Federated Sketching LoRA: On-Device Collaborative Fine-Tuning of Large Language Models

Wenzhi Fang[1], **Dong-Jun Han**[2], **Liangqi Yuan**[1],
Seyyedali Hosseinalipour[3], **Christopher G. Brinton**[1]

Abstract

Fine-tuning large language models (LLMs) on devices remains a challenging problem. Recent works have fused low-rank adaptation (LoRA) techniques with federated fine-tuning to mitigate challenges associated with device model sizes and data scarcity. Still, the heterogeneity of resources remains a critical bottleneck: while higher-rank modules generally enhance performance, varying device capabilities constrain LoRA's feasible rank range. Existing approaches attempting to resolve this issue either lack analytical justification or impose additional computational overhead, leaving a wide gap for efficient and theoretically-grounded solutions. To address these challenges, we propose federated sketching LoRA (FSLoRA), which leverages a sketching mechanism to enable devices to selectively update submatrices of global LoRA modules maintained by the server. By adjusting the sketching ratios, which determine the ranks of the submatrices on the devices, FSLoRA flexibly adapts to device-specific communication and computational constraints. We provide a rigorous convergence analysis of FSLoRA that characterizes how the sketching ratios affect the convergence rate. Through comprehensive experiments on multiple datasets and LLM models, we demonstrate FSLoRA's performance improvements compared to various baselines. The code is available at `https://github.com/wenzhifang/Federated-Sketching-LoRA-Implementation`.

1 Introduction

On-device large language models (LLMs) have recently gained significant attention as a promising complement to cloud-based LLMs [10]. They align with the typical paradigm of LLMs: starting from a base model pre-trained on large-scale datasets to learn general linguistic patterns, semantics, and context, and then undergoing fine-tuning on task-specific data to enhance performance on specialized or domain-specific applications. However, an LLM fine-tuned on a single client device often achieves unsatisfactory performance due to the limited data available on each device. Federated learning [29, 4] has been investigated as a potential solution here, enabling the model to be fine-tuned across a distributed group of clients within the same task domain, without any data sharing.

However, federated learning imposes significant computational and memory costs, as each device must fine-tune the LLM using its local dataset and send updates to the server for model aggregation. Recently, many parameter-efficient fine-tuning methods have been proposed [23, 24, 15] to reduce the cost associated with model adaptation. Among them, low-rank adaptation (LoRA) [15] stands out as a particularly effective approach due to its flexibility. LoRA enables efficient fine-tuning by approximating weight updates $\Delta \mathbf{W}$ through a low-rank decomposition $\Delta \mathbf{W} = \mathbf{B}\mathbf{A}$, where matrices $\mathbf{B}$ and $\mathbf{A}$ contain significantly fewer trainable parameters than the original weight matrix. To support distributed on-device LLM, Zhang et al. [43], Ye et al. [39] incorporated LoRA into conventional federated averaging (FedAvg) [29], significantly reducing the fine-tuning cost by cutting down the number of parameters that need to be synchronized across distributed devices.

[1]Purdue University, [2]Yonsei University, [3]University at Buffalo-SUNY.
Correspondence to: `fang375@purdue.edu`

Challenges. While integrating federated learning with LoRA reduces the number of trainable parameters via matrix decomposition, *computation and communication costs are still forced to increase with the decomposition rank*. This poses challenges when complex tasks demand higher-rank LoRA modules, particularly on resource-constrained mobile devices. Furthermore, the *heterogeneity in resource availability across distributed devices makes a uniform rank inefficient*: a fixed rank r may be too large for some constrained devices, while being too small for more powerful ones, resulting in underutilized resources. Consequently, an approach that reduces computation and communication overhead while adapting LoRA ranks to heterogeneous device capabilities is highly desirable for collaborative fine-tuning of LLMs. Although some existing approaches have attempted to provide a solution here [5, 1, 37], they either lack theoretical justification or impose additional computational overhead, leaving a large gap for an efficient and theoretically-grounded solution. As we discuss in Section 2.2, a comprehensive approach that preserves the analytical and practical benefits of LoRA while enabling heterogeneous on-device fine-tuning under tight resource constraints remains elusive.

1.1 Contributions

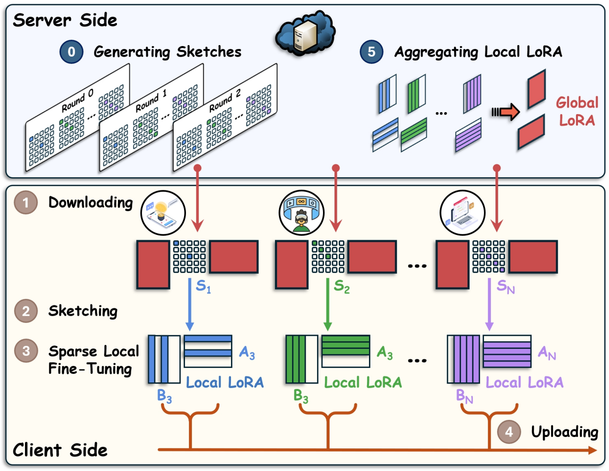


Figure 1: An illustration of our proposed methodology where the server maintains a pair of global LoRA modules while the devices adaptively update submatrices of the global LoRA modules through sketching during each round.

Motivated by these limitations, this work develops a methodology for collaborative on-device LLM fine-tuning that (i) retains the flexibility of LoRA, (ii) provides theoretical convergence guarantees, and (iii) addresses the challenges posed by system heterogeneity and resource constraints across distributed devices. As depicted in Figure 1, our key idea is to introduce a sketching-based LoRA update to the local fine-tuning stage, which allows devices to selectively update a subset of columns and rows of the LoRA modules during each round, reducing the computation and communication consumption. Additionally, our method customizes the fine-tuning process by adjusting the sparsity level of the sketching matrix, i.e., the size of the updated submatrices for each device in each iteration. As we will see, the impact of the introduced sketching mechanism on the overall optimization landscape requires careful modeling consideration, posing additional challenges for the theoretical analysis that we address in this work.

Overall, we make the following contributions:

- We propose federated sketching LoRA (FSLoRA), which leverages a sketching mechanism to enable devices to selectively update submatrices of global LoRA modules maintained by the server. By adjusting the sketching ratios, which determine the ranks of the submatrices on devices, FSLoRA effectively adapts to device-specific communication and computational constraints.
- We present a rigorous convergence analysis of FSLoRA under non-uniform submatrix update scenarios (i.e., heterogeneous LoRA configurations) across devices, revealing how the sketching ratios affect the convergence rate via scaled smoothness constants. Further, our results show that while increasing the sketching ratios improves convergence theoretically, it also raises communication and computation costs, suggesting a potential trade-off in selecting the sketching ratios.
- We conduct extensive experiments across multiple datasets and LLM models with diverse parameter settings, demonstrating FSLoRA's superior performance compared to various baselines in accuracy, training time, and resource utilization. Our ablation studies further validate the effectiveness of the sketching mechanism and the ability of devices to exploit larger global ranks under FSLoRA.

1.2 Related Works

LoRA-based parameter-efficient fine-tuning: LoRA was introduced in [15] as a parameter-efficient alternative to full model fine-tuning via low-rank matrix approximations. Subsequently, Kalajdzievski [17] proposed rank-stabilized LoRA (rsLoRA), an approach that enhances LoRA's performance in high-rank scenarios by modifying the scaling factor applied to each low-rank product. Shuttleworth et al. [33] demonstrated that with this design, rsLoRA can approach the performance of full model fine-tuning as the rank of LoRA modules increases. Han et al. [14] introduced a sparse matrix in parallel with LoRA modules to improve the overall adaptation capability. In [25, 28, 38], the authors proposed sequential low-rank adaptation schemes, including ReLoRA and CoLA, to enable high-rank fine-tuning through iterative low-rank updates. However, the works mentioned above focus on centralized scenarios, assuming that the data required for fine-tuning is available at the server.

Collaborative fine-tuning via federated LoRA: Federated LoRA has recently gained attention as a promising approach for efficient and collaborative fine-tuning of LLMs across distributed devices [4, 13]. Sun et al. [34] examined the performance of federated LoRA incorporating differential privacy. To reduce client-server communication overhead, Kuo et al. [22] proposed integrating communication compression with federated LoRA. Meanwhile, Bai et al. [1], Cho et al. [5], Byun and Lee [3], Wang et al. [37], Koo et al. [21] explored the challenges of resource heterogeneity among distributed devices and introduced heterogeneous LoRA as a solution. However, the approaches proposed in [5, 21, 3, 1] lack a theoretical foundation. Moreover, the FlexLoRA method introduced in [1] incurs additional computational overhead due to its reliance on singular value decomposition (SVD). Furthermore, the stack LoRA method proposed in [37] requires the devices to integrate the LoRA modules into the base model, thereby compromising the inherent flexibility of LoRA. Overall, there is still a lack of a systematic and theoretically grounded solution that can effectively tackle the challenges of heterogeneity in collaborative on-device LLM fine-tuning.

Enhancing adaptability via higher-rank LoRA modules: In [15], the authors showed that small ranks can be sufficient for certain tasks; however, they also acknowledge that small rank LoRA modules may not work universally, especially when the downstream task differs significantly from pretraining. Following this, several works explored the effect of increasing the rank in LoRA modules. In a centralized setup, Kalajdzievski [17] and Shuttleworth et al. [33] showed that higher-rank LoRA models can closely approximate full fine-tuning under rsLoRA. In a distributed on-device LLM fine-tuning regime, Bai et al. [1] demonstrated improved performance with larger ranks under FlexLoRA. Similarly, Cho et al. [5] reported that, with proper overfitting control, HeteroLoRA can also benefit from larger ranks. Overall, while small ranks may suffice for simpler tasks or strong base models, higher-rank modules are necessary to compensate for limited backbone capability, such as in on-device LLMs, and to enable effective adaptation to more complex downstream tasks.

2 Problem Background

2.1 LoRA-based Federated LLM Fine-tuning

The federated LoRA fine-tuning problem can be formulated as

$$\min_{\mathbf{B},\mathbf{A}} f(\mathbf{B},\mathbf{A}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{B},\mathbf{A}), \text{ where } f_i(\mathbf{B},\mathbf{A}) := \mathbb{E}_{\xi\sim\mathcal{D}_i}\left[\ell(\mathbf{W}_0+\mathbf{B}\mathbf{A},\xi)\right], \tag{1}$$

$\mathbf{W}_0$ denotes the frozen base model, $\mathbf{B} \in \mathbb{R}^{m\times r}, \mathbf{A} \in \mathbb{R}^{r\times n}$ are LoRA modules, N denotes the number of devices, ξ denotes a data sample, and $\mathcal{D}_i$ is the local dataset on device i. ℓ, f_i, and f are the sample loss function, the local loss for device i, and the global loss, respectively.

Problem (1) aligns with the conventional federated optimization formulation, which thus can be solved using the FedAvg algorithm. Based on the FedAvg framework, Zhang et al. [43] developed federated LoRA, which applies a uniform rank r across all devices, overlooking system heterogeneity. This one-size-fits-all approach leads to resource mismatches, where computationally constrained devices may struggle, while more powerful devices remain underutilized with a fixed rank.

2.2 Aren't the Existing Solutions Good Enough?

To address this issue, researchers have proposed heterogeneous federated LoRA approaches, where devices maintain non-uniform LoRA modules with varying ranks. They also introduce mechanisms to overcome the challenges of directly aggregating matrices with different dimensions. However, these methods often lack theoretical foundation or incur additional computational and memory overhead.

HeteroLoRA [5] lets the server pad the updates from the devices with smaller ranks to match the size of the largest rank during aggregation. During model dissemination, devices receive a truncated version of the global LoRA modules from the server. Although easy to implement, HeteroLoRA is primarily heuristic in nature and lacks a rigorous theoretical foundation, potentially limiting its performance, as we will see in Section 5.

FlexLoRA [1] requires the server to collect the individual LoRA matrices $\mathbf{B}_i$ and $\mathbf{A}_i$ from the devices and then computes their product $\mathbf{B}_i\mathbf{A}_i$. To support the initialization of non-uniform LoRA modules, the server applies truncated SVD to the averaged product $\frac{1}{N}\sum_{i=1}^{N}\mathbf{B}_i\mathbf{A}_i$. However, this approach introduces extra computational and memory overhead on the server due to truncated SVD, and the associated error can limit the performance as demonstrated in Section 5.

FedStackLoRA [37] introduces a stacking mechanism where the server concatenates LoRA modules from the devices. The concatenated matrices are then sent back to the devices, which compute their product and merge it into the base model before initializing new LoRA modules for the next fine-tuning round. However, this approach increases communication complexity linearly with the number of devices, imposes higher computation and memory demands on the devices, and compromises LoRA's flexibility to support multiple adapters for different tasks.

More detailed comparisons on computation, memory, and communication are presented in Appendix A. In summary, a theoretically-grounded solution that preserves LoRA's benefits while effectively addressing system heterogeneity and the constraints of resource-limited devices remains lacking.

3 Federated Sketching LoRA

Motivated by the limitations of existing methods, we propose a new federated LoRA reformulation. Building on this foundation, we develop FSLoRA, a resource-adaptive algorithm that preserves LoRA's flexibility while accommodating for heterogeneous device capabilities.

3.1 Our Formulation

We propose a sketching-based LoRA formulation for collaborative LLM fine-tuning as follows:

$$\min_{\mathbf{B},\mathbf{A}} f^{\mathcal{S}}(\mathbf{B},\mathbf{A}) \coloneqq \frac{1}{N}\sum_{i=1}^{N} f_i^{\mathcal{S}}(\mathbf{B},\mathbf{A}) \text{ where } f_i^{\mathcal{S}}(\mathbf{B},\mathbf{A}) \coloneqq \mathbb{E}_{\mathbf{S}\sim\mathcal{S}_i;\xi\sim\mathcal{D}_i}\left[\ell(\mathbf{W}_0+\mathbf{BSA},\xi)\right], \quad (2)$$

where $\mathbf{B}\in\mathbb{R}^{m\times r}, \mathbf{A}\in\mathbb{R}^{r\times n}$ are LoRA modules, $f_i^{\mathcal{S}}$ is the local loss function at device i with sketching, and $\mathbf{S}$ denotes a sketching matrix randomly sampled from the diagonal matrix set $\mathcal{S}_i = \mathcal{S}(r,k_i)$. The set $\mathcal{S}(r,k_i)$ comprises diagonal matrices of size $r\times r$ with exactly k_i non-zero entries. The formal definition of $\mathcal{S}(r,k)$ is provided below:

Definition 3.1 (Random-k sketching)**.** A random-k diagonal matrix set is defined as:

$$\mathcal{S}(r,k)=\left\{\mathbf{S}\,|\,\mathbf{S}=\frac{r}{k}\sum_{j\in\mathcal{I}}\mathbf{e}_j\mathbf{e}_j^\top,\, \mathcal{I}\subseteq\{1,\ldots,r\},\, |\mathcal{I}|=k\right\},$$

where $\mathbf{e}_1,\ldots,\mathbf{e}_r\in\mathbb{R}^r$ are standard unit basis vectors and index set $\mathcal{I}$ is a random subset of $[r]\coloneqq\{1,2,\ldots,r\}$ sampled uniformly from all subsets of $[r]$ with cardinality k.

With $\mathbf{S}$ being a matrix sampled from $\mathcal{S}_i$, we have $\mathbf{BSA}=\frac{r}{k_i}\sum_{j\in\mathcal{I}_i}\mathbf{B}\mathbf{e}_j\mathbf{e}_j^\top\mathbf{A}$, where $\mathcal{I}_i$ corresponds to the index set of non-zero entries of $\mathbf{S}$. $\mathbf{B}\mathbf{e}_j$ extracts the j-th column of $\mathbf{B}$ while $\mathbf{e}_j^\top\mathbf{A}$ extracts the j-th row of $\mathbf{A}$. In other words, only k_i columns and rows in the LoRA modules $\mathbf{B}$ and $\mathbf{A}$ are activated by the sketching matrix in the loss $\ell(\mathbf{W}_0+\mathbf{BSA},\xi)$ at device i. On the other hand, the sketching matrix $\mathbf{S}$ satisfies $\mathbb{E}_{\mathbf{S}\sim\mathcal{S}_i}[\mathbf{S}]=\mathbf{I}_r$ where $\mathbf{I}_r$ is a r-dimensional identity matrix. Based upon this property, $\mathbf{W}_0+\mathbf{BSA}$ can be treated as an unbiased estimate of $\mathbf{W}_0+\mathbf{BA}$.

Intuition: A larger rank allows LoRA modules to be more expressive, leading to better performance [1, 17, 33]. However, resource-constrained devices cannot afford the computational or communication demands of large-rank modules. Our formulation (2) leverages the sketching matrix to balance the expressiveness of high-rank LoRA modules with the resource constraints of different devices. With the sketching mechanism introduced, the local gradients with respect to the LoRA modules on the devices will exhibit structured sparsity. By adjusting the sketching ratio k_i/r, we can tailor the sparsity of the gradient to match the capabilities of each device, ensuring affordable training while maintaining performance across heterogeneous systems, as elaborated in the following subsection. Overall, compared to the original objective in (1), our formulation offers a more resource-adaptive and flexible framework, tailored to address the diverse capabilities of heterogeneous devices.

3.2 Sparsity in the Gradients

In this subsection, we analyze the gradient structure of LoRA modules and highlight the gradients' sparsity properties under a given sketching matrix. To begin, we present the gradient expressions for the LoRA modules $\mathbf{B}$ and $\mathbf{A}$ in the following lemma. The proof is provided in Appendix F.2.

Lemma 3.2 (Gradient Formulation)**.** *For a given sketching matrix* $\mathbf{S}$*, the gradients of* $\ell(\mathbf{W_0} + \mathbf{BSA}, \xi)$ *with respect to* $\mathbf{B}$ *and* $\mathbf{A}$ *take the following form*

$$\begin{aligned} \nabla_{\mathbf{B}}\ell(\mathbf{W}_0 + \mathbf{BSA}, \xi) &= \nabla\ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)\mathbf{A}^\top\mathbf{S}^\top \\ \nabla_{\mathbf{A}}\ell(\mathbf{W}_0 + \mathbf{BSA}, \xi) &= \mathbf{S}^\top\mathbf{B}^\top\nabla\ell(\mathbf{W_0} + \mathbf{BSA}, \xi), \end{aligned} \tag{3}$$

where $\nabla_{\mathbf{B}}\ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)$*,* $\nabla_{\mathbf{A}}\ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)$*, and* $\nabla\ell(\mathbf{W_0} + \mathbf{BSA}, \xi)$ *represent the gradients of* $\ell(\mathbf{W_0} + \mathbf{BSA}, \xi)$ *with respect to* $\mathbf{B}$*,* $\mathbf{A}$*, and* $\mathbf{W_0} + \mathbf{BSA}$*, respectively.*

In particular, a random-k diagonal sketching matrix selectively samples k rows or columns of a matrix through left product or right product, respectively. With $\mathbf{S}$ being a random-k diagonal matrix, the gradients of $\ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)$ with respect to LoRA modules $\mathbf{B}$ and $\mathbf{A}$, as shown in (3), naturally become structurally sparse matrices. This sparsity reduces the computational and memory overhead during training, allowing for faster gradient computation and parameter updates. Additionally, sparse training enables better scalability across distributed devices by reducing communication costs, as only the non-zero elements need to be transmitted.

Remark 3.3 (Sparsity Level Control)**.** A key advantage of our formulation is its flexible control over the sparsity level of local gradients, achieved by configuring the parameter k_i of the sketching matrix set $\mathcal{S}_i = \mathcal{S}(r, k_i)$. This mechanism allows each device to tailor its local updates according to its communication and computation resource constraints, ensuring efficient and scalable fine-tuning in heterogeneous federated systems. Lowering k_i helps resource-constrained devices reduce computation and communication overhead, while more capable devices can increase k_i to conduct more informative local updates. Additionally, the distinction in sparsity level control between the proposed FSLoRA and the FedBCGD algorithm [26] is elaborated in Appendix B.

3.3 FSLoRA Algorithm

Based on the formulation in (2), we propose a resource-adaptive algorithm termed FSLoRA for collaborative on-device fine-tuning. FSLoRA allows each device to update submatrices of the original modules $\mathbf{B}$ and $\mathbf{A}$ in each round. The server maintains a pair of global LoRA modules $\mathbf{B}$ and $\mathbf{A}$ and periodically updates them by aggregating sparse local updates received from distributed devices. Specifically, the procedure of FSLoRA at each round t is detailed below.

- The server begins by sampling sketching matrices $\{\mathbf{S}_i^t \sim \mathcal{S}_i\}_{i=1}^N$ for all devices, where $\mathcal{S}_i$ represents the set of possible sketching matrices for device i. These sketches are then sent to the corresponding devices. Additionally, the server broadcasts the current global LoRA modules $[\mathbf{B}^t; \mathbf{A}^t]$ to all devices. Note that the communication load introduced by transmitting the sketching matrix is negligible compared to global LoRA modules, as it involves only *binary sketching indices* (i.e., the diagonal elements of the sketching matrix); see Appendix A for details.
- Devices perform local fine-tuning using sketch $\mathbf{S}_i^t$. Specifically, guided by sketching matrix $\mathbf{S}_i^t$, the update at device i during the h-th iteration of the t-th round is given by:

  $$\left[\mathbf{B}_i^{t,h+1}; \mathbf{A}_i^{t,h+1}\right] = \left[\mathbf{B}_i^{t,h}; \mathbf{A}_i^{t,h}\right] - \gamma\left[\Delta\mathbf{B}_i^{t,h}(\mathbf{S}_i^t)^\top; (\mathbf{S}_i^t)^\top\Delta\mathbf{A}_i^{t,h}\right], \tag{4}$$

  where γ denotes the learning rate and $[\Delta\mathbf{B}_i^{t,h}; \Delta\mathbf{A}_i^{t,h}]$ is a shorthand representation for:

  $$\left[\Delta\mathbf{B}_i^{t,h}; \Delta\mathbf{A}_i^{t,h}\right] = \left[\nabla\ell(\mathbf{W}_0 + \mathbf{B}_i^{t,h}\mathbf{S}_i^t\mathbf{A}_i^{t,h}, \xi_i^{t,h})(\mathbf{A}_i^{t,h})^\top; (\mathbf{B}_i^{t,h})^\top\nabla\ell(\mathbf{W}_0 + \mathbf{B}_i^{t,h}\mathbf{S}_i^t\mathbf{A}_i^{t,h}, \xi_i^{t,h})\right].$$

  The update direction in (4) corresponds to the negative stochastic gradient of $\ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)$ with respect to $[\mathbf{B}; \mathbf{A}]$ for a given sketch $\mathbf{S}_i^t$, as established in Lemma 3.2. The total update for device i during one round of training, consisting of H local steps, can be expressed as follows:

  $$\left[\mathbf{B}_i^{t,H} - \mathbf{B}_i^{t,0}; \mathbf{A}_i^{t,H} - \mathbf{A}_i^{t,0}\right] = \left[\gamma\left(\sum_{h=0}^{H-1}\Delta\mathbf{B}_i^{t,h}\right)(\mathbf{S}_i^t)^\top; \gamma(\mathbf{S}_i^t)^\top\left(\sum_{h=0}^{H-1}\Delta\mathbf{A}_i^{t,h}\right)\right].$$

  From the above equation, we see that only the columns of $\mathbf{B}$ and the rows of $\mathbf{A}$ corresponding to the nonzero entries of $\mathbf{S}_i^t$ are updated during the t-th round at device i. In essence, $\mathbf{S}_i^t$ selectively activates specific columns of $\mathbf{B}$ and rows of $\mathbf{A}$ for each round. Afterward, devices transmit these nonzero columns and rows of the sparse model updates to the server.

Algorithm 1 Federated Sketching LoRA (FSLoRA)

```
Require: Base model $\mathbf{W}_0$, LoRA modules $\mathbf{B}_0$ and $\mathbf{A}_0$, learning rate $\gamma$, and sketching set $\{\mathcal{S}_i\}_{i=1}^N$
for $t = 0, 1, \dots, T-1$ do
    Server samples sketching matrices $\{\mathbf{S}_i^t \sim \mathcal{S}_i\}_{i=1}^N$ and sends them back to the devices
    Server broadcasts the current global LoRA modules to the devices
    for $h = 0, 1, \dots, H-1$ do
        Devices update the local LoRA modules via (4)
    end for
    Devices upload the non-zero columns of $(\mathbf{B}_i^{t,H} - \mathbf{B}_i^{t,0})$ and the non-zero rows $(\mathbf{A}_i^{t,H} - \mathbf{A}_i^{t,0})$ to the server
    Server updates the global LoRA modules via (5)
end for
```

- Using the sketch information, the server reconstructs the corresponding sparse matrices from the received updates and aggregates them to update the global model:

$$\left[\mathbf{B}^{t+1}; \mathbf{A}^{t+1}\right] = \left[\mathbf{B}^t; \mathbf{A}^t\right] + \frac{1}{N}\sum_{i=1}^{N}\left[\mathbf{B}_i^{t,H} - \mathbf{B}_i^{t,0}; \mathbf{A}_i^{t,H} - \mathbf{A}_i^{t,0}\right]. \tag{5}$$

The above procedure is repeated for $t = 0, 1, \dots, T-1$ across T rounds. Algorithm 1 summarizes the overall process of FSLoRA.

Remark 3.4 (Aggregation)**.** Existing works on federated LoRA primarily adopt two aggregation strategies: (1) aggregating the LoRA modules as $[\mathbf{B}; \mathbf{A}]$ (e.g., vanilla Federated LoRA [43]), and (2) aggregating the product $\mathbf{BA}$ (e.g., FlexLoRA [1]). Both methods have demonstrated effectiveness, as evidenced by their promising performance in prior studies. In this work, we adopt the former, as it introduces minimal overhead and retains the simplicity of LoRA. Additionally, we establish the convergence of Algorithm 1 under this aggregation choice, as shown in Section 4.

Remark 3.5 (Computation, memory, and communication)**.** The proposed FSLoRA introduces no additional operations compared to the vanilla Federated LoRA [43], resulting in minimal overhead for both the server and the devices relative to other heterogeneous LoRA baselines [1, 37]. A more detailed comparison of computation, memory, and communication is provided in Appendix A.

3.4 Comparison with Communication Compression

Although both the sketching approach in FSLoRA and communication compression [22] reduce communication overhead, the sketching approach fundamentally differs from traditional compression techniques. Compression methods focus solely on reducing the transmission load, leaving the gradient computation and model updates unchanged from the vanilla Federated LoRA. FSLoRA goes beyond communication savings by also reducing gradient computation and model update overhead through sparse training. Notably, these two methods are orthogonal and can be combined to achieve greater efficiency. Specifically, the compression can be applied to the transmission of non-zero columns of $\mathbf{B}$ and the non-zero rows of $\mathbf{A}$ in FSLoRA to further enhance communication efficiency. We demonstrate the effectiveness of this combination in Appendix E.2.

4 Analysis

In this section, we analyze the convergence of the proposed FSLoRA algorithm. We show that the iterate sequence generated by FSLoRA algorithm converges to a stationary point of the function (2). Our analysis relies on the following notations.

Notations: We define $\tilde{\ell}(\mathbf{B}, \mathbf{A}, \xi; \mathbf{S}) = \ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)$ and $\tilde{f}_i(\mathbf{B}, \mathbf{A}; \mathbf{S}) = \mathbb{E}_{\xi \sim \mathcal{D}_i}\left[\ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)\right]$ for a given $\mathbf{S}$ and $f_i^{\mathcal{S}}(\mathbf{B}, \mathbf{A}) = \mathbb{E}_{\mathbf{S} \sim \mathcal{S}_i}[\tilde{f}_i(\mathbf{B}, \mathbf{A}; \mathbf{S})]$. For simplicity, we denote $\mathbf{X} = [\mathbf{B}; \mathbf{A}]$ and rewrite $f(\mathbf{B}, \mathbf{A})$, $f_i(\mathbf{B}, \mathbf{A})$, $f^{\mathcal{S}}(\mathbf{B}, \mathbf{A})$, $f_i^{\mathcal{S}}(\mathbf{B}, \mathbf{A})$, $\tilde{f}_i(\mathbf{B}, \mathbf{A}; \mathbf{S})$, and $\tilde{\ell}(\mathbf{B}, \mathbf{A}, \xi; \mathbf{S})$ as $f(\mathbf{X})$, $f_i(\mathbf{X})$, $f^{\mathcal{S}}(\mathbf{X})$, $f_i^{\mathcal{S}}(\mathbf{X})$, $\tilde{f}_i(\mathbf{X}; \mathbf{S})$, and $\tilde{\ell}(\mathbf{X}, \xi; \mathbf{S})$ respectively. In addition, we use $\|\cdot\|$ to denote the Frobenius norm.

We conduct analysis based on the following assumptions.

Assumption 4.1. $f_i(\mathbf{X})$ is differentiable and L-smooth, i.e., there exists a positive constant L such that $\forall \mathbf{X}, \mathbf{Y}$,

$$\|\nabla f_i(\mathbf{X}) - \nabla f_i(\mathbf{Y})\| \leq L\|\mathbf{X} - \mathbf{Y}\|, \forall i.$$

Assumption 4.2. $\nabla_{\mathbf{X}} \tilde{\ell}(\mathbf{X}, \xi; \mathbf{S})$ is an unbiased estimate of $\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X})$ and its variance is bounded as

$$\mathbb{E}\|\nabla_{\mathbf{X}} \tilde{\ell}(\mathbf{X}, \xi; \mathbf{S}) - \nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X})\|^2 \leq \rho\|\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X})\|^2 + \sigma^2, \forall i,$$

where the expectation is computed over $\xi \sim \mathcal{D}_i$ and $\mathbf{S} \sim \mathcal{S}_i$.

Assumption 4.3. The gradient dissimilarity between the global loss $f^{\mathcal{S}}(\mathbf{X})$ and each local loss $f_i^{\mathcal{S}}(\mathbf{X})$ satisfies

$$\left\|\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}) - \nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X})\right\|^2 \leq c_h \|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X})\|^2 + \delta_h^2, \forall i,$$

where $c_h \geq 0$ and $f^{\mathcal{S}}(\mathbf{X}) = \frac{1}{N}\sum_{i=1}^{N} f_i^{\mathcal{S}}(\mathbf{X})$.

Assumptions 4.1 and 4.2 are standard in stochastic optimization [8, 12], while Assumption 4.3 is commonly used in distributed optimization [11, 40] to characterize data heterogeneity. Building on these assumptions, we analyze the convergence behavior of FSLoRA. Our main results are summarized in the following theorem.

Theorem 4.4. *Suppose that Assumptions 4.1-4.3 hold and the learning rate satisfies* $\gamma \leq \min\{\frac{N}{24\rho(c_h+1)H\bar{L}}, \frac{1}{8\sqrt{\widetilde{L}L(\rho+1)(c_h+1)H}}\}$. *Then the iterates* $\{\mathbf{X}^t\}_{t=0}^{T-1}$ *generated by FSLoRA satisfy*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 \leq 4\frac{f^{\mathcal{S}}(\mathbf{X}^0) - f^*}{\gamma T H} + \gamma\frac{4\bar{L}}{N}\sigma_\rho^2 + 10\gamma^2 H^2 \widetilde{L} L \sigma_\rho^2, \tag{6}$$

where $\sigma_\rho^2 = \sigma^2 + 3(\rho+1)\sigma_h^2$, $\bar{L} = \left(\frac{1}{N}\sum_{i=1}^{N}\frac{r}{k_i}\right)L$, $\widetilde{L} = \left(\frac{1}{N}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\right)L$, *and* f^* *denotes the lower bound of* $f^{\mathcal{S}}(\mathbf{X})$.

Technical highlights of Theorem 4.4: A key step in the proof of Theorem 4.4 is characterizing the impact of sketching mechanism on the optimization landscape. Our analysis reveals how the sketching operation modifies the smoothness properties of the objective, introducing scaled smoothness constants, $\frac{r}{k_i}L$ and $\frac{r^2}{k_i^2}L$, which directly influence the convergence behavior. Further details are presented in Appendix F.3.

Based on the results in Theorem 4.4, we obtain the following corollary by applying an appropriate learning rate γ to Algorithm 1. The proof can be found in Appendix F.4.

Corollary 4.5. *Under the same assumptions of Theorem 4.4, let* $\mathcal{F}_0 = f^{\mathcal{S}}(\mathbf{X}^0) - f^*$. *Then there exists a learning rate* γ *meeting the condition outlined in Theorem 4.4 such that the iterates* $\{\mathbf{X}^t\}_{t=0}^{T-1}$ *generated by FSLoRA satisfy*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 \leq 8\frac{\sqrt{\bar{L}\mathcal{F}_0\sigma_\rho^2}}{\sqrt{NTH}} + 10(\widetilde{L}L)^{\frac{1}{3}}\left(\frac{\mathcal{F}_0\sigma_\rho}{T}\right)^{\frac{2}{3}} + \frac{4\mathcal{F}_0}{T}. \tag{7}$$

Discussion: Corollary 4.5 establishes an upper bound on the convergence of the proposed FSLoRA algorithm. The parameters $\bar{L}$ and $\widetilde{L}$ provide insight into how the sketching operation influences the convergence rate. Increasing k_i would lead to a faster convergence for FSLoRA. However, this comes at the cost of increased communication and computational overhead for device i, indicating a trade-off in the selection of the sketching ratios. Additionally, the upper bound vanishes as $T \to \infty$. Moreover, the rate at which the bound diminishes is dominated by the first term, which recovers the convergence behavior of FedAvg [41, 19, 18] as the sketching ratio $k_i/r \to 1$(i.e., $\bar{L} = L$). This highlights the tightness of our analysis and shows that FSLoRA retains the convergence guarantees of vanilla Federated LoRA in the limit.

5 Experiments

Our experiments focus on RoBERTa (125M) [27] and LLaMA-3.2-3B [9], which represent typical model sizes suitable for on-device deployment, as well as the LLaMA-7B model to reflect large-scale

Table 5.1: Testing accuracy over 3 independent runs on GLUE and commonsense reasoning benchmarks. FSLoRA achieves a notable improvement in average performance compared to the baselines.

GLUE benchmark (RoBERTa model)

Method	GPU hours	QNLI	MRPC	CoLA	MNLI	RTE	SST-2	QQP	Avg.
HeteroLoRA	10.7h	87.5 ±0.5	84.4 ±0.9	75.3 ±1.2	66.3 ±0.8	69.0 ±1.7	89.5 ±0.0	85.3 ±0.1	79.6
FlexLoRA	12.6h	88.5 ±0.2	81.2 ±0.4	77.5 ±1.2	63.0 ±0.5	62.2 ±1.9	92.8 ±0.4	87.4 ±0.1	78.9
FedStackLoRA	12.3h	87.2 ±0.3	78.1 ±0.7	77.4 ±1.7	74.6 ±0.5	54.4 ±2.1	93.4 ±0.1	87.1 ±0.3	78.9
FSLoRA	10.9h	88.0 ±0.3	87.3 ±0.2	82.2 ±0.5	76.4 ±0.2	69.8 ±1.2	93.5 ±0.1	85.8 ±0.1	83.3

Commonsense reasoning benchmark (LLaMA-3.2-3B model)

Method	GPU hours	ARC-c	ARC-e	BoolQ	HellaSwag	OBQA	PIQA	SIQA	WinoGrande	Avg.
HeteroLoRA	43.7h	73.4 ±0.3	86.6 ±0.2	65.8 ±0.5	73.0 ±0.5	71.4 ±0.3	80.9 ±0.7	73.8 ±0.3	72.0 ±0.3	74.6
FlexLoRA	68.3h	74.2 ±0.3	86.7 ±0.6	68.6 ±0.8	79.4 ±0.7	75.8 ±0.4	81.0 ±0.3	75.9 ±0.4	77.9 ±0.3	77.4
FedStackLoRA	49.8h	68.3 ±0.6	83.1 ±0.5	65.8 ±0.9	77.2 ±0.5	74.2 ±0.3	80.5 ±0.6	76.1 ±0.5	71.5 ±0.5	74.6
FSLoRA	44.3h	76.1 ±0.4	87.2 ±0.5	69.3 ±0.7	82.2 ±1.1	80.7 ±0.6	84.0 ±0.2	76.8 ±0.0	79.1 ±0.2	79.4

scenarios. For RoBERTa and LLaMA-3.2-3B models, we fine-tune and evaluate them on the GLUE [35] and commonsense reasoning benchmark [16], respectively. For the LLaMA-7B model, we utilize Wizard, Dolly-15k, and Alpaca datasets, where the results are reported in Appendix D. Similar to [43, 37], we adopt Dirichlet-based partitioning for dataset splits. All the experiments are conducted on a cluster equipped with 4 NVIDIA A100 GPUs, each with 40 GB of memory. The number of devices is set to 20 in the main manuscript, and to 50 and 100 in Appendix C. Further details are provided in the Appendix G.

5.1 Main Results Under Heterogeneous LoRA Setup

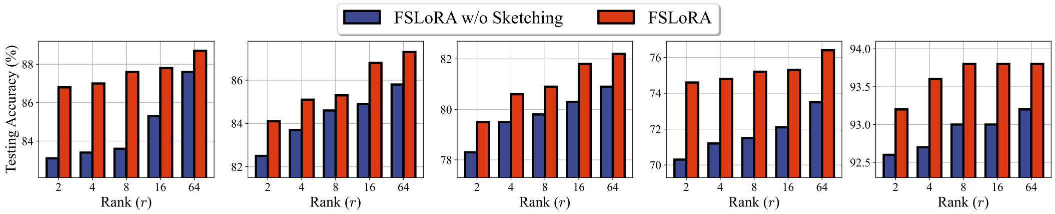
Performance comparison with baselines: We consider three state-of-the-art baselines listed in Section 2.2. For FSLoRA, the rank of the global LoRA modules is fixed as $r = 64$, while the sketching ratio for device i is sampled from the set $\{0.125, 0.25, 0.5, 0.75\}$. For a fair comparison, we apply the same rank configuration to all baseline methods. Table 5.1 presents the performance of FSLoRA and baseline methods. Across both settings, FSLoRA consistently achieves superior accuracy while maintaining low GPU hours. In the GLUE & RoBERTa task, FSLoRA outperforms all baselines on average, with significant gains in MRPC, CoLA, and MNLI. In the commonsense reasoning & LLaMA task, which introduces higher model complexity, FSLoRA also delivers the best overall performance. Notably, FSLoRA achieves this while preserving computational efficiency comparable to HeteroLoRA as reflected in GPU hours. These results highlight FSLoRA's effectiveness and scalability in heterogeneous LoRA fine-tuning scenarios.

Evaluation under broader heterogeneity, increased number of devices, and larger model: In Appendix C, we extend our evaluation to 50 and 100 devices, incorporating greater diversity in devices' communication and computation capabilities, as well as varying levels of data heterogeneity. In Appendix D, we further assess the effectiveness of our method on the LLaMA-7B model.

5.2 Ablation Study

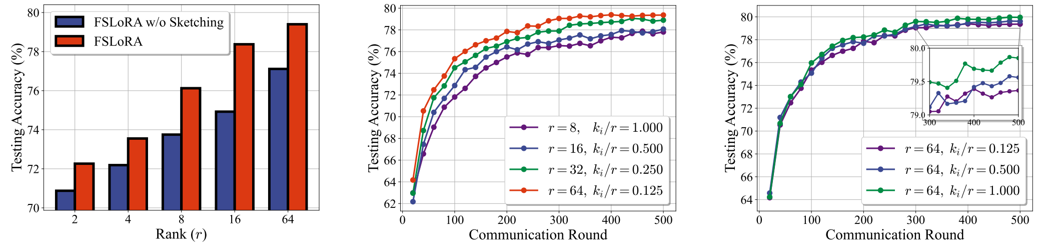
Impact of sketching: In Figures 2 and 3(a), we compare the performance of FSLoRA with and without sketching on fine-tuning the RoBERTa model and the LLaMA-3.2-3B model, respectively. Notably, FSLoRA without sketching is equivalent to the vanilla Federated LoRA. For FSLoRA with sketching, we apply a uniform sketching ratio of $k_i/r = 0.5$ across all distributed devices. The upload budget for each device is set to 100 and 200 times the size of the full global LoRA modules at the corresponding rank for the RoBERTa and the LLaMA-3.2-3B models, respectively. As shown in Figures 2 and 3(a), both FSLoRA with and without sketching achieve higher accuracy when the rank r increases due to the availability of more tunable parameters. In addition, FSLoRA consistently outperforms its non-sketched counterpart across all the ranks and datasets. The use of sketching increases the communication frequency for devices under the same communication budget, thereby facilitating the optimization process and enhancing fine-tuning efficiency.

Impact of the global rank: In Figure 3(b), we investigate the impact of the rank of the global LoRA modules on FSLoRA's performance. We vary the rank of the global LoRA modules while keeping the rank of submatrices updated by the devices to be consistent (i.e., $k_i = 8$). This ensures that the communication and computational resources on the client side remain unchanged. As illustrated in Figure 3(b), FSLoRA maintains stable convergence across all the configurations. Moreover, FSLoRA



(a) QNLI **(b) MRPC** **(c) COLA** **(d) MNLI** **(e) SST2**

Figure 2: Comparison between FSLoRA with and without sketching (i.e., vanilla Federated LoRA), where the upload budget for devices is set to 100× the size of the global LoRA modules at each rank. FSLoRA with sketching obtains a better performance, validating its communication efficiency.



(a) Comparison of FSLoRA with and without sketching, with an upload budget 200× the size of the global LoRA modules at each rank.

(b) Impact of the rank of global LoRA modules on FSLoRA, given a fixed rank k_i for the updated submatrices at the devices.

(c) Impact of the sketching ratio on FSLoRA's performance under a fixed rank $r = 64$ for the global LoRA modules.

Figure 3: Fine-tuning the LLaMA-3.2-3B model on the commonsense reasoning benchmark. The results are averaged over eight tasks, illustrating FSLoRA's ability to maintain strong performance while adapting to different rank and sketching configurations.

demonstrates improved performance as the global rank increases. This observation confirms that the proposed sketching mechanism enables resource-constrained systems to reap the benefits of a higher global rank, striking an effective balance between efficiency and performance.

Impact of sketching ratio: Finally, we investigate the impact of the sketching ratio on FSLoRA's performance by maintaining a constant global LoRA rank $r = 64$ while varying the sketching ratio k_i/r in the range $\{0.125, 0.5, 1\}$. As shown in Figure 3(c), there is a slight performance degradation as the sketching ratio decreases, which is consistent with our theoretical analysis. This reflects an inherent tradeoff: while a larger sketching ratio improves convergence and accuracy, a smaller ratio reduces both computational and communication overhead. Notably, the observed degradation remains minor, highlighting FSLoRA's ability to maintain strong performance even under constrained resources. This demonstrates its effectiveness in balancing efficiency and accuracy, making it well-suited for resource-limited scenarios.

Further experiments: Additional results, including detailed comparisons on each task in the commonsense reasoning benchmark corresponding to Figures 3(a) and 3(b), and the integration of compression and sketching, are provided in Appendix E. Results with more devices and broader heterogeneity, and a larger model can be found in Appendix C and Appendix D, respectively.

6 Conclusion and Limitation

We have proposed FSLoRA, a novel on-device collaborative LLM fine-tuning framework that introduces a sketching mechanism to enhance both performance and efficiency in resource-constrained systems. By maintaining large-rank LoRA modules on the server and allowing devices to selectively update submatrices based on the sketching ratios, FSLoRA effectively adapts to heterogeneous communication and computational constraints. We provide a rigorous convergence analysis of FSLoRA that characterizes how the sketching ratios affect the convergence rate. Finally, we confirmed the effectiveness of FSLoRA through extensive experiments across multiple datasets and models. A limitation of this work is the absence of a real-world deployment study that evaluates FSLoRA and baseline methods on the actual server and devices due to hardware constraints.

References

[1] J. Bai, D. Chen, B. Qian, L. Yao, and Y. Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources, 2024. URL `https://arxiv.org/abs/2402.11505`.

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

[3] Y. Byun and J. Lee. Towards federated low-rank adaptation of language models with rank heterogeneity. *arXiv preprint arXiv:2406.17477*, 2024.

[4] C. Chen, X. Feng, J. Zhou, J. Yin, and X. Zheng. Federated large language model: A position paper. *arXiv preprint arXiv:2307.08925*, 2023.

[5] Y. J. Cho, L. Liu, Z. Xu, A. Fahrezi, and G. Joshi. Heterogeneous lora for federated fine-tuning of on-device foundation models. *arXiv preprint arXiv:2401.06432*, 2024.

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

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

[8] Y. Demidovich, G. Malinovsky, E. Shulgin, and P. Richtárik. Mast: Model-agnostic sparsified training. *arXiv preprint arXiv:2311.16086*, 2023.

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

[10] D. Fan, B. Messmer, and M. Jaggi. On-device collaborative language modeling via a mixture of generalists and specialists. *arXiv preprint arXiv:2409.13931*, 2024.

[11] W. Fang, Z. Yu, Y. Jiang, Y. Shi, C. N. Jones, and Y. Zhou. Communication-efficient stochastic zeroth-order optimization for federated learning. *IEEE Transactions on Signal Processing*, 70: 5058–5073, 2022.

[12] W. Fang, D.-J. Han, E. Chen, S. Wang, and C. G. Brinton. Hierarchical federated learning with multi-timescale gradient correction. *arXiv preprint arXiv:2409.18448*, 2024.

[13] P. Guo, S. Zeng, Y. Wang, H. Fan, F. Wang, and L. Qu. Selective aggregation for low-rank adaptation in federated learning, 2025. URL `https://arxiv.org/abs/2410.01463`.

[14] A. Han, J. Li, W. Huang, M. Hong, A. Takeda, P. Jawanpuria, and B. Mishra. Sltrain: a sparse plus low-rank approach for parameter and memory efficient pretraining. *arXiv preprint arXiv:2406.02214*, 2024.

[15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[16] Z. Hu, L. Wang, Y. Lan, W. Xu, E.-P. Lim, L. Bing, X. Xu, S. Poria, and R. K.-W. Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*, 2023.

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

[18] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

[19] A. Khaled, K. Mishchenko, and P. Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 4519–4529. PMLR, 2020.

[20] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International conference on machine learning*, pages 5381–5393. PMLR, 2020.

[21] J. Koo, M. Jang, and J. Ok. Towards robust and efficient federated low-rank adaptation with heterogeneous clients, 2024. URL `https://arxiv.org/abs/2410.22815`.

[22] K. Kuo, A. Raje, K. Rajesh, and V. Smith. Federated lora with sparse communication, 2024.

[23] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[24] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[25] V. Lialin, S. Muckatira, N. Shivagunde, and A. Rumshisky. Relora: High-rank training through low-rank updates. In *The Twelfth International Conference on Learning Representations*, 2023.

[26] J. Liu, F. Shang, Y. Liu, H. Liu, Y. Li, and Y. Gong. Fedbcgd: Communication-efficient accelerated block coordinate gradient descent for federated learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 2955–2963, 2024.

[27] Y. Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019.

[28] G. Malinovsky, U. Michieli, H. A. A. K. Hammoud, T. Ceritli, H. Elesedy, M. Ozay, and P. Richtárik. Randomized asymmetric chain of lora: The first meaningful theoretical framework for low-rank adaptation. *arXiv preprint arXiv:2410.08305*, 2024.

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

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

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

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

[33] R. Shuttleworth, J. Andreas, A. Torralba, and P. Sharma. Lora vs full fine-tuning: An illusion of equivalence. *arXiv preprint arXiv:2410.21228*, 2024.

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

[35] A. Wang. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[36] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor. A novel framework for the analysis and design of heterogeneous federated learning. *IEEE Transactions on Signal Processing*, 69: 5234–5249, 2021.

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

[38] W. Xia, C. Qin, and E. Hazan. Chain of lora: Efficient fine-tuning of language models via residual learning. *arXiv preprint arXiv:2401.04151*, 2024.

[39] R. Ye, W. Wang, J. Chai, D. Li, Z. Li, Y. Xu, Y. Du, Y. Wang, and S. Chen. Openfedllm: Training large language models on decentralized private data via federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6137–6147, 2024.

[40] X. Yi, S. Zhang, T. Yang, and K. H. Johansson. Zeroth-order algorithms for stochastic distributed nonconvex optimization. *Automatica*, 142:110353, 2022.

[41] H. Yu, S. Yang, and S. Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5693–5700, 2019.

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

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

Appendix

A Comparison of Computation, Memory, and Communication

Computation and memory: Let P and q denote the memory cost of the full model and the global LoRA module (rank r), respectively. The computational cost is expressed with the big O notation. Forward and backward computations, as well as activation memory, are omitted as they are identical across all the considered methods. The results are summarized in Tables A.1 and A.2, where m and n denote the shape of the base model, k_i denotes the LoRA rank for device i, H denotes the number of iterations per round, and N is the number of devices. Additionally, the results for the vanilla Federated LoRA, denoted as FedLoRA, are reported under the case of homogeneous LoRA ranks, i,e., $k_i = r$.

Table A.1: Client-side computation load and memory usage comparison.

Method	Memory	Computation (per round)
FedLoRA	$P + q$	$\mathcal{O}(Hr(m+n))$
HeteroLoRA	$P + \frac{k_i}{r}q$	$\mathcal{O}(Hk_i(m+n))$
FlexLoRA	$P + \frac{k_i}{r}q$	$\mathcal{O}(Hk_i(m+n))$
FedStackLoRA	$P + \max\left\{\sum_{i=1}^{N} \frac{k_i}{r}q, P\right\}$	$\mathcal{O}\left(Hk_i(m+n)) + (\sum_{i=1}^{N} k_i)mn + mn\right)$
FSLoRA	$P{+}\frac{k_i}{r}q$	$\mathcal{O}(Hk_i(m+n))$

Table A.2: Server-side computation load and memory usage comparison.

Method	Memory	Computation (per round)
FedLoRA	Nq	$\mathcal{O}(N(m+n)r)$
HeteroLoRA	$\sum_{i=1}^{N} \frac{k_i}{r}q$	$\mathcal{O}(N(m+n)r)$
FlexLoRA	$\max\left\{\sum_{i=1}^{N} \frac{k_i}{r}q, 2P\right\}$	$\mathcal{O}\left((\sum_{i=1}^{N} k_i)mn + Nmn + \min\{m,n\}mn\right)$
FedStackLoRA	$\sum_{i=1}^{N} \frac{k_i}{r}q$	$\mathcal{O}\left((\sum_{i=1}^{N} k_i)(m+n)\right)$
FSLoRA	$\sum_{i=1}^{N} \frac{k_i}{r}q$	$\mathcal{O}(N(m+n)r)$

As shown in Tables A.1 and A.2, FSLoRA matches HetLoRA in both computation and memory cost. FedStackLoRA introduces additional client-side overhead due to merging LoRA modules. FlexLoRA incurs extra server-side costs from conducting SVD on the full model. In summary, FSLoRA guarantees convergence with minimum overhead.

Communication: We detailed the communication load for baselines and our methods in Table A.3, where q denotes the communication cost of a global LoRA module with rank r, k_i denotes the local LoRA rank for device i, m and n denote the shape of the base model, and N denotes the number of devices.

Table A.3: Communication complexity, assuming float 32 parameters and binary sketching indices.

	FedLoRA	HeteroLoRA	FlexLoRA	FedStackLoRA	FSLoRA
Uplink	q	$\frac{k_i}{r}q$	$\frac{k_i}{r}q$	$\frac{k_i}{r}q$	$\frac{k_i}{r}q$
Downlink	q	q	q	$\sum_{i=1}^{N} \frac{k_i}{r}q$	$q(1 + \frac{Nr}{32mn})$

For the uplink, all four heterogeneous LoRA algorithms incur the same communication overhead for transmitting updated local LoRA modules, which is lower than that of FedLoRA. For the downlink, FedStackLoRA requires broadcasting the stacked LoRA modules, while HeteroLoRA and FlexLoRA broadcast the updated global LoRA modules. FSLoRA, on the other hand, broadcasts both the global LoRA modules and additional sketching matrices. The extra communication introduced by the sketching matrices is negligible compared to that of the global LoRA modules, as it consists only of *binary sketching indices* (i.e., the diagonal elements of the sketching matrix). For instance, in the case of the LLaMA-3.2-3B model under our experimental LoRA configuration, the global LoRA modules contain 66,060,288 parameters, equivalent to approximately 252 MB when using float32. With a global rank of $r = 64$, the sketching indices require only 64 bits per device, covering all LoRA layers. Even with 100 devices, the total sketching overhead is merely 0.78 KB, which accounts for only 0.0003% of the global LoRA modules.

B Difference between FSLoRA and FedBCGD

Both FSLoRA and federated block coordinate gradient descent (FedBCGD) [26] are motivated by device heterogeneity but are designed for fundamentally different deployment contexts. FedBCGD partitions the full model $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{x}_s]$, assigning each block $\mathbf{x}_j$ to a subset of devices with similar resource constraints, while the shared block $\mathbf{x}_s$ is optimized across all devices. While this block-partitioning strategy is effective for smaller models, it relies on explicit and static allocation, which can limit scalability and flexibility. As such, FedBCGD and similar block coordinate methods based on the full model are less suitable for on-device LLM fine-tuning.

FSLoRA, in contrast, builds on LoRA and introduces sparse diagonal sketching. Given a sketch matrix $\mathbf{S}$, the gradients of the loss $\ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)$ with respect to the LoRA matrices $\mathbf{B}$ and $\mathbf{A}$ are sparse: only selected columns of $\mathbf{B}$ and rows of $\mathbf{A}$ are updated in each round. By configuring the rank and sparsity of the sketch matrix $\mathbf{S}$, FSLoRA flexibly controls both the computational and communication load per device, enabling adaptation to heterogeneous device capabilities.

The distinctions between FedBCGD and FSLoRA are summarized in Table B.1. To wrap up, these two algorithms are tailored for distinct purposes and deployment contexts.

Table B.1: Conceptual distinctions between FSLoRA and FedBCGD.

Aspect	FedBCGD	FSLoRA
Partition Type	Explicit & static	Random & sketching-based
Model Scope	Full model	LoRA modules
Adaptation Strategy	Assign different blocks	Adjust sketch rank (sparsity)

C Evaluation under Broader Heterogeneity and Increased Number of Devices

C.1 Increasing Device Heterogeneity and the Number of Devices

We extend our experiments on LLaMA-3.2-3B with the commonsense reasoning benchmark to 50 devices. We adopt Dirichlet-based partitioning for dataset splits. Specifically, the commonsense reasoning benchmark includes 8 tasks, and we partitioned them based on the Dirichlet distribution to construct task heterogeneity among 50 devices. The Dirichlet concentration parameter is set to $\alpha = 0.1$. We simulate device heterogeneity via different LoRA rank distributions (beyond the limited sketching ratio considered in Section 5). More capable devices are assigned higher ranks, reflecting varying compute capacities. We consider two different rank distributions: normal and heavy-tail distributions in the range $[4, 64]$.

Normal distribution: Ranks are sampled from a normal distribution with mean $\mu = \frac{a+b}{2}$ and standard deviation $\sigma = \frac{b-a}{6}$, where $a = 4$ and $b = 64$. This models a balanced distribution of device capabilities centered around the middle of the range.

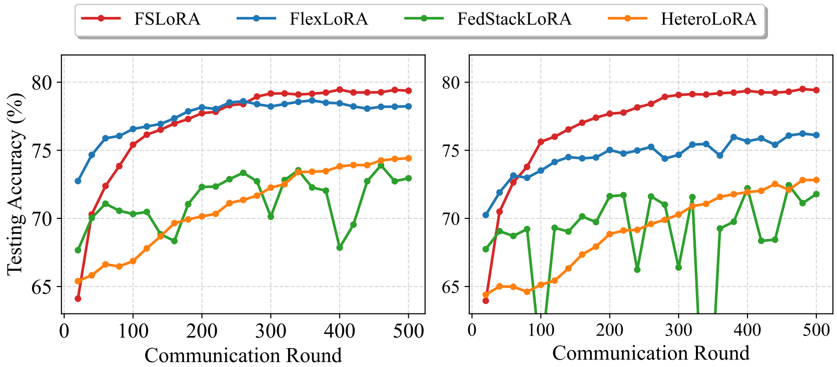
Heavy-tail distribution: We sample ranks using an inverse log-normal distribution. Specifically, we draw $x_i \sim \text{LogNormal}(\mu, \sigma)$ with $\mu = \log\left(\frac{a+b}{4}\right)$ and $\sigma = 1.0$, then set $k_i = 1/x_i$ and apply min-max normalization to scale values into the range $[a, b]$. This results in a heavy-tailed distribution where most devices receive low ranks, reflecting a scenario with many low-capability devices and a few high-capability ones.

Table C.1: Accuracy comparison under different device heterogeneity settings. FSLoRA outperforms baseline methods across both normal and heavy-tail LoRA rank distributions.

Rank setup	Method	ARC-c	ARC-e	BoolQ	HellaSwag	OBQA	PIQA	SIQA	WinoGrande	Avg.
Normal	HeteroLoRA	73.38	85.82	62.17	71.23	77.40	80.14	74.72	72.53	74.67
	FlexLoRA	74.23	87.84	68.37	79.77	76.00	82.97	75.90	78.13	77.90
	FedStackLoRA	68.17	83.75	64.93	75.67	71.40	77.20	71.24	70.09	72.81
	FSLoRA	75.77	86.95	69.67	81.53	80.60	84.06	76.20	78.85	79.20
Heavy-tail	HeteroLoRA	72.44	86.78	63.60	73.10	72.00	81.34	71.65	68.75	73.71
	FlexLoRA	73.04	86.70	62.23	75.57	78.00	81.12	74.77	73.32	75.59
	FedStackLoRA	67.92	81.90	64.90	72.77	74.00	80.41	75.28	70.24	73.43
	FSLoRA	75.77	86.70	69.67	81.40	80.40	83.90	76.15	78.77	79.10

As shown in Table C.1, FSLoRA outperforms other methods under both heterogeneity settings. As we move from normal to heavy-tail, where more devices are low-resource, overall performance decreases for all methods. However, FSLoRA exhibits the smallest drop, demonstrating stronger robustness to extreme device heterogeneity.

In Figure 4, we compare the convergence behavior of FSLoRA and three baseline methods under the aforementioned two types of device heterogeneity. Under the normal distribution, FlexLoRA exhibits fast initial progress but falls behind FSLoRA in final accuracy, likely due to approximation errors introduced by truncated SVD. This issue is exacerbated in the heavy-tail distribution, where low-rank devices dominate and SVD truncation causes greater distortion, further degrading FlexLoRA's performance. Similarly, HeteroLoRA's reliance on zero-padding reduces optimization efficiency, preventing it from achieving higher accuracy. FedStackLoRA fails to show steady improvement as communication progresses. One potential reason is that frequent model merging and random reinitialization of LoRA modules in each round disrupt the convergence continuity. In contrast, FSLoRA demonstrates robust and stable convergence across both scenarios, achieving the highest overall accuracy.



(a) Normal Distribution **(b) Heavy-Tail Distribution**

Figure 4: Convergence behavior of FSLoRA and baselines on the commonsense reasoning benchmark with the LLaMA-3.2-3B model. Notably, FSLoRA's per-round communication cost is at most equal to the baselines (as detailed in Appendix A). Testing accuracy is averaged over eight tasks.

C.2 Further Increasing the Number of Devices

We further evaluated the performance of FSLoRA by increasing the number of devices to 100. The results are presented in Table C.2. In this setting, local ranks follow a heavy-tailed distribution as described in the previous subsection, and all other experimental configurations remain unchanged. As shown in the table, FSLoRA maintains its advantage in terms of the average performance when scaling to more devices.

Table C.2: Accuracy comparison when the number of devices is $N = 100$. FSLoRA maintains its advantage in terms of the average accuracy.

Method	ARC-c	ARC-e	BoolQ	HellaSwag	OBQA	PIQA	SIQA	WinoGrande	Avg.
HeteroLoRA	71.76	86.24	62.57	68.07	76.60	79.38	74.10	69.69	73.55
FlexLoRA	73.38	87.54	69.03	75.27	78.60	80.47	74.16	73.80	76.53
FedStackLoRA	69.97	83.25	67.10	71.67	73.60	78.94	72.21	70.80	73.44
FSLoRA	74.40	87.54	70.13	79.90	79.40	83.57	76.51	78.93	78.80

C.3 Varying the Level of Data Heterogeneity

In Table C.3, we investigate the impact of the degree of data heterogeneity on performance. We increase the heterogeneity by decreasing the Dirichlet concentration parameter from $\alpha = 1$ to $\alpha = 0.1$. The local ranks follow the heavy-tail distribution described in the previous subsection, and all other experimental configurations remain consistent with Appendix C.1. As observed from

Table C.3, the performance of all methods degrades as heterogeneity increases. FSLoRA consistently achieves higher accuracy.

Table C.3: Accuracy comparison under different data heterogeneity settings. FSLoRA maintains its advantage over the baselines as the data heterogeneity increases. The number of devices is set to 50.

Data setup	Method	ARC-c	ARC-e	BoolQ	HellaSwag	OBQA	PIQA	SIQA	WinoGrande	Avg.
Dir(1)	HeteroLoRA	72.18	86.11	62.57	73.10	77.60	79.82	74.26	69.46	74.39
	FlexLoRA	74.06	87.25	65.67	74.90	78.80	81.01	74.16	74.27	76.27
	FedStackLoRA	70.14	83.29	67.27	71.60	73.60	78.73	72.16	70.96	73.47
	FSLoRA	75.85	87.50	70.93	81.47	81.00	82.86	76.66	78.53	79.35
Dir(0.1)	HeteroLoRA	72.44	86.78	63.60	73.10	72.00	81.34	71.65	68.75	73.71
	FlexLoRA	73.04	86.70	62.23	75.57	78.00	81.12	74.77	73.32	75.59
	FedStackLoRA	67.92	81.90	64.90	72.77	74.00	80.41	75.28	70.24	73.43
	FSLoRA	75.77	86.70	69.67	81.40	80.40	83.90	76.15	78.77	79.10

D Experiments on LLaMA-7B

Although our primary focus is on models suitable for on-device deployment, such as RoBERTa and LLaMA-3.2-3B models, we also include experiments on the larger LLaMA-7B model to demonstrate the scalability of FSLoRA in more complex models. Specifically, we fine-tune the LLaMA-7B model on the Wizard, Dolly-15k, and Alpaca datasets and evaluate it on 1444 MMLU samples (available at: https://github.com/ATP-1010/FederatedLLM). For Wizard and Dolly-15k, we adopt the same heterogeneous data partitioning as [37]. Since the Alpaca dataset lacks a clear task or domain structure, we apply a uniform random partitioning strategy to distribute the data across devices. We tune the q_proj and v_proj modules and set the local LoRA ranks $k_i = [64, 32, 16, 16, 8, 8, 4, 4, 4, 4]$ for 10 devices. The parameter settings are aligned with those in [37].

Table D.1: Performance comparison on LLaMA-7B model.

Method	Wizard	Dolly-15k	Alpaca	Avg
HeteroLoRA	27.15	26.70	28.74	27.53
FlexLoRA	28.25	35.60	30.40	31.42
FedStackLoRA	27.91	28.50	29.54	28.65
FSLoRA	30.33	40.79	30.68	33.93

As shown in Table D.1, FSLoRA achieves the highest average performance across all three datasets compared to baselines. These results demonstrate FSLoRA's potential for effective fine-tuning with the large-scale LLaMA-7B model under heterogeneous device settings.

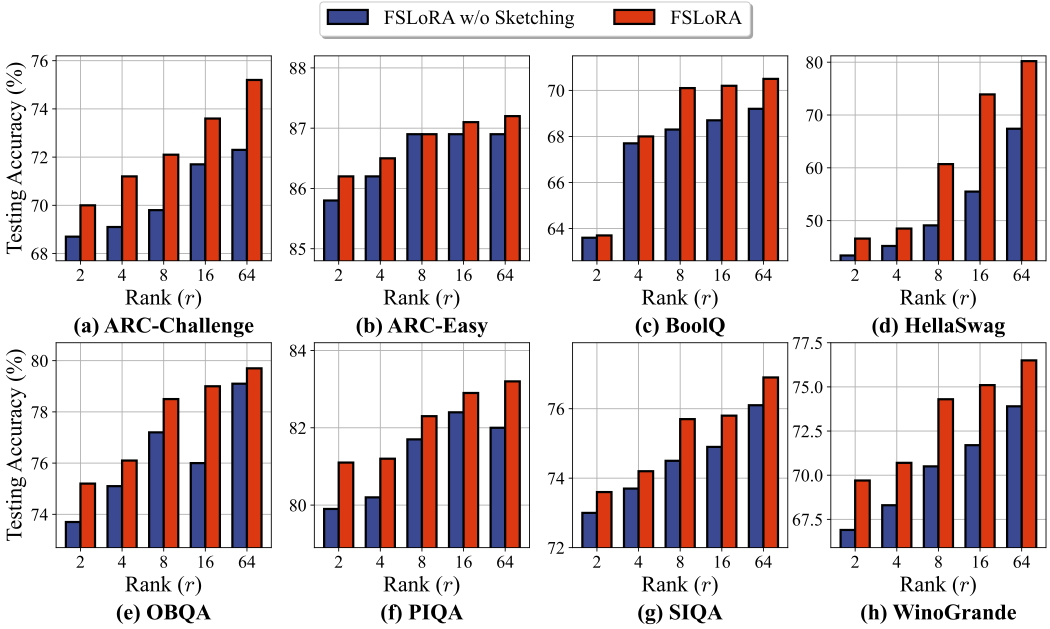
E Further Experiments

In this section, we provide additional results, including detailed per-task comparisons from the commonsense reasoning benchmark corresponding to Figures 3(a) and 3(b) and the investigation of the integration of communication compression and sketching.

E.1 Further Details on the Ablation Study

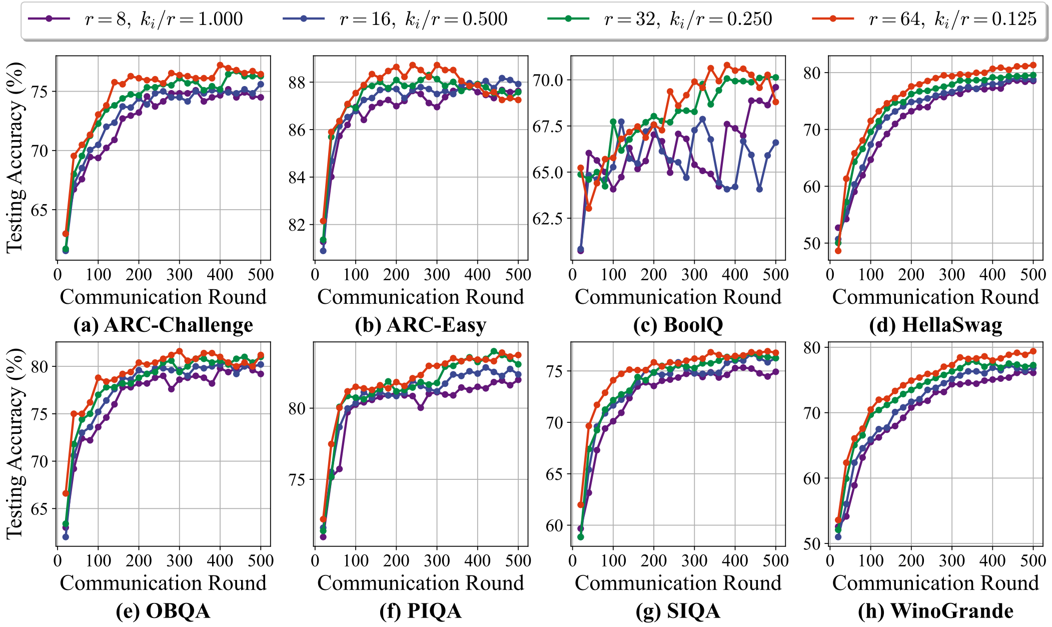
Impact of sketching: In Figure 5, we compare the performance of FSLoRA with and without sketching on eight tasks from the commonsense reasoning benchmark using the LLaMA-3.2-3B model. Notably, FSLoRA without sketching is equivalent to the vanilla Federated LoRA. For FSLoRA with sketching, we apply a uniform sketching ratio of $k_i/r = 0.5$ across all distributed devices. The uploading budget for each device is set to 200 times the size of the full global LoRA modules at the corresponding rank. It is clear that FSLoRA with sketching consistently outperforms its non-sketched counterpart across these eight tasks, demonstrating the effectiveness of sketching in improving performance.

Impact of the global rank: In Figure 6, we present the impact of the rank of the global LoRA modules on FSLoRA's performance across eight tasks from the commonsense reasoning benchmark. We



(a) ARC-Challenge (b) ARC-Easy (c) BoolQ (d) HealthSwag
(e) OBQA (f) PIQA (g) SIQA (h) WinoGrande

Figure 5: Comparison of FSLoRA with and without sketching, with an upload budget 200× the global LoRA module size at each rank. This is based on the commonsense reasoning benchmark and the LLaMA-3.2-3B model. We observe that the sketching mechanism improves performance across all considered tasks. The average accuracy of the eight tasks is shown in Figure 3(a).



(a) ARC-Challenge (b) ARC-Easy (c) BoolQ (d) HellaSwag
(e) OBQA (f) PIQA (g) SIQA (h) WinoGrande

Figure 6: Impact of the rank of global LoRA modules on FSLoRA, given a fixed rank for the updated submatrices at the devices. This is based on the commonsense reasoning benchmark and the LLaMA-3.2-3B model. Overall, FSLoRA demonstrates improved performance as the global rank increases. The average accuracy of the eight tasks is shown in Figure 3(b).

consider four configurations: 1) $r = 8,\ k_i/r = 1$, 2) $r = 16,\ k_i/r = 0.5$, 3) $r = 32,\ k_i/r = 0.25$, and 4) $r = 64,\ k_i/r = 0.125$. The rank of submatrices updated by the devices at each iteration remains consistent across all configurations (i.e., $k_i = 8$), ensuring that the communication and computational resources on the client side are kept fixed for all cases. In the ARC-Easy task, performance decreases as the rank increases to 64, potentially due to overfitting. In general, FSLoRA shows improved performance as the rank increases.

E.2 Integration of Sketching and Top-k Compression

Building on the commonsense reasoning benchmark and the LLaMA-3.2-3B model, we further explore the integration of two orthogonal techniques, sketching and top-k compression. Specifically, we employ top-k compression, a widely used compression approach introduced to LoRA by Kuo et al. [22], in FSLoRA to reduce communication overhead when uploading model updates. In our setup, the compression ratio is fixed at 0.5 for all methods, while the sketching ratio k_i/r varies over $\{0.125, 0.25, 0.5, 1\}$. Notably, FSLoRA with sketching ratio $k_i/r = 1$ corresponds to the vanilla Federated LoRA (i.e., without sketching). Figure 7 plots testing accuracy versus communication overhead, where the x-axis represents the amount of data uploaded per device (in MB), assuming parameters are stored in float 32 precision. The results clearly show that integrating sketching with top-k compression further improves communication efficiency: methods with lower sketching ratios consistently achieve higher accuracy under the same communication budget, highlighting the potential of FSLoRA for scalable and communication-efficient on-device LLM fine-tuning.

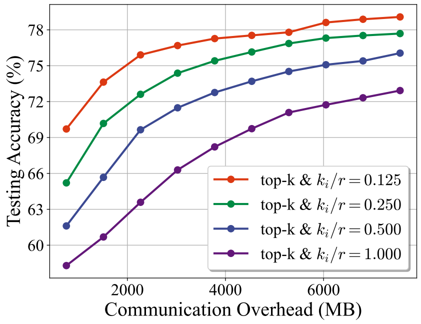


Figure 7: Testing accuracy versus communication overhead using float 32 precision. Lower sketching ratios achieve higher accuracy at the same communication cost, demonstrating that combining sketching with top-k compression leads to more communication-efficient training.

F Proof of the Theoretical Results

F.1 Preliminaries

Before presenting the proof of the main results, we first introduce some preliminary facts that will be used later.

Lemma F.1. *Suppose a sequence of independent random matrices $\{\mathbf{P}_i\}_{i=1}^N$ satisfy $\mathbb{E}[\mathbf{P}_i] = \mathbf{0}, \forall i$. Then,*

$$\mathbb{E}\left\|\frac{1}{N}\sum_{i=1}^N \mathbf{P}_i\right\|^2 = \frac{1}{N^2}\sum_{i=1}^N \mathbb{E}\left\|\mathbf{P}_i\right\|^2.$$

Lemma F.2. [36, Lemma 2] *Suppose a sequence of random matrices $\{\mathbf{P}_i\}_{i=1}^N$ satisfy $\mathbb{E}\left[\mathbf{P}_i \mid \mathbf{P}_{i-1}, \mathbf{P}_{i-2}, \ldots, \mathbf{P}_1\right] = \mathbf{0}, \forall i$. Then,*

$$\mathbb{E}\left[\left\|\sum_{i=1}^N \mathbf{P}_i\right\|^2\right] = \sum_{i=1}^N \mathbb{E}\left[\left\|\mathbf{P}_i\right\|^2\right].$$

Lemma F.3. [20, Lemma 17] *For any $a_0 \geq 0, b \geq 0, c \geq 0, d > 0$, there exist a constant $\eta \leq \frac{1}{d}$ such that*

$$\frac{a_0}{T\eta} + b\eta + c\eta^2 \leq 2\left(\frac{a_0 b}{T}\right)^{\frac{1}{2}} + 2c^{\frac{1}{3}}\left(\frac{a_0}{T}\right)^{\frac{2}{3}} + \frac{d a_0}{T}. \tag{8}$$

Lemma F.4 (Random sketching bounds)**.** *Let* $\mathbf{S}$ *be a random diagonal sketching matrix of the form*

$$\mathbf{S} \;=\; \frac{r}{k} \sum_{j \in \mathcal{I}} \mathbf{e}_j \, \mathbf{e}_j^\top,$$

where $\mathbf{e}_1, \ldots, \mathbf{e}_r \in \mathbb{R}^r$ *are standard unit basis vectors and* $\mathcal{I} \subseteq \{1, \ldots, r\}$ *is chosen uniformly at random with* $|\mathcal{I}| = k$*. Then any matrix* $\mathbf{X}$ *we have*

$$\|\mathbf{X}\,\mathbf{S}\|^2 \;\leq\; \frac{r^2}{k^2}\, \|\mathbf{X}\|^2, \tag{9}$$

and in expectation we have

$$\mathbb{E}_{\mathbf{S}}\Big[\|\mathbf{X}\,\mathbf{S}\|^2\Big] \;\leq\; \frac{r}{k}\, \|\mathbf{X}\|^2. \tag{10}$$

Proof. Since $\mathbf{S}$ is diagonal with exactly k diagonal entries equal to $\frac{r}{k}$ and the rest zero, its largest eigenvalue is $\frac{r}{k}$. Squaring gives

$$\mathbf{S}\,\mathbf{S}^\top \;=\; \mathbf{S}^2 \;\preceq\; \frac{r^2}{k^2}\,\mathbf{I},$$

Equivalently,

$$\mathbf{x}^\top \big(\mathbf{S}\,\mathbf{S}^\top\big)\mathbf{x} \;\leq\; \frac{r^2}{k^2}\, \|\mathbf{x}\|^2, \forall \mathbf{x}.$$

Setting $\mathbf{x} = \mathbf{x}_j$ to be the j-th column of $\mathbf{X}$ and summing over j implies

$$\|\mathbf{X}\,\mathbf{S}\|^2 \;=\; \sum_j \|\mathbf{S}^\top \mathbf{x}_j\|^2 \;=\; \sum_j \mathbf{x}_j^\top \,(\mathbf{S}\,\mathbf{S}^\top)\,\mathbf{x}_j \;\leq\; \frac{r^2}{k^2} \sum_j \|\mathbf{x}_j\|^2 \;=\; \frac{r^2}{k^2}\, \|\mathbf{X}\|^2,$$

which proves (9).

For the expected bound (10), note that each diagonal index $j \in \{1, \ldots, r\}$ is included in $\mathcal{I}$ with probability $\frac{k}{r}$. Hence the expectation of $\mathbf{S}^2$ satisfies

$$\mathbb{E}_{\mathbf{S}}\big[\mathbf{S}^2\big] \;=\; \frac{r^2}{k^2}\,\mathbb{E}\Big[\sum_{j \in \mathcal{I}} \mathbf{e}_j\,\mathbf{e}_j^\top\Big] \;=\; \frac{r^2}{k^2}\,\frac{k}{r}\,\mathbf{I} \;=\; \frac{r}{k}\,\mathbf{I}.$$

Thus for any vector $\mathbf{x}$,

$$\mathbb{E}_{\mathbf{S}}\Big[\|\mathbf{S}^\top \mathbf{x}\|^2\Big] \;=\; \mathbb{E}_{\mathbf{S}}\Big[\mathbf{x}^\top\,\mathbf{S}\,\mathbf{S}^\top\,\mathbf{x}\Big] \;=\; \mathbf{x}^\top\Big(\mathbb{E}[\mathbf{S}^2]\Big)\mathbf{x} \;=\; \frac{r}{k}\,\|\mathbf{x}\|^2.$$

Summing over columns of $\mathbf{X}$ again establishes

$$\mathbb{E}_{\mathbf{S}}\big[\|\mathbf{X}\,\mathbf{S}\|^2\big] \;=\; \sum_j \mathbb{E}_{\mathbf{S}}\big[\|\mathbf{S}^\top \mathbf{x}_j\|^2\big] \;=\; \sum_j \mathbf{x}_j^\top\Big(\mathbb{E}[\mathbf{S}^2]\Big)\mathbf{x}_j \;=\; \frac{r}{k}\,\|\mathbf{X}\|^2.$$

This completes the proof of Lemma F.4. □

F.2 Proof of Lemma 3.2

From the chain rule for matrix calculus, we know that:

$$\nabla_{\mathbf{Y}} g(\mathbf{XY}) = \mathbf{X}^\top \nabla g(\mathbf{XY}),\ \nabla_{\mathbf{X}} g(\mathbf{XY}) = \nabla g(\mathbf{XY})\mathbf{Y}^\top,$$

where $\nabla g(\mathbf{XY})$ denotes the gradient of g to $\mathbf{XY}$. Applying this to $\ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)$, we proceed as follows:
To compute the gradient with respect to $\mathbf{B}$, set $\mathbf{X} = \mathbf{B}$ and $\mathbf{Y} = \mathbf{SA}$:

$$\nabla_{\mathbf{B}} \ell(\mathbf{W}_0 + \mathbf{BSA}, \xi) = \nabla \ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)(\mathbf{SA})^\top.$$

Similarly, to compute the gradient with respect to $\mathbf{A}$, set $\mathbf{X} = \mathbf{BS}$ and $\mathbf{Y} = \mathbf{A}$:

$$\nabla_{\mathbf{A}} \ell(\mathbf{W}_0 + \mathbf{BSA}, \xi) = \mathbf{S}^\top \mathbf{B}^\top \nabla \ell(\mathbf{W}_0 + \mathbf{BSA}, \xi).$$

F.3 Proof of Theorem 4.4

The proof of Theorem 4.4 relies on the following proposition.

Proposition F.5. *Under Assumption 4.1,* $\tilde{f}_i(\mathbf{X};\mathbf{S}) = f_i(\mathbf{BS},\mathbf{A})$, $\mathbf{S} \in \mathcal{S}_i$, $f_i^{\mathcal{S}}(\mathbf{X}) = \mathbb{E}_{\mathbf{S}\sim\mathcal{S}_i}[\tilde{f}_i(\mathbf{X};\mathbf{S})]$, *and* $f^{\mathcal{S}}(\mathbf{X}) = \frac{1}{N}\sum_{i=1}^N f_i^{\mathcal{S}}(\mathbf{X})$ *are smooth with parameters* $L\frac{r^2}{k_i^2}$, $L\frac{r}{k_i}$, *and* $\left(\frac{1}{N}\sum_{i=1}^N \frac{r}{k_i}\right)L$*, respectively.*

The proof of Proposition F.5 is deferred to Appendix F.5. With this proposition, we are ready to prove Theorem 4.4.

In FSLoRA, the update direction in (4) corresponds to the negative stochastic gradient of $\ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)$ with respect to $[\mathbf{B};\mathbf{A}]$ for a given sketch $\mathbf{S}_i^t$. We have defined $\tilde{\ell}(\mathbf{X},\xi;\mathbf{S}) = \ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)$. The iterative equation for the proposed FSLoRA algorithm thus can be written as

$$\mathbf{X}^{t+1} = \mathbf{X}^t - \gamma\frac{1}{N}\sum_{i=1}^N\sum_{h=0}^{H-1}\nabla_{\mathbf{X}}\tilde{\ell}(\mathbf{X}_i^{t,h},\xi_i^{t,h};\mathbf{S}_i^t), \tag{11}$$

where $\mathbf{g}_i^{t,h}$ denotes the stochastic gradient $\nabla_{\mathbf{X}}\tilde{\ell}(\mathbf{X}_i^{t,h},\xi_i^{t,h};\mathbf{S}_i^t)$. Based on the smoothness of $f^{\mathcal{S}}(\mathbf{X})$, i.e., Proposition F.5, we have

$$\mathbb{E}[f^{\mathcal{S}}(\mathbf{X}^{t+1})] \le \mathbb{E}[f^{\mathcal{S}}(\mathbf{X}^t)] \underbrace{-\mathbb{E}\left\langle \nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t), \gamma\frac{1}{N}\sum_{i=1}^N\sum_{h=0}^{H-1}\mathbf{g}_i^{t,h}\right\rangle}_{T_1} + \frac{\gamma^2\bar{L}}{2}\underbrace{\mathbb{E}\left\|\frac{1}{N}\sum_{i=1}^N\sum_{h=0}^{H-1}\mathbf{g}_i^{t,h}\right\|^2}_{T_2}, \tag{12}$$

where $\bar{L} = \left(\frac{1}{N}\sum_{i=1}^N\frac{r}{k_i}\right)L$.

For T_1, we have

$$\begin{aligned}
T_1 =& - H\mathbb{E}\left\langle \nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t), \gamma\frac{1}{NH}\sum_{i=1}^N\sum_{h=0}^{H-1}\mathbf{g}_i^{t,h}\right\rangle \\
=& - H\mathbb{E}\left\langle \nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t), \gamma\frac{1}{NH}\sum_{i=1}^N\sum_{h=0}^{H-1}\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\rangle \\
=& - \frac{\gamma H}{2}\mathbb{E}\left\|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 - \frac{\gamma H}{2}\mathbb{E}\left\|\frac{1}{NH}\sum_{i=1}^N\sum_{h=0}^{H-1}\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\|^2 \\
&+ \frac{\gamma H}{2}\mathbb{E}\left\|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t) - \frac{1}{NH}\sum_{i=1}^N\sum_{h=0}^{H-1}\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\|^2 \\
\le& - \frac{\gamma H}{2}\mathbb{E}\left\|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 - \frac{\gamma H}{2}\mathbb{E}\left\|\frac{1}{N}\sum_{i=1}^N\sum_{h=0}^{H-1}\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\|^2 \\
&+ \frac{\gamma}{2}\sum_{h=0}^{H-1}\mathbb{E}\left\|\frac{1}{N}\sum_{i=1}^N\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}^t) - \frac{1}{N}\sum_{i=1}^N\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\|^2 \\
\le& - \frac{\gamma H}{2}\mathbb{E}\left\|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 - \frac{\gamma}{2H}\mathbb{E}\left\|\frac{1}{N}\sum_{i=1}^N\sum_{h=0}^{H-1}\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\|^2 \\
&+ \frac{\gamma H L^2}{2}\frac{1}{NH}\sum_{i=1}^N\frac{r^2}{k_i^2}\sum_{h=0}^{H-1}\mathbb{E}\left\|\mathbf{X}_i^{t,h} - \mathbf{X}^t\right\|^2,
\end{aligned} \tag{13}$$

where the last inequalities follow Jensen's inequality and Proposition F.5.

For T_2, we have

$$\begin{aligned}
T_2 =& \mathbb{E}\left\|\frac{1}{N}\sum_{i=1}^{N}\sum_{h=0}^{H-1}(\mathbf{g}_i^{t,h} \mp \nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h}))\right\|^2 \\
\leq& \frac{2}{N^2}\sum_{i=1}^{N}\mathbb{E}\left\|\sum_{h=0}^{H-1}(\mathbf{g}_i^{t,h} - \nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h}))\right\|^2 + 2\mathbb{E}\left\|\frac{1}{N}\sum_{i=1}^{N}\sum_{h=0}^{H-1}\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\|^2,
\end{aligned}$$

where the inequality follows the fact that $\mathbb{E}[\sum_{h=0}^{H-1}(\mathbf{g}_i^{t,h} - \nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h}))] = 0$ and the independence between devices.

Furthermore, we bound the first term on the right-hand side of the above inequality as

$$\mathbb{E}\left\|\sum_{h=0}^{H-1}(\mathbf{g}_i^{t,h} - \nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h}))\right\|^2 = \sum_{h=0}^{H-1}\mathbb{E}\left\|\mathbf{g}_i^{t,h} - \nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\|^2 \leq H\sigma^2 + \rho\sum_{h=0}^{H-1}\mathbb{E}\left\|\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\|^2,$$

where the equality follows Lemma F.2 and the inequality follows Assumption 4.2. For $\left\|\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\|^2$, utilizing Assumption 4.3 and Proposition F.5, we have

$$\begin{aligned}
&\left\|\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\|^2 = \left\|\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h}) \mp \nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}^t) \mp \nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 \\
\leq& 3\left\|\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h}) - \nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 + 3\left\|\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}^t) - \nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 + 3\left\|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 \\
\leq& 3\frac{r^2}{k_i^2}L^2\left\|\mathbf{X}_i^{t,h} - \mathbf{X}^t\right\|^2 + 3(c_h+1)\|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\|^2 + 3\rho\delta_h^2. \qquad (14)
\end{aligned}$$

Combining the above three inequalities gives rise to

$$\begin{aligned}
T_2 \leq& \frac{2H}{N}(\sigma^2 + 3\rho\delta_h^2) + 2\mathbb{E}\left\|\frac{1}{N}\sum_{i=1}^{N}\sum_{h=0}^{H-1}\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\|^2 + \frac{6\rho(c_h+1)H}{N}\mathbb{E}\|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\|^2 \\
&+ \frac{6\rho HL^2}{N}T_3, \qquad (15)
\end{aligned}$$

where $T_3 = \frac{1}{NH}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\sum_{h=0}^{H-1}\mathbb{E}\left\|\mathbf{X}_i^{t,h} - \mathbf{X}^t\right\|^2$. Combining (12), (13), and (15) yields

$$\begin{aligned}
\mathbb{E}[f^{\mathcal{S}}(\mathbf{X}^{t+1})] \leq& \mathbb{E}[f^{\mathcal{S}}(\mathbf{X}^t)] - (\frac{\gamma H}{2} - 3\gamma^2\rho(c_h+1)\frac{H}{N}\bar{L})\mathbb{E}\left\|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 + \gamma^2\bar{L}\frac{H}{N}(\sigma^2 + 3\rho\sigma_h^2) \\
&- (\frac{\gamma}{2H} - \gamma^2\bar{L})\mathbb{E}\left\|\frac{1}{N}\sum_{i=1}^{N}\sum_{h=0}^{H-1}\nabla_{\mathbf{X}} f_i^{\mathcal{S}}(\mathbf{X}_i^{t,h})\right\|^2 + (\frac{\gamma HL^2}{2} + 3\gamma^2\rho\bar{L}L^2\frac{H}{N})T_3,
\end{aligned}$$

where $\bar{L} = \left(\frac{1}{N}\sum_{i=1}^{N}\frac{r}{k_i}\right)L$. Let $\gamma \leq \min\{\frac{N}{24\rho(c_h+1)H\bar{L}}, \frac{1}{2H\bar{L}}, \frac{N}{6\rho\bar{L}}\}$, we have

$$\mathbb{E}[f^{\mathcal{S}}(\mathbf{X}^{t+1})] \leq \mathbb{E}[f^{\mathcal{S}}(\mathbf{X}^t)] - \frac{3\gamma H}{8}\mathbb{E}\left\|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 + \gamma^2\bar{L}\frac{H}{N}(\sigma^2 + 3\rho\sigma_h^2) + \frac{5\gamma}{8}HL^2T_3. \quad (16)$$

For T_3, we have

$$
\begin{aligned}
T_3 =& \frac{1}{NH}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\sum_{h=0}^{H-1}\mathbb{E}\left\|\gamma\sum_{\tau=0}^{h-1}\mathbf{g}_i^{t,\tau}\right\|^2 \\
=& \gamma^2\frac{1}{NH}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\sum_{h=0}^{H-1}\mathbb{E}\left\|\sum_{\tau=0}^{h-1}(\mathbf{g}_i^{t,\tau}\mp\nabla_{\mathbf{X}}f_i^{\mathcal{S}}(\mathbf{X}_i^{t,\tau}))\right\|^2 \\
\leq& 2\gamma^2\frac{1}{NH}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\sum_{h=0}^{H-1}\sum_{\tau=0}^{h-1}\mathbb{E}\left\|\mathbf{g}_i^{t,\tau}-\nabla_{\mathbf{X}}f_i^{\mathcal{S}}(\mathbf{X}_i^{t,\tau})\right\|^2+2\gamma^2\frac{1}{NH}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\sum_{h=0}^{H-1}h\sum_{\tau=0}^{h-1}\mathbb{E}\left\|\nabla_{\mathbf{X}}f_i^{\mathcal{S}}(\mathbf{X}_i^{t,\tau})\right\|^2 \\
\leq& 2\gamma^2H\sigma^2\left(\frac{1}{N}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\right)+\frac{2\rho\gamma^2}{NH}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\sum_{h=0}^{H-1}\sum_{\tau=0}^{h-1}\mathbb{E}\left\|\nabla_{\mathbf{X}}f_i^{\mathcal{S}}(\mathbf{X}_i^{t,\tau})\right\|^2 \\
&+\frac{2\gamma^2}{NH}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\sum_{h=0}^{H-1}h\sum_{\tau=0}^{h-1}\mathbb{E}\left\|\nabla_{\mathbf{X}}f_i^{\mathcal{S}}(\mathbf{X}_i^{t,\tau})\right\|^2 \\
\leq& 2\gamma^2H\sigma^2\left(\frac{1}{N}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\right)+\frac{2(\rho+1)\gamma^2H}{N}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\sum_{h=0}^{H-1}\mathbb{E}\left\|\nabla_{\mathbf{X}}f_i^{\mathcal{S}}(\mathbf{X}_i^{t,\tau})\right\|^2. \qquad (17)
\end{aligned}
$$

Plugging inequality (14) into inequality (17) yeilds

$$
\begin{aligned}
T_3 \leq& 2\gamma^2H\left(\frac{1}{N}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\right)\sigma^2+6(\rho+1)\gamma^2H^2\left(\frac{1}{N}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\right)\sigma_h^2 \\
&+6(\rho+1)\gamma^2L^2H^2T_3+6(\rho+1)\left(\frac{1}{N}\sum_{i=1}^{N}\frac{r^2}{k_i^2}\right)(c_h+1)\gamma^2H^2\mathbb{E}\left\|\nabla_{\mathbf{X}}f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2. \qquad (18)
\end{aligned}
$$

Denote $\kappa=\frac{1}{N}\sum_{i=1}^{N}\frac{r^2}{k_i^2}$, we simplify the above inequality as

$$(1-6(\rho+1)\gamma^2L^2H^2)T_3\leq 2\kappa\gamma^2H^2(\sigma_g^2+3(\rho+1)\sigma_h^2)+6\kappa(\rho+1)(c_h+1)\gamma^2H^2\mathbb{E}\left\|\nabla_{\mathbf{X}}f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2.$$

Let $\gamma\leq\frac{1}{\sqrt{12(\rho+1)}HL}$, we get the bound for T_3

$$T_3\leq 4\kappa\gamma^2H^2(\sigma^2+3(\rho+1)\sigma_h^2)+12\kappa(\rho+1)(c_h+1)\gamma^2H^2\mathbb{E}\left\|\nabla_{\mathbf{X}}f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2. \qquad (19)$$

Plugging the bound for T_3 into inequality (16) gives rise to

$$
\begin{aligned}
\mathbb{E}[f^{\mathcal{S}}(\mathbf{X}^{t+1})]\leq&\mathbb{E}[f^{\mathcal{S}}(\mathbf{X}^t)]-(\frac{3\gamma H}{8}-\frac{5\gamma H}{8}L^2\left(12\kappa(\rho+1)(c_h+1)\gamma^2H^2)\right)\mathbb{E}\left\|\nabla_{\mathbf{X}}f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 \\
&+\gamma^2\bar{L}\frac{H}{N}(\sigma^2+3\rho\sigma_h^2)+\frac{5\gamma}{8}HL^2\cdot 4\kappa\gamma^2H^2(\sigma^2+3(\rho+1)\sigma_h^2). \qquad (20)
\end{aligned}
$$

Let $\gamma\leq\frac{1}{8\sqrt{\kappa(\rho+1)(c_h+1)}HL}$, we obtain

$$\mathbb{E}[f^{\mathcal{S}}(\mathbf{X}^{t+1})]\leq\mathbb{E}[f^{\mathcal{S}}(\mathbf{X}^t)]-\frac{\gamma H}{4}\mathbb{E}\left\|\nabla_{\mathbf{X}}f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2+\gamma^2\bar{L}\frac{H}{N}\sigma_\rho^2+\frac{5}{2}\kappa\gamma^3H^3L^2\sigma_\rho^2, \qquad (21)$$

where $\sigma_\rho^2=\sigma^2+3(\rho+1)\sigma_h^2$.

Telescoping the above inequality from $t=0$ to $T-1$, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla_{\mathbf{X}}f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2\leq 4\frac{f^{\mathcal{S}}(\mathbf{X}^0)-f^*}{\gamma TH}+\gamma\frac{4\bar{L}}{N}\sigma_\rho^2+10\gamma^2H^2\widetilde{L}L\sigma_\rho^2, \qquad (22)$$

where f^* denotes the lower bound of $f^{\mathcal{S}}(\mathbf{X})$ and $\widetilde{L}=\kappa L$. This completes the proof of Theorem 4.4.

F.4 Proof of Corollary 4.5

Applying Lemma F.3 to the bound derived in Theorem 4.4 and letting $d = H$, it follows that there exists a learning rate $\gamma \leq \min\{\frac{N}{24\rho(c_h+1)H\bar{L}}, \frac{1}{8\sqrt{\widetilde{L}L(\rho+1)(c_h+1)H}}, \frac{1}{H}\}$ such that

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla_{\mathbf{X}} f^{\mathcal{S}}(\mathbf{X}^t)\right\|^2 \leq 8\frac{\sqrt{\bar{L}\mathcal{F}_0\sigma_\rho^2}}{\sqrt{NTH}} + 10(\widetilde{L}L)^{\frac{1}{3}}\left(\frac{\mathcal{F}_0\sigma_\rho}{T}\right)^{\frac{2}{3}} + \frac{4\mathcal{F}_0}{T}.$$

This completes the proof of Corollary 4.5.

F.5 Proof of Proposition F.5

i) For illustration, we need to recover $\mathbf{X}$ to $[\mathbf{B};\mathbf{A}]$ in this proof. According to the definition of $\tilde{f}_i(\mathbf{X};\mathbf{S})$ and $f_i(\mathbf{B},\mathbf{A})$, we have

$$\begin{aligned}\tilde{f}_i(\mathbf{X};\mathbf{S}) =& \tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S}) \qquad (23)\\ =& \mathbb{E}_{\xi\sim\mathcal{D}_i}\left[\ell(\mathbf{W}_0 + \mathbf{BSA}, \xi)\right]\\ =& f_i(\mathbf{BS},\mathbf{A}). \qquad (24)\end{aligned}$$

As $f_i(\mathbf{B},\mathbf{A})$ is L-smooth, we have

$$f_i(\mathbf{BS}+\Delta\mathbf{BS},\mathbf{A}+\Delta\mathbf{A}) \leq f_i(\mathbf{BS},\mathbf{A}) + \left\langle \begin{bmatrix}\nabla_{\mathbf{BS}} f_i(\mathbf{BS},\mathbf{A})\\ \nabla_{\mathbf{A}} f_i(\mathbf{BS},\mathbf{A})\end{bmatrix}, \begin{bmatrix}\Delta\mathbf{BS}\\ \Delta\mathbf{A}\end{bmatrix}\right\rangle + \frac{L}{2}\left\|\begin{bmatrix}\Delta\mathbf{BS}\\ \Delta\mathbf{A}\end{bmatrix}\right\|^2. \qquad (25)$$

According to (23) and (24), we have $\tilde{f}_i(\mathbf{B}+\Delta\mathbf{B},\mathbf{A}+\Delta\mathbf{A};\mathbf{S}) = f_i(\mathbf{BS}+\Delta\mathbf{BS},\mathbf{A}+\Delta\mathbf{A})$ and $\tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S}) = f_i(\mathbf{BS},\mathbf{A})$. Combining these with (25) gives rise to

$$\tilde{f}_i(\mathbf{B}+\Delta\mathbf{B},\mathbf{A}+\Delta\mathbf{A};\mathbf{S}) \leq \tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S}) + \left\langle \begin{bmatrix}\nabla_{\mathbf{BS}} f_i(\mathbf{BS},\mathbf{A})\\ \nabla_{\mathbf{A}} f_i(\mathbf{BS},\mathbf{A})\end{bmatrix}, \begin{bmatrix}\Delta\mathbf{BS}\\ \Delta\mathbf{A}\end{bmatrix}\right\rangle + \frac{L}{2}\left\|\begin{bmatrix}\Delta\mathbf{BS}\\ \Delta\mathbf{A}\end{bmatrix}\right\|^2. \qquad (26)$$

We denote

$$L(\mathbf{W}_0+\mathbf{BSA}) = \tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S}) = \mathbb{E}_{\xi\sim\mathcal{D}_i}\left[\ell(\mathbf{W}_0+\mathbf{BSA},\xi)\right]. \qquad (27)$$

Note that $\nabla_{\mathbf{BS}} f_i(\mathbf{BS},\mathbf{A}) = \nabla L(\mathbf{W}_0+\mathbf{BSA})\mathbf{A}^\top$ and $\nabla_{\mathbf{A}} f_i(\mathbf{BS},\mathbf{A}) = \mathbf{S}^\top\mathbf{B}^\top\nabla L(\mathbf{W}_0+\mathbf{BSA})$. We thus have

$$\begin{aligned}\left\langle \begin{bmatrix}\nabla_{\mathbf{BS}} f_i(\mathbf{BS};\mathbf{A})\\ \nabla_{\mathbf{A}} f_i(\mathbf{BS};\mathbf{A})\end{bmatrix}, \begin{bmatrix}\Delta\mathbf{BS}\\ \Delta\mathbf{A}\end{bmatrix}\right\rangle &= \left\langle \begin{bmatrix}\nabla L(\mathbf{W}_0+\mathbf{BSA})\mathbf{A}^\top\\ \mathbf{S}^\top\mathbf{B}^\top\nabla\mathbf{L}(\mathbf{W_0}+\mathbf{BSA})\end{bmatrix}, \begin{bmatrix}\Delta\mathbf{BS}\\ \Delta\mathbf{A}\end{bmatrix}\right\rangle\\ &= \left\langle \begin{bmatrix}\nabla L(\mathbf{W}_0+\mathbf{BSA})\mathbf{A}^\top\mathbf{S}^\top\\ \mathbf{S}^\top\mathbf{B}^\top\nabla\mathbf{L}(\mathbf{W_0}+\mathbf{BSA})\end{bmatrix}, \begin{bmatrix}\Delta\mathbf{B}\\ \Delta\mathbf{A}\end{bmatrix}\right\rangle\\ &= \left\langle \begin{bmatrix}\nabla_{\mathbf{B}}\tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S})\\ \nabla_{\mathbf{A}}\tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S})\end{bmatrix}, \begin{bmatrix}\Delta\mathbf{B}\\ \Delta\mathbf{A}\end{bmatrix}\right\rangle, \qquad (28)\end{aligned}$$

where the last equality follows the fact that $\tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S}) = L(\mathbf{W}_0+\mathbf{BSA})$ defined in (27) and

$$\begin{bmatrix}\nabla_{\mathbf{B}}\tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S})\\ \nabla_{\mathbf{A}}\tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S})\end{bmatrix} = \begin{bmatrix}\nabla L(\mathbf{W}_0+\mathbf{BSA})\mathbf{A}^\top\mathbf{S}^\top\\ \mathbf{S}^\top\mathbf{B}^\top\nabla L(\mathbf{W}_0+\mathbf{BSA})\end{bmatrix}.$$

Plugging (28) into (26) gives rise to

$$\tilde{f}_i(\mathbf{B}+\Delta\mathbf{B},\mathbf{A}+\Delta\mathbf{A};\mathbf{S}) \leq \tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S}) + \left\langle \begin{bmatrix}\nabla_{\mathbf{B}}\tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S})\\ \nabla_{\mathbf{A}}\tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S})\end{bmatrix}, \begin{bmatrix}\Delta\mathbf{B}\\ \Delta\mathbf{A}\end{bmatrix}\right\rangle + \frac{L}{2}\left\|\begin{bmatrix}\Delta\mathbf{BS}\\ \Delta\mathbf{A}\end{bmatrix}\right\|^2. \qquad (29)$$

In particular, $\left\|\begin{bmatrix}\Delta\mathbf{BS}\\ \Delta\mathbf{A}\end{bmatrix}\right\|^2 = \|\Delta\mathbf{BS}\|^2 + \|\Delta\mathbf{A}\|^2$. From (9), we know $\|\Delta\mathbf{BS}\|^2 \leq \frac{r^2}{k_i^2}\|\Delta\mathbf{B}\|^2$. Therefore, we have $\left\|\begin{bmatrix}\Delta\mathbf{BS}\\ \Delta\mathbf{A}\end{bmatrix}\right\|^2 = \frac{r^2}{k_i^2}\left\|\begin{bmatrix}\Delta\mathbf{B}\\ \Delta\mathbf{A}\end{bmatrix}\right\|^2$. As a result, $\tilde{f}_i(\mathbf{B},\mathbf{A};\mathbf{S})$ (i.e., $\tilde{f}_i(\mathbf{X},\mathbf{S})$) is $L\frac{r^2}{k_i^2}$-smooth.

ii) Note that $f_i^{\mathcal{S}}(\mathbf{X}) = \mathbb{E}_{\mathbf{S}\sim\mathcal{S}_i}[\tilde{f}_i(\mathbf{X}, \mathbf{S})]$. Therefore, we further take expectation for (29) over $\mathbf{S} \sim \mathcal{S}_i$, leading to

$$f_i^{\mathcal{S}}(\mathbf{B} + \Delta\mathbf{B}, \mathbf{A} + \Delta\mathbf{A}) \le f_i^{\mathcal{S}}(\mathbf{B}, \mathbf{A}) + \left\langle \begin{bmatrix} \nabla_{\mathbf{B}} f_i^{\mathcal{S}}(\mathbf{B}, \mathbf{A}) \\ \nabla_{\mathbf{A}} f_i^{\mathcal{S}}(\mathbf{B}, \mathbf{A}) \end{bmatrix}, \begin{bmatrix} \Delta\mathbf{B} \\ \Delta\mathbf{A} \end{bmatrix} \right\rangle + \frac{L}{2} \mathbb{E}_{\mathbf{S}\sim\mathcal{S}_i} \left\| \begin{bmatrix} \Delta\mathbf{B}\mathbf{S} \\ \Delta\mathbf{A} \end{bmatrix} \right\|^2 .$$

In particular, $\mathbb{E}_{\mathbf{S}\sim\mathcal{S}_i} \left\| \begin{bmatrix} \Delta\mathbf{B}\mathbf{S} \\ \Delta\mathbf{A} \end{bmatrix} \right\|^2 = \mathbb{E}_{\mathbf{S}\sim\mathcal{S}_i} \|\Delta\mathbf{B}\mathbf{S}\|^2 + \|\Delta\mathbf{A}\|^2$. From (10), we know $\mathbb{E}_{\mathbf{S}\sim\mathcal{S}_i} \|\Delta\mathbf{B}\mathbf{S}\|^2 \le \frac{r}{k_i} \|\Delta\mathbf{B}\|^2$. In other words, $\mathbb{E}_{\mathbf{S}\sim\mathcal{S}_i} \left\| \begin{bmatrix} \Delta\mathbf{B}\mathbf{S} \\ \Delta\mathbf{A} \end{bmatrix} \right\|^2 = \frac{r}{k_i} \left\| \begin{bmatrix} \Delta\mathbf{B} \\ \Delta\mathbf{A} \end{bmatrix} \right\|^2$. We thus claim that $f_i^{\mathcal{S}}(\mathbf{B}, \mathbf{A})$ (i.e., $f_i^{\mathcal{S}}(\mathbf{X})$) is $L\frac{r}{k_i}$-smooth.

iii) Finally, for $f^{\mathcal{S}}(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^{N} f_i^{\mathcal{S}}(\mathbf{X})$, we have

$$\nabla f^{\mathcal{S}}(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^{N} \nabla f_i^{\mathcal{S}}(\mathbf{X}).$$

Since $f_i^{\mathcal{S}}(\mathbf{X})$ is $L\frac{r}{k_i}$-smooth, we thus have

$$\|\nabla f_i^{\mathcal{S}}(\mathbf{X}) - \nabla f_i^{\mathcal{S}}(\mathbf{Y})\| \le L\frac{r}{k_i} \|\mathbf{X} - \mathbf{Y}\|, \quad \forall \mathbf{X}, \mathbf{Y}.$$

To find the Lipschitz constant of $f^{\mathcal{S}}(\mathbf{X})$, we analyze the difference between the gradients at two points $\mathbf{X}$ and $\mathbf{Y}$:

$$\begin{aligned} \|\nabla f^{\mathcal{S}}(\mathbf{X}) - \nabla f^{\mathcal{S}}(\mathbf{Y})\| &= \left\| \frac{1}{N} \sum_{i=1}^{N} \left(\nabla f_i^{\mathcal{S}}(\mathbf{X}) - \nabla f_i^{\mathcal{S}}(\mathbf{Y}) \right) \right\| \\ &\le \frac{1}{N} \sum_{i=1}^{N} \left\| \nabla f_i^{\mathcal{S}}(\mathbf{X}) - \nabla f_i^{\mathcal{S}}(\mathbf{Y}) \right\| \\ &\le \left(\frac{1}{N} \sum_{i=1}^{N} \frac{r}{k_i} L \right) \|\mathbf{X} - \mathbf{Y}\| . \end{aligned} \tag{30}$$

Therefore, $f^{\mathcal{S}}(\mathbf{X})$ is $\left(\frac{1}{N} \sum_{i=1}^{N} \frac{r}{k_i} L \right)$-smooth.

G Implementation Details

G.1 Details on Hyperparameters

Unless stated otherwise, the hyperparameters used in this work are as follows.

Table G.1: The hyperparameters for RoBERTa & GLUE and LLaMA-3.2-3B & commonsense reasoning benchmarks.

Hyperparameter	RoBERTa & GLUE	LLaMA-3.2-3B & commonsense reasoning
Dirichlet parameter	0.1	0.1
Batch size	16	16
LoRA dropout rate	0.1	0.1
Learning rate, γ	5e-4	3e-4
Communication round, T	200	500
Local iteration number, H	50	20
Target module	[“query”, “value”, “classification head”]	[“q_proj”, “k_proj”, “v_proj”, “up_proj”, “down_proj”]

G.2 Details on Datasets

G.2.1 GLUE Benchmark

GLUE is a widely recognized benchmark designed to assess the natural language understanding capabilities of language models [35].

- **CoLA** focuses on whether a given sentence is acceptable according to linguistic rules. It evaluates a model's ability to recognize well-formed sentences.
  - ▷ Input: A single sentence.
  - ☆ Output: A label indicating whether the sentence is acceptable or unacceptable.
- **SST-2** is designed for sentiment classification on movie reviews or short texts. It tests whether a model can correctly identify positive or negative sentiment in a given sentence.
  - ▷ Input: A single sentence.
  - ☆ Output: A label indicating positive or negative sentiment.
- **MRPC** checks if two sentences are paraphrases of each other, i.e., if they mean the same thing.
  - ▷ Input: Two sentences ('sentence1' and 'sentence2').
  - ☆ Output: A label indicating either equivalent or not equivalent.
- **QQP** tests a model's ability to determine if two questions ask the same thing.
  - ▷ Input: Two questions.
  - ☆ Output: A label indicating duplicate or not duplicate.
- **MNLI** tests whether a given hypothesis is entailed, contradicted, or neutral with respect to a premise.
  - ▷ Input: A premise (first sentence) and a hypothesis (second sentence).
  - ☆ Output: A label indicating entailment, contradiction, or neutral.
- **QNLI** aims to determine if a context sentence correctly answers a given question.
  - ▷ Input: A question and a sentence.
  - ☆ Output: A label indicating the sentence answers the question or it does not.
- **RTE** provides pairs of sentences to see if one implies the other.
  - ▷ Input: Two sentences ('sentence1' and 'sentence2')
  - ☆ Output: A label indicating whether the meaning of one sentence is entailed from the other one.

G.2.2 Commonsense Reasoning Benchmark

The training set of the commonsense reasoning benchmark is a mixture of multiple datasets including about 170K training samples from ARC-c/e [7], BoolQ [6], HellaSwag [42], OBQA [30], PIQA [2], SIQA [32], and WinoGrande [31] datasets.

- **ARC-c/e** contains the challenge and easy question set from the ARC dataset of genuine grade-school level, multiple-choice science questions.
- **BoolQ** is a question-answering dataset with yes/no questions derived from natural, real-world scenarios.
- **HellaSwag** includes questions for commonsense natural language inference, where a context and multiple endings are given, requiring the most coherent ending to be selected.
- **OBQA** involves multi-step problem-solving that combines commonsense knowledge, reasoning, and comprehension of accompanying textual information.
- **PIQA** focuses on questions requiring physical commonsense to solve. Each question offers two answer choices.
- **SIQA** targets reasoning about human actions and their social implication.
- **WinoGrande** is designed as a binary-choice fill-in-the-blank task, this dataset evaluates the ability to resolve ambiguous sentences through commonsense reasoning.

The input template, i.e., prompt format for these datasets is detailed in Table G.2.

Table G.2: The prompt template of the commonsense reasoning datasets [16].

Dataset	Input Template
ARC-c/e	Please choose the correct answer to the question: [QUESTION] Answer1: [ANSWER_1] Answer2: [ANSWER_2] Answer3: [ANSWER_3] Answer4: [ANSWER_4] Answer format: answer1/answer2/answer3/answer4 the correct answer is [ANSWER]
BoolQ	Please answer the following question with true or false, question: [QUESTION] Answer format: true/false the correct answer is [ANSWER]
HellaSwag	Please choose the correct ending to complete the given sentence: [ACTIVITY_LABEL]: [CONTEXT] Ending1: [ENDING_1] Ending2: [ENDING_2] Ending3: [ENDING_3] Ending4: [ENDING_4] Answer format: ending1/ending2/ending3/ending4 the correct answer is [ANSWER]
OBQA	Please choose the correct answer to the question: [QUESTION] Answer1: [ANSWER_1] Answer2: [ANSWER_2] Answer3: [ANSWER_3] Answer4: [ANSWER_4] Answer format: answer1/answer2/answer3/answer4 the correct answer is [ANSWER]
PIQA	Please choose the correct solution to the question: [QUESTION] Solution1: [SOLUTION_1] Solution2: [SOLUTION_2] Answer format: solution1/solution2 the correct answer is [ANSWER]
SIQA	Please choose the correct answer to the question: [QUESTION] Answer1: [ANSWER_1] Answer2: [ANSWER_2] Answer3: [ANSWER_3] Answer format: answer1/answer2/answer3 the correct answer is [ANSWER]
WinoGrande	Please choose the correct answer to fill in the blank to complete the given sentence: [SENTENCE] Option1: [OPTION_1] Option2: [OPTION_2] the correct answer is [ANSWER]