-of-the-art** across all models and tasks, making it ideal for scalable (private) federated fine-tuning.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements and Disclosure of Funding

This research was supported by Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), with partial funding from the ADIA Lab Fellowship.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Babakniya, S., Elkordy, A. R., Ezzeldin, Y. H., Liu, Q., Song, K.-B., El-Khamy, M., and Avestimehr, S. Slora: Federated parameter efficient fine-tuning of language models. *arXiv preprint arXiv:2308.06522*, 2023.

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pp. 464–473. IEEE, 2014.

Bałazy, K., Banaei, M., Aberer, K., and Tabor, J. Lora-xs: Low-rank adaptation with extremely small number of parameters, 2024. URL https://arxiv.org/abs/2405.17604.

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, H. B., Overveldt, T. V., Petrou, D., Ramage, D., and Roselander, J. Towards federated learning at scale: System design, 2019. URL https://arxiv.org/abs/1902.01046.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

Bun, M., Ullman, J., and Vadhan, S. Fingerprinting codes and the price of approximate differential privacy. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 1–10, 2014.

Cho, Y. J., Liu, L., Xu, Z., Fahrezi, A., and Joshi, G. Heterogeneous lora for federated fine-tuning of on-device foundation models, 2024. URL https://arxiv.org/abs/2401.06432.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dwork, C. Differential privacy. In *International colloquium on automata, languages, and programming*, pp. 1–12. Springer, 2006.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Hu, Z., Wang, L., Lan, Y., Xu, W., Lim, E.-P., Bing, L., Xu, X., Poria, S., and Lee, R. K.-W. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models, 2023. URL https://arxiv.org/abs/2304.01933.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

Kalajdzievski, D. A rank stabilization scaling factor for fine-tuning with lora, 2023. URL https://arxiv.org/abs/2312.03732.

Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency, 2017. URL https://arxiv.org/abs/1610.05492.

Kopiczko, D. J., Blankevoort, T., and Asano, Y. M. Vera: Vector-based random matrix adaptation, 2024. URL https://arxiv.org/abs/2310.11454.

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. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5260–5271, 2024.

Lai, F., Dai, Y., Singapuram, S., Liu, J., Zhu, X., Madhyastha, H., and Chowdhury, M. Fedscale: Benchmarking model and system performance of federated learning at scale. In *International conference on machine learning*, pp. 11814–11827. PMLR, 2022.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Li, Z., Ding, B., Zhang, C., Li, N., and Zhou, J. Federated matrix factorization with privacy guarantee. *Proc. VLDB Endow.*, 15(4):900–913, December 2021. ISSN 2150-8097. doi: 10.14778/3503585.3503598. URL https://doi.org/10.14778/3503585.3503598.

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. URL https://arxiv.org/abs/2402.09353.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Mahla, N., Jadhav, K. S., and Ramakrishnan, G. Exploring gradient subspaces: Addressing and overcoming lora's limitations in federated fine-tuning of large language models, 2025. URL https://arxiv.org/abs/2410.23111.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Meng, F., Wang, Z., and Zhang, M. Pissa: Principal singular values and singular vectors adaptation of large language models, 2024. URL https://arxiv.org/abs/2404.02948.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Mironov, I. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pp. 263–275. IEEE, 2017.

Ponkshe, K., Singhal, R., Gorbunov, E., Tumanov, A., Horvath, S., and Vepakomma, P. Initialization using update approximation is a silver bullet for extremely efficient low-rank fine-tuning. *arXiv preprint arXiv:2411.19557*, 2024.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Sap, M., Rashkin, H., Chen, D., LeBras, R., and Choi, Y. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

Singhal, R., Ponkshe, K., and Vepakomma, P. Fedex-lora: Exact aggregation for federated and efficient fine-tuning of foundation models. *arXiv preprint arXiv:2410.09432*, 2025.

Sun, Y., Li, Z., Li, Y., and Ding, B. Improving lora in privacy-preserving federated learning. *arXiv preprint arXiv:2403.12313*, 2024.

Tang, X., Panda, A., Nasr, M., Mahloujifar, S., and Mittal, P. Private fine-tuning of large language models with zeroth-order optimization. *arXiv preprint arXiv:2401.04343*, 2024.

Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Team, G., Riviere, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Tian, Y., Wan, Y., Lyu, L., Yao, D., Jin, H., and Sun, L. Fedbert: When federated learning meets pre-training. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–26, 2022.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Truong, N., Sun, K., Wang, S., Guitton, F., and Guo, Y. Privacy preservation in federated learning: An insightful survey from the gdpr perspective. *Computers & Security*, 110:102402, 2021.

Wang, Y.-X., Balle, B., and Kasiviswanathan, S. P. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1226–1235. PMLR, 2019.

Wang, Z., Shen, Z., He, Y., Sun, G., Wang, H., Lyu, L., and Li, A. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *arXiv preprint arXiv:2409.05976*, 2024.

Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021.

Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H. A., Kamath, G., Kulkarni, J., Lee, Y. T., Manoel, A., Wutschitz, L., et al. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*, 2021.

Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., Kwok, J. T., Li, Z., Weller, A., and Liu, W. Metamath: Bootstrap your own mathematical questions for large language models, 2024. URL https://arxiv.org/abs/2309.12284.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Zhang, B., Liu, Z., Cherry, C., and Firat, O. When scaling meets llm finetuning: The effect of data, model and finetuning method, 2024a. URL https://arxiv.org/abs/2402.17193.

Zhang, J., Vahidian, S., Kuo, M., Li, C., Zhang, R., Yu, T., Wang, G., and Chen, Y. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6915–6919. IEEE, 2024b.

Zhang, J., Vahidian, S., Kuo, M., Li, C., Zhang, R., Yu, T., Zhou, Y., Wang, G., and Chen, Y. Towards building the federated gpt: Federated instruction tuning, 2024c. URL https://arxiv.org/abs/2305.05644.

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, 2023. URL https://arxiv.org/abs/2303.10512.

Zhang, Z., Yang, Y., Dai, Y., Qu, L., and Xu, Z. When federated learning meets pre-trained language models' parameter-efficient tuning methods. *arXiv preprint arXiv:2212.10025*, 2022.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Appendix

A. Preliminaries and Motivation

We summarize some of the state-of-the-art federated FT methods below.

Fed-IT (Zhang et al., 2024b) updates the adapters $\mathbf{A}$ and $\mathbf{B}$ using the standard FedAvg (McMahan et al., 2017) algorithm:

$$\mathbf{A}^{\text{agg}} = \frac{1}{c}\sum_{i=1}^{c}\mathbf{A}_i, \quad \mathbf{B}^{\text{agg}} = \frac{1}{c}\sum_{i=1}^{c}\mathbf{B}_i. \tag{8}$$

FedEx-LoRA (Singhal et al., 2025) follows the same aggregation but introduces an additional error correction matrix $\mathbf{W}_{\text{err}}$ of rank $\min(cr, m, n)$:

$$\mathbf{W}_{\text{err}} = (\frac{1}{c}\sum_{i=1}^{c}\mathbf{A}_i\mathbf{B}_i) - (\frac{1}{c}\sum_{i=1}^{c}\mathbf{A}_i)(\frac{1}{c}\sum_{i=1}^{c}\mathbf{B}_i). \tag{9}$$

FLoRA (Wang et al., 2024) follows the same principle as FedEx-LoRA but achieves it by stacking the adapter matrices, and reinitializes them randomly at the end of each communication round. **FFA-LoRA** (Sun et al., 2024) keeps $\mathbf{A}$ fixed while training (and aggregating) only $\mathbf{B}$ matrices.

$$\mathbf{B}^{\text{agg}} = \frac{1}{c}\sum_{i=1}^{c}\mathbf{B}_i. \tag{10}$$

(Approximate) Differential Privacy. DP, introduced by (Dwork, 2006), is a widely adopted mathematical framework for privacy preservation. A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$, mapping a domain $\mathcal{D}$ to a range $\mathcal{R}$, satisfies (ϵ, δ)-differential privacy if, for any two adjacent inputs $d, d' \in \mathcal{D}$ and any subset of outputs $S \subseteq \mathcal{R}$, the following holds:

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\epsilon}\Pr[\mathcal{M}(d') \in S] + \delta. \tag{11}$$

DP-SGD. DP-SGD (Abadi et al., 2016) is a privacy-preserving variant of stochastic gradient descent (SGD) designed to ensure DP during training. It enforces privacy by clipping per-sample gradients to a fixed norm C to limit their sensitivity and then adding isotropic Gaussian noise $\mathcal{N}\left(0, \sigma^2 C^2 \mathbf{I}\right)$, where σ controls the noise magnitude. The cumulative privacy loss over iterations is quantified using the moments accountant (Wang et al., 2019) and Rényi DP (Mironov, 2017), which offer a tight bound on the final privacy parameter ϵ.

Exact Aggregation in Fed. LoRA: Tradeoff b/w Performance and Communication Costs.

Standard federated averaging of individual LoRA adapters (FedIT (Zhang et al., 2024b)) introduces *inexactness* in aggregation, as the ideal update should be the average of client updates.

$$\underbrace{\mathbf{W}_0 + \frac{1}{c}\sum_{i=1}^{c}\mathbf{B}_i \times \frac{1}{c}\sum_{i=1}^{c}\mathbf{A}_i}_{\text{Vanilla aggregation in LoRA (FedIT)}} \neq \underbrace{\mathbf{W}_0 + \frac{1}{c}\sum_{i=1}^{c}(\mathbf{B}_i\mathbf{A}_i)}_{\text{Ideal aggregation}}. \tag{12}$$

The inexactness arises because the ideal averaged updates, given by $\sum_{i=1}^{c}\mathbf{B}_i\mathbf{A}_i$, often exceed rank r, violating the low-rank constraint imposed by LoRA. To address this, FedEx-LoRA and FLoRA introduce $\mathbf{W}_{\text{err}}$ as a higher-rank correction term within the pre-trained weight matrix $\mathbf{W}_0$, which is inherently high-rank. This correction ensures exact aggregation, leading to consistently improved performance over FedIT.

This, however, comes at the cost of increased communication. Since the error matrix is high rank, it substantially increases the amount of data transmitted per round. The communication cost is determined by the number of parameters sent during aggregation, which, for an $m \times n$ matrix, is proportional to its rank. As a result, in FedEx-LoRA and similar methods that enforce exact aggregation, communication cost scales linearly with the number of clients relative to Fed-IT. This becomes particularly concerning when the number of clients grows large, **potentially requiring the transmission of the entire model's weights**.

FFA-LoRA addresses inexact aggregation by keeping only $\mathbf{B}$ trainable while fixing $\mathbf{A}$ uniformly across clients. However, this comes at the cost of reduced expressivity and limits the benefits of jointly optimizing $\mathbf{A}$ and $\mathbf{B}$. As a result, performance degrades, as demonstrated previously (Singhal et al., 2025). This stems from two factors: suboptimal individual updates and the need for higher-rank adaptations. Freezing $\mathbf{A}$ leads to suboptimal updates, even in centralized training, where FFA-LoRA underperforms compared to LoRA. Additionally, recent work (Mahla et al., 2025) shows that models trained using FFA-LoRA progressively deviate from the optimal hypothesis. Empirical evidence shows that the advantages of exactness are outweighed by the degradation caused by these factors.

Private Fine-Tuning. Pre-training on public data followed by FT on user-specific private data[1] is a common approach for adapting models under privacy constraints (Yu et al., 2021; Tang et al., 2024). This two-stage process enhances performance in private learning while preserving user data privacy. FL naturally improves privacy by keeping data decentralized. However, even without direct data sharing, client-specific model updates can still leak sensitive information (Truong et al., 2021). Thus, developing privacy-preserving FT methods for FL is essential to ensure strong privacy guarantees while maintaining performance.

Training a model with DP-SGD introduces noise into the gradient, and consequently, into the model update itself. In the case of LoRA, this deviation from the ideal update is more pronounced than in full FT due to second-order noise terms. To illustrate this, let $\mathbf{A}$ and $\mathbf{B}$ represent the adapter updates learned without privacy. Under DP-SGD, these updates are perturbed by noise terms $\boldsymbol{\xi}_A$ and $\boldsymbol{\xi}_B$, respectively. The difference between the ideal update $\Delta\mathbf{W}$ and the noisy update $\Delta\mathbf{W}_{DP}$ is:

$$\Delta\mathbf{W}_{DP} - \Delta\mathbf{W} = (\mathbf{B} + \boldsymbol{\xi}_B)(\mathbf{A} + \boldsymbol{\xi}_A) - \mathbf{BA} \tag{13}$$

$$= \boldsymbol{\xi}_B\mathbf{A} + \mathbf{B}\boldsymbol{\xi}_A + \boldsymbol{\xi}_B\boldsymbol{\xi}_A. \tag{14}$$

The first-order noise term, $\boldsymbol{\xi}_B\mathbf{A} + \mathbf{B}\boldsymbol{\xi}_A$, is expected and occurs even in full FT with DP-SGD. However, the second-order noise term, $\boldsymbol{\xi}_B\boldsymbol{\xi}_A$, causes **noise amplification**, leading to further performance degradation in LoRA-based methods (Sun et al., 2024). This issue is exacerbated in FL, as individual client updates deviate even further from the ideal global update. FFA-LoRA avoids this problem by freezing $\mathbf{A}$, preventing the introduction of additional noise terms.

A Silver Bullet Indeed. The bilinear parameterization in LoRA introduces two key challenges: inexact aggregation and noise amplification. FedEx-LoRA/FLoRA addresses the inexactness issue by enabling exact aggregation, but at the cost of communication overhead that scales prohibitively with the number of clients. FFA-LoRA mitigates inexact aggregation and excessive communication but sacrifices performance, as it operates in a low-rank space and has reduced expressivity. An ideal method would efficiently learn higher-rank updates while inherently enabling exact aggregation without increasing communication costs. However, any LoRA-based formulation that attempts to resolve these challenges must inevitably trade off expressivity, ultimately compromising performance. We prove that LoRA-SB provides an optimal reparameterization of the updates, effectively overcoming all limitations of LoRA in both non-private and privacy-preserving federated settings.

B. Proof of Lemma 2.1

Lemma. Consider a model with d learnable parameters trained using DP-SGD. The privacy parameter ϵ for δ-approximate differential privacy, given T training steps and a batch size of q, is expressed as:

$$\epsilon = O(q\sqrt{Td\log(1/\delta)}) = O(\sqrt{d}). \tag{15}$$

Proof. The following result (Abadi et al., 2016) describes the relationship between noise variance, privacy parameters, number of optimization steps, batch size, and sample size in DP-SGD.

Theorem. There exist constants c_1 and c_2 such that, given the sampling probability $q = L/N$ and the number of optimization steps T, for any $\epsilon < c_1 q^2 T$, DP-SGD is (ϵ, δ)-differentially private for any $\delta > 0$ if the noise scale satisfies:

$$\sigma \geq c_2 \frac{q\sqrt{T\log(1/\delta)}}{\epsilon}. \tag{16}$$

[1]Although pre-training data may be public, it often contains sensitive or proprietary information, raising privacy concerns. However, any privacy loss from pre-training has already occurred upon the model's release.

Each DP-SGD step introduces noise following $\mathcal{N}\left(0, \sigma^2 C^2 \mathbf{I}_d\right)$ and satisfies $(\alpha, \alpha/(2\sigma^2))$-RDP (Rényi DP) for the Gaussian mechanism. For a function with ℓ_2-sensitivity Δ_2, the Gaussian mechanism satisfies (α, ϵ)-RDP with:

$$\epsilon(\alpha) = \frac{\alpha \Delta_2^2}{2\sigma_{\text{noise}}^2}. \tag{17}$$

Since DP-SGD has $\Delta_2 = C$ and $\sigma_{\text{noise}} = \sigma C$, applying privacy amplification due to sampling probability q results in each step satisfying (α, γ)-RDP, where, for small q:

$$\gamma = O\left(\frac{q^2 \alpha}{\sigma^2}\right). \tag{18}$$

Using composition over T steps, the total RDP privacy parameter becomes:

$$\gamma_{\text{total}} = O\left(\frac{q^2 T \alpha}{\sigma^2}\right). \tag{19}$$

Converting this RDP bound back to (ϵ, δ)-DP and setting α proportional to $1/\sqrt{d}$, given that the ℓ_2-norm of the gradient scales as $\sqrt{d}$, we obtain:

$$\epsilon = O\left(\frac{q^2 T \alpha}{\sigma^2} + \frac{\log(1/\delta)}{\alpha - 1}\right). \tag{20}$$

Substituting $\sigma \propto 1/\sqrt{d}$, we derive:

$$\epsilon = O(q\sqrt{Td\log(1/\delta)}) = O(\sqrt{d}). \tag{21}$$

□

C. Related Work

Parameter-Efficient Fine-Tuning (PEFT). LoRA (Hu et al., 2021) has become ubiquitous for fine-tuning LLMs (Zhang et al., 2024a) by modeling weight updates as product of low-rank matrices. Several variants have been proposed to improve efficiency, stability, and adaptability. QLoRA (Dettmers et al., 2024) enables efficient fine-tuning through quantization strategies, reducing memory usage while maintaining performance. AdaLoRA (Zhang et al., 2023) dynamically allocates a layer-specific rank budget by assigning importance scores to individual weight matrices. LoRA-XS (Bałazy et al., 2024) further reduces trainable parameters by inserting a trainable matrix between frozen LoRA matrices. VeRA (Kopiczko et al., 2024) enhances parameter efficiency by learning shared adapters across layers. DoRA (Liu et al., 2024) decomposes the pre-trained matrix into two parts—*magnitude* and *direction*—and applies LoRA modules only to the *direction* component. PiSSA (Meng et al., 2024) improves adaptation by initializing adapters using the singular value decomposition (SVD) of pre-trained weights. rsLoRA (Kalajdzievski, 2023) introduces a rank-scaling factor to stabilize learning. LoRA-SB (Ponkshe et al., 2024) provably approximates gradients optimally in low-rank spaces, achieving superior performance with significantly higher parameter efficiency.

Federated Fine-Tuning. Federated Learning (FL) consists of a centralized global model and multiple clients, each with its own local dataset and computational capacity. The global model is updated by aggregating client updates (Kairouz et al., 2021). FedBERT (Tian et al., 2022) focuses on federated pre-training, while other methods work on federated fine-tuning (Zhang et al., 2022; Kuang et al., 2024; Babakniya et al., 2023). Fed-IT (Zhang et al., 2024c) aggregates low-rank adapters across clients using standard federated averaging (McMahan et al., 2017) before updating the global model. To address inexact aggregation, FedEx-LoRA (Singhal et al., 2025) introduces an error matrix to correct residual errors, ensuring more precise updates. FLoRA (Wang et al., 2024) follows the same exact aggregation principle by stacking matrices and extends this approach to heterogeneous rank settings. FFA-LoRA (Sun et al., 2024) mitigates aggregation inexactness by freezing $\mathbf{A}$ and updating only the trainable low-rank adapter, averaging the latter to compute the global update. In some scenarios, clients require heterogeneous LoRA ranks due to varying computational budgets (Zhao et al., 2018; Li et al., 2019). Methods like HetLoRA (Cho et al., 2024) enable rank heterogeneity through self-pruning and sparsity-aware

aggregation strategies, but incur significant overhead.

Differential Privacy (DP) and FL. A common limitation of standard FL frameworks is their susceptibility to privacy attacks, as clients publicly share model updates with a central server. To address this issue, DP is incorporated into FL methods to ensure the privacy of client updates. This work follows the approximate DP framework (Dwork, 2006; Dwork et al., 2014), which provides formal privacy guarantees for model updates. Privacy is enforced during training using the DP-SGD optimizer (Abadi et al., 2016), which applies gradient clipping and noise injection to protect individual contributions. Since DP is preserved under composition and post-processing (Dwork, 2006; Li et al., 2021), the final global model update also retains DP guarantees. Prior methods, such as Fed-IT and FedEx-LoRA, did not explicitly incorporate DP. This study extends these approaches to DP settings and benchmarks them alongside FFA-LoRA and the proposed method.

D. Memory and Training Time Details

D.1. Memory and Training Time

Memory. Fed-SB, being derived from LoRA-SB, requires less training memory (per client) during training due to its significantly reduced number of trainable parameters, resulting in memory savings compared to other methods. We benchmark the peak per-client training memory for all models and configurations used in our study in Table 7. Notably, these results reflect the worst-case setting for Fed-SB, with the highest rank ($r = 200$) used in our experiments.

Table 7: Peak per-client training memory (in GB) for different methods across the various models used in this work. Fed-SB consistently exhibits lower memory usage across all model configurations.

Method	Peak Memory (GB)		
	Mistral-7B	Gemma-2 9B	LLaMA-3.2 3B
FedIT	15.92	19.99	7.71
FFA-LoRA	15.51	19.44	7.46
FedEx-LoRA	15.92	19.99	7.71
FLoRA	15.92	19.99	7.71
Fed-SB	15.18	19.03	7.30

Training Time. Fed-SB introduces a negligible training time overhead compared to other methods, primarily due to its lightweight initialization process. To quantify this, we measure the additional training time introduced by Fed-SB relative to the average per-epoch training time per client in baseline methods. These measurements are conducted across the various experimental settings described in our paper. As shown in Table 8, the overhead remains consistently minimal, approximately 2%, across multiple model configurations.

Table 8: Training time overhead introduced by Fed-SB relative to the average per-epoch training time per client in baseline methods. The overhead is minimal ($\approx 2\%$) across different model configurations.

Model	Fed-SB Overhead (mm:ss)	Avg. Epoch Time / Client (mm:ss)
Mistral-7B	00:13	09:22
Gemma-2 9B	00:16	12:43
LLaMA-3.2 3B	01:43	62:54

E. Experiment Details

Our experiments consider both performance and communication efficiency. For federated data distribution, we adopt a standard protocol where client datasets are randomly sampled, following established practice in FL (Sun et al., 2024; He et al., 2020; Lai et al., 2022). We conduct experiments on a single NVIDIA A6000 GPU (48 GB) and report the average results from three independent runs. All non-private models are trained using the AdamW optimizer (Loshchilov & Hutter, 2019).

To optimize memory efficiency, all base models (except BERT) are loaded in `torch.bfloat16`. In line with LoRA-SB

(Ponkshe et al., 2024), we initialize the adapter matrices using just $1/1000$ of the respective training dataset size.

Instruction Tuning. Table 9 presents the key hyperparameters and configurations for Mistral-7B, Gemma-2 9B, and Llama-3.2 3B. Our setup closely follows previous works (Hu et al., 2023; Ponkshe et al., 2024), ensuring consistency with established best practices. For the baseline experiments, we further set $\alpha = 16$, consistent with prior literature (Singhal et al., 2025; Sun et al., 2024). We additionally perform a sweep over the learning rate for our experiments.

(Federated) Private Fine-Tuning. Table 10 outlines the key hyperparameters and configurations for BERT-base in both centralized private and federated private settings. We train our models using the Opacus library (Yousefpour et al., 2021) with the DP-SGD optimizer (Abadi et al., 2016). Following standard DP practices, we set the privacy parameter as $\delta = \frac{1}{|\text{trainset}|}$. To ensure adherence to best practices, we adopt hyperparameter choices from prior works (Singhal et al., 2025; Hu et al., 2021). For baseline experiments, we additionally set $\alpha = 16$, aligning with previous literature (Singhal et al., 2025; Sun et al., 2024). We additionally perform a sweep over the learning rate and maximum gradient norm in DP-SGD for our experiments.

Table 9: Hyperparameter settings for Mistral-7B, Gemma-2 9B, and Llama-3.2 3B.

	Mistral-7B	**Gemma-2 9B**	**Llama-3.2 3B**
Optimizer	AdamW	AdamW	AdamW
Learning Rate	5e−4	5e−4	2e−4
LR Scheduler	Cosine	Cosine	Linear
Warmup Ratio	0.02	0.02	0.02
Batch Size	1	1	8
Grad Acc. Steps	32	32	24
Max. Seq. Len	512	512	256
Dropout	0	0	0
# Clients	25	25	5
Local Epochs	1	2	2
Rounds	1	1	1

Table 10: Hyperparameter settings for BERT-base in centralized private and federated private setups.

	BERT-base (centralized)	**BERT-base (federated)**
Optimizer	DP-SGD	DP-SGD
Learning Rate	5e−4	5e−4
LR Scheduler	-	-
Warmup Ratio	0	0
Batch Size	32	32
Max. Phy. Batch Size	8	8
Max. Seq. Len	128	128
Dropout	0.05	0.05
Max. Grad. Norm	0.1	0.1
Epochs	3	-
# Clients	-	3
Local Epochs	-	6
Rounds	-	1

F. Dataset Details

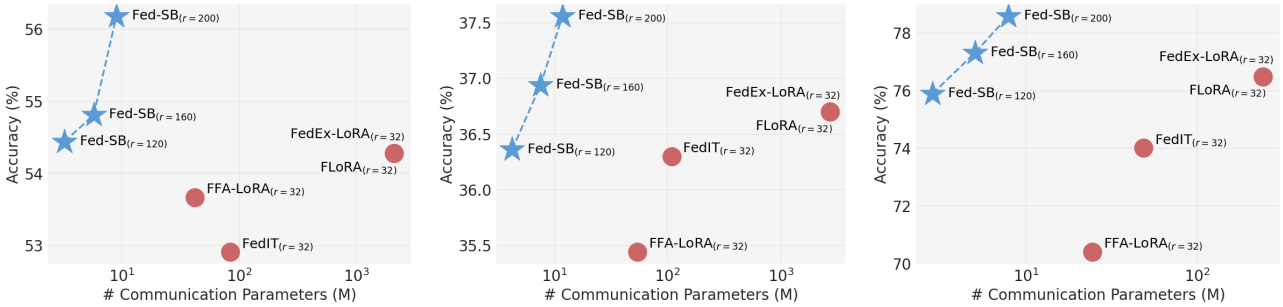
COMMONSENSE170K is a large-scale dataset that brings together eight benchmarks designed to assess various aspects of commonsense reasoning (Hu et al., 2023). Below is an overview of its constituent datasets:

1. **PIQA** (Bisk et al., 2020) evaluates physical commonsense by asking models to determine the most reasonable action in a given scenario.
2. **ARC Easy (ARC-e)** (Clark et al., 2018) consists of elementary-level science questions, serving as a fundamental test of a model's reasoning abilities.
3. **OBQA** (Mihaylov et al., 2018) presents knowledge-intensive, open-book multiple-choice questions that require multi-step reasoning and retrieval.
4. **HellaSwag** (Zellers et al., 2019) tests contextual reasoning by asking models to predict the most plausible continuation of a passage from a set of candidates.
5. **SIQA** (Sap et al., 2019) examines social intelligence, requiring models to predict human actions and their social consequences.
6. **ARC Challenge (ARC-c)** (Clark et al., 2018) includes difficult multiple-choice science questions that demand deeper logical inference beyond statistical co-occurrence.
7. **BoolQ** (Clark et al., 2019) consists of naturally occurring yes/no questions, requiring models to infer relevant information from provided contexts.
8. **WinoGrande** (Sakaguchi et al., 2021) assesses commonsense knowledge through binary-choice sentence completion tasks that require resolving ambiguities.

The **MetaMathQA** dataset (Yu et al., 2024) constructs mathematical questions by reformulating them from different viewpoints while preserving their original knowledge content. We assess its performance using two well-established benchmarks: (1) **GSM8K** (Cobbe et al., 2021), a collection of grade-school-level math problems requiring step-by-step reasoning to reach a solution, and (2) **MATH** (Hendrycks et al., 2021), which consists of high-difficulty, competition-style problems designed to test advanced mathematical skills.

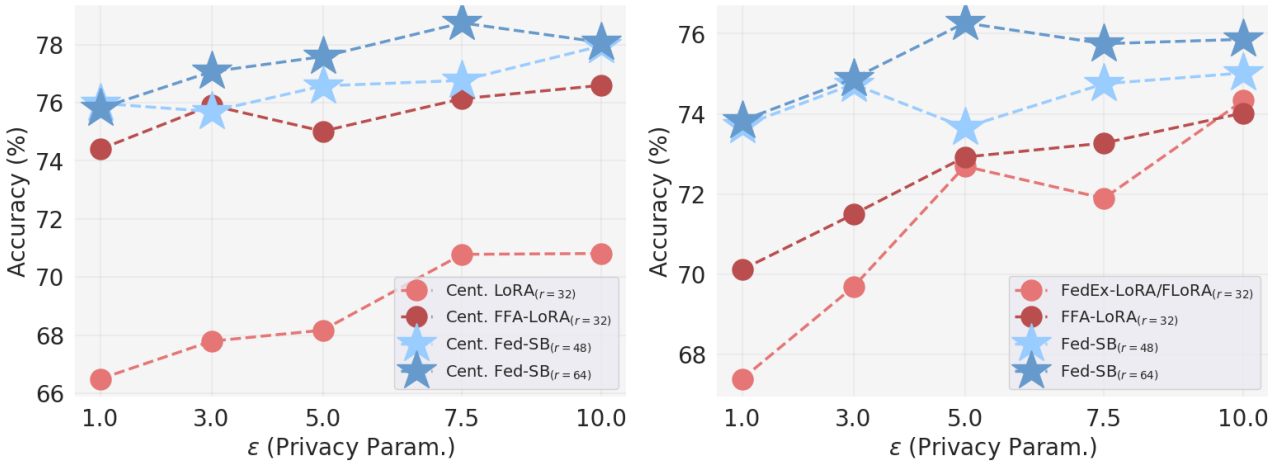
Stanford Natural Language Inference (SNLI) is a widely used benchmark for assessing textual entailment models in natural language understanding. It contains approximately 570,000 sentence pairs, each categorized into one of three classes: entailment, contradiction, or neutral, requiring models to infer the relationship between a given premise and hypothesis.

G. Additional Plots



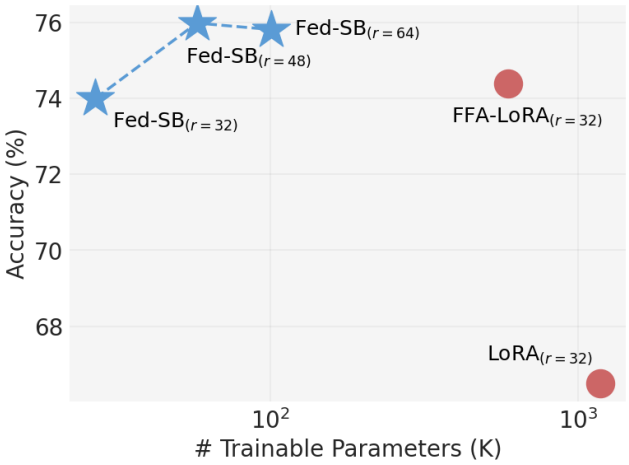
(a) Mistral-7B (GSM8K) (b) Gemma-2 9B (MATH) (c) Llama-3.2 3B (Commonsense)

Figure 3: Performance vs. number of communicated parameters (in log scale) for various methods in federated fine-tuning across multiple models on arithmetic and commonsense reasoning tasks.

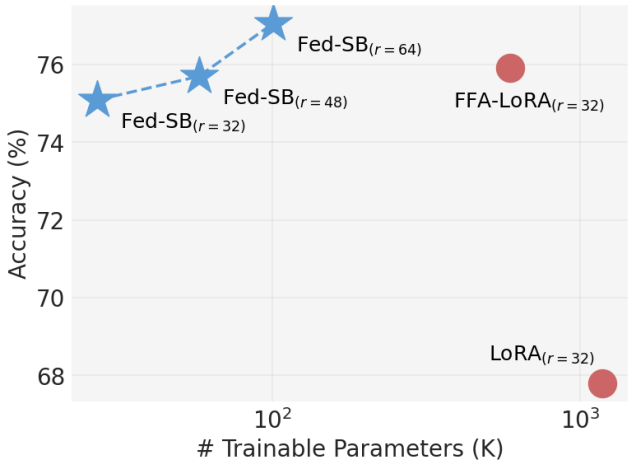


(a) Centralized Private (b) Federated Private

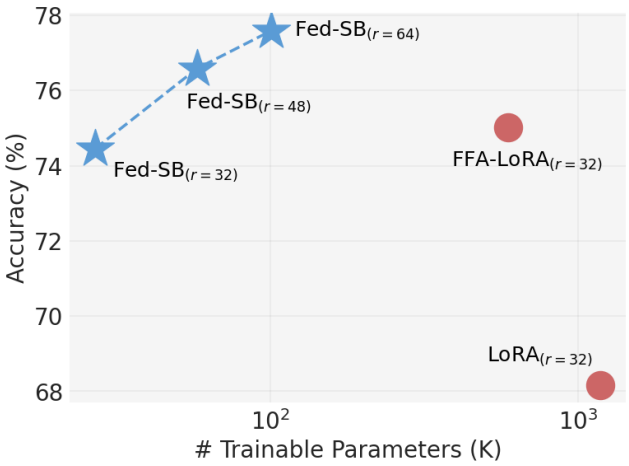
Figure 4: Performance comparison of various methods in centralized (Cent.) private and federated private fine-tuning (BERT-base) on SNLI across varying values of ϵ.



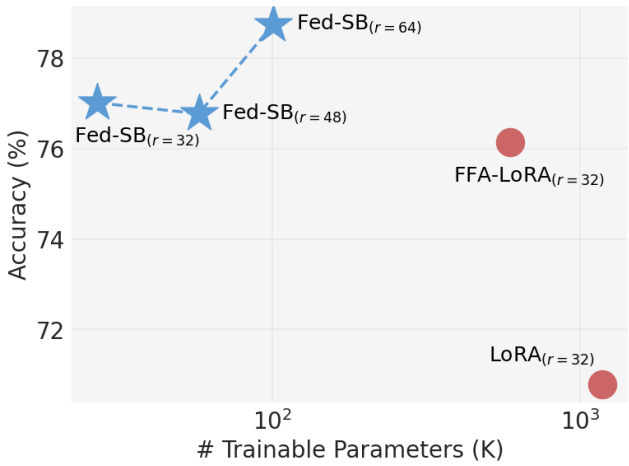
(a) $\epsilon = 1$



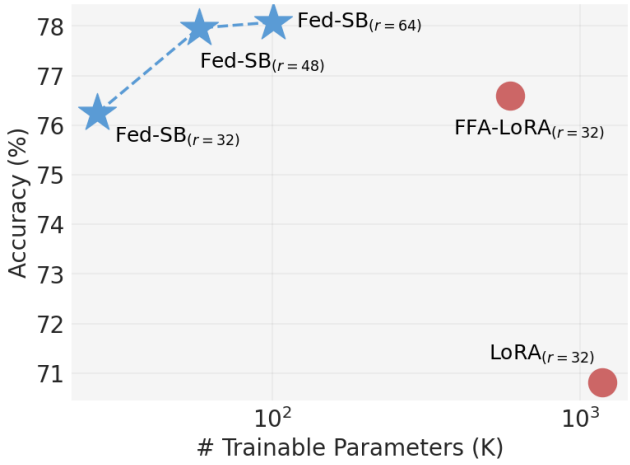
(b) $\epsilon = 3$



(c) $\epsilon = 5$

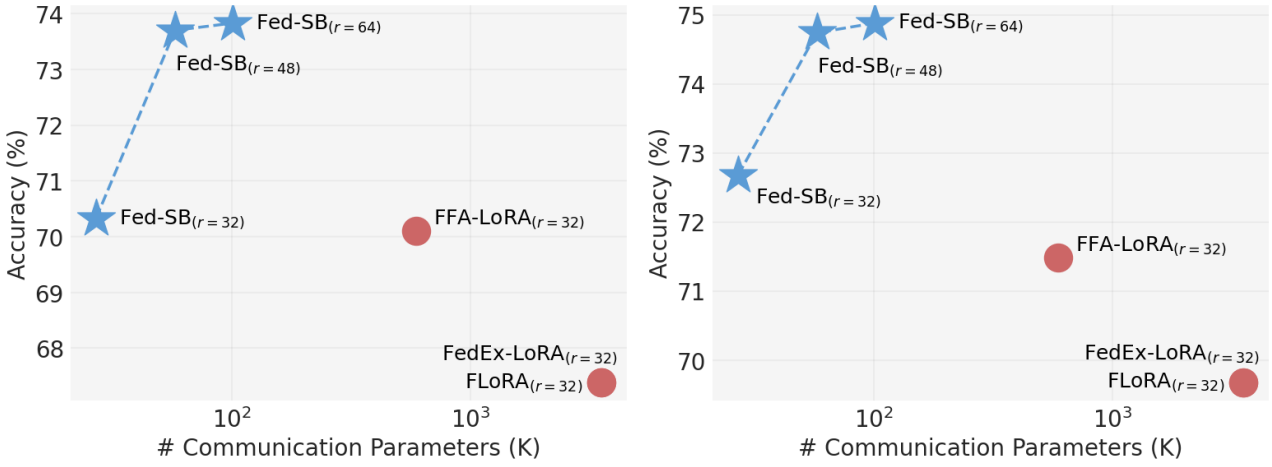


(d) $\epsilon = 7.5$



(e) $\epsilon = 10$

Figure 5: Performance vs. number of trainable parameters (in log scale) for various methods in centralized private fine-tuning (BERT-base) across different privacy budgets (ϵ).



(a) $\epsilon = 1$

(b) $\epsilon = 3$

(c) $\epsilon = 5$

(d) $\epsilon = 7.5$

(e) $\epsilon = 10$

Figure 6: Performance vs. number of communicated parameters (in log scale) for various methods in federated private fine-tuning (BERT-base) across different privacy budgets (ϵ).