Decentralized Low-Rank Fine-Tuning of Large Language Models

Sajjad Ghiasvand, Mahnoosh Alizadeh, Ramtin Pedarsani
Electrical and Computer Engineering Department, UC Santa Barbara
{sajjad,alizadeh,ramtin}@ucsb.edu

Abstract

While parameter-efficient fine-tuning (PEFT) techniques like Low-Rank Adaptation (LoRA) offer computationally efficient adaptations of Large Language Models (LLMs), their practical deployment often assumes centralized data and training environments. However, real-world scenarios frequently involve distributed, privacy-sensitive datasets that require decentralized solutions. Federated learning (FL) addresses data privacy by coordinating model updates across clients without sharing raw data. While most federated fine-tuning methods adopt *centralized FL*, which relies on a parameter server for aggregating model updates—introducing potential bottlenecks and communication constraints—*decentralized FL* enables direct peer-to-peer communication among clients, bypassing the need for a server as an intermediary. Despite its advantages, decentralized fine-tuning for LLMs remains largely unexplored in the literature. To address this gap, we introduce `Dec-LoRA`, a decentralized fine-tuning algorithm based on LoRA. We conduct extensive experiments using BERT and LLaMA-2 models to benchmark `Dec-LoRA` against centralized LoRA and several other popular PEFT approaches in decentralized settings. Our results demonstrate that `Dec-LoRA` consistently achieves performance on par with centralized LoRA under various conditions, including data heterogeneity and quantization constraints. These findings highlight the potential of `Dec-LoRA` for scalable LLM fine-tuning in decentralized environments.

1 Introduction

The advent of Large Language Models (LLMs) such as GPT-4 (Achiam et al., 2023), LLaMA (Touvron et al., 2023), and BERT (Devlin et al., 2018) has revolutionized artificial intelligence by enabling remarkable capabilities in tasks such as translation and summarization (Bommasani et al., 2021), powered by sophisticated architectures like Transformers (Vaswani, 2017). These versatile models can be fine-tuned for domain-specific applications such as toxicity classification (Oskouie et al., 2025) using targeted datasets (Howard and Ruder, 2018), showcasing their adaptability across diverse fields. However, the sheer scale of these models, often comprising billions of parameters, makes complete fine-tuning computationally prohibitive and prone to overfitting. To address this, parameter-efficient fine-tuning (PEFT) techniques—such as Adapters (Houlsby et al., 2019), Prompt-Tuning (Lester et al., 2021), LoRA (Hu et al., 2021)—have emerged as practical solutions. These approaches selectively adjust only a fraction of the model parameters while keeping the rest static, significantly cutting computational demands without compromising performance (Ding et al., 2023). Among these, LoRA is preferred in certain applications and has been shown to have excellent efficiency, making it the focal point of our study.

Traditional PEFT methods often assume that LLMs are fine-tuned using data from a single machine or client. However, in real-world scenarios, sensitive data sets, such as medical records or legal documents, are frequently distributed across multiple devices (Manoel et al., 2023; Shoham and Rappoport, 2023; Soltanmohammadi and Hikmet, 2024). Privacy concerns make centralizing such data impractical, creating the urgent need for fine-tuning techniques capable of adapting LLMs at the edge while maintaining strict data privacy. In response to this challenge, Federated Learning (FL) (McMahan et al., 2017) emerges as a powerful solution by ensuring sensitive information remains on local devices throughout the training process. Instead of transferring raw data to a centralized server for training, FL enables clients to update model parameters locally and share only aggregated information, such as gradients or parameters (McMahan et al., 2017). Consequently,

Proceedings of the 1st Workshop for Research on Agent Language Models (REALM 2025), pages 334–345
July 31, 2025 ©2025 Association for Computational Linguistics

FL has been seamlessly integrated into PEFT approaches (Zhang et al., 2023; Fan et al., 2023; Zhao et al., 2023; Ghiasvand et al., 2024c), with federated fine-tuning of LoRA receiving particular attention for its ability to efficiently balance privacy, communication overhead, and model adaptability across different clients (Babakniya et al., 2023; Yan et al., 2024; Cho et al., 2023; Bai et al., 2024; Wang et al., 2024; Kuo et al., 2024; Sun et al., 2024; Chen et al.; Amini et al., 2025).

Almost all previous work on federated fine-tuning focuses on *centralized FL*, which relies on a centralized server to coordinate the aggregation of model updates. This dependency poses challenges, particularly in scenarios where communication resources are limited or where a centralized server introduces potential bottlenecks. Another FL architecture, called *decentralized FL*, enables direct peer-to-peer communication among clients, bypassing the need for a server as an intermediary (Yuan et al., 2024), while still preserving the key advantages of centralized FL. Recent advances have demonstrated the effectiveness of decentralization in LLM-based multi-agent systems, facilitating scalable and robust collaboration among distributed agents (Guo et al., 2024; Chen et al., 2024). Despite its broader applicability and critical role in emerging applications, decentralized fine-tuning for LLMs remains largely unexplored in the literature. In this work, we address this gap by proposing a decentralized fine-tuning algorithm and provide both empirical evidence and theoretical guarantees of its effectiveness.

Before delving into details, we summarize our contributions:

- We introduce `Dec-LoRA`, which, to the best of our knowledge, is the first FL algorithm designed to fine-tune LLMs in a decentralized setting.
- We benchmark `Dec-LoRA` against several popular PEFT approaches in decentralized settings and show that it consistently achieves superior accuracy and faster convergence on average across various tasks and settings.
- We conduct extensive experiments using BERT and LLaMA-2 family models, comparing centralized LoRA and `Dec-LoRA` under diverse settings, including data heterogeneity and quantization constraints. The results show that `Dec-LoRA` is an effective and practical solution for decentralized fine-tuning of LLMs.

2 Related Work

2.1 Parameter Efficient Fine-Tuning on LLMs

LLMs such as GPT-4 (Achiam et al., 2023), LLaMA (Touvron et al., 2023), and BERT (Devlin et al., 2018) have achieved remarkable performance across various tasks like translation and summarization (Bommasani et al., 2021) due to architectures like Transformers (Vaswani, 2017). However, these models typically contain billions of trainable parameters, making full fine-tuning (FFT) computationally expensive and inefficient, particularly for task-specific adaptations. To address this, PEFT methods have been introduced, enabling adaptation with significantly fewer trainable parameters while maintaining performance close to FFT. PEFT methods can be generally divided into three categories (Han et al., 2024). *Additive* introduces a small set of trainable parameters while keeping the original model frozen, as seen in Serial Adapter (Houlsby et al., 2019), Parallel Adapter (He et al., 2021), Prefix-Tuning (Li and Liang, 2021), and Prompt-Tuning (Lester et al., 2021). *Selective* PEFT fine-tunes only a subset of existing model parameters, with techniques like BitFit (Zaken et al., 2021) and PaFi (Liao et al., 2023). *Reparameterized* PEFT introduces a low-rank parameterization of pre-trained weights for training, with methods such as LoRA (Hu et al., 2021) and DoRA (Liu et al., 2024). *Among these, LoRA stands out for its efficiency, effectiveness, and adaptability, making it a compelling choice for fine-tuning LLMs. In this work, we specifically focus on the decentralization of LoRA.*

2.2 PEFT in Federated Setting

In their studies, (Zhang et al., 2023; Fan et al., 2023) evaluate and compare various PEFT methods, including Adapters, LoRA, Prompt Tuning, and BitFit in FL. Several adaptations of LoRA have been introduced to enhance its efficiency in highly heterogeneous federated settings. For instance, SLoRA (Babakniya et al., 2023; Yan et al., 2024) modifies the initialization process to better handle data heterogeneity, while HetLoRA (Cho et al., 2023) and FlexLoRA (Bai et al., 2024) dynamically adjust LoRA ranks per client to account for system heterogeneity. More recently, FLoRA (Wang et al., 2024) introduces slack matrices A and B for all clients and multiplies the resulting matrices to mitigate interference caused by the FedAvg algorithm. To reduce communication overhead in federated LoRA, (Kuo et al., 2024) propose sparse fine-tuning techniques. Meanwhile, FFA-LoRA (Sun et al., 2024) and RoLoRA (Chen et al.) aim to enhance model accuracy in heterogeneous environments while minimizing the number of trainable parameters. Additionally, FedTT (Ghiasvand et al., 2024c) integrates tensorized adapters for federated fine-tuning, significantly reducing trainable parameters and improving communication efficiency. *Although extensive research has explored PEFT methods, particularly LoRA in centralized FL, no study has examined their performance in a fully decentralized setting without a central server, despite its relevance to many real-world applications.*

2.3 Decentralized Optimization/Learning

The exploration of decentralized optimization techniques dates back to at least the 1980s (Tsitsiklis, 1984). These algorithms, often called *gossip algorithms* (Kempe et al., 2003; Boyd et al., 2006), are characterized by the absence of a central authority for spreading information. Instead, information propagates through the network, similar to how gossip spreads along the edges defined by the communication graph. Among the most commonly used methods in decentralized optimization are those based on (sub)gradient descent (Nedic and Ozdaglar, 2009; Johansson et al., 2010).

Decentralized optimization has recently facilitated the growth of decentralized learning, which has found applications in various domains, including autonomous vehicles (Chellapandi et al., 2023), healthcare systems (Warnat-Herresthal et al., 2021), industrial IoT environments (Qiu et al., 2022; Hexmoor and Maghsoudlou, 2024; Ghajari et al., 2025), and social networks (He et al., 2022). In particular, decentralization has demonstrated exceptional effectiveness in LLM-based multi-agent systems, enabling scalable and robust collaboration among distributed agents (Guo et al., 2024; Chen et al., 2024). *Although PEFT methods, such as LoRA, can be beneficial for decentralized FL of LLMs due to the large scale of these models, there is a lack of analysis on the use of such methods in decentralized scenarios. This paper aims to address this gap.*

3 Preliminaries

3.1 Low-Rank Adaptation: LoRA

Low-Rank Adaptation (LoRA) (Hu et al., 2021) is one of the most promising PEFT methods, enabling effective fine-tuning of large language models by freezing the entire model and adding low-rank trainable matrices in each layer. LoRA has been shown to outperform other PEFT methods, even in federated learning settings (Kuang et al., 2024).

In LoRA, for a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, the weight update is performed by a low-rank decomposition:

$$W_0 + \Delta W = W_0 + BA, \qquad (1)$$

where the training occurs on matrices $A \in \mathbb{R}^{r\times k}$ and $B \in \mathbb{R}^{d\times r}$, with $r \ll \min(d, k)$. Throughout the paper, we refer to r as the *rank* of LoRA, which is typically selected from $\{2, 4, 8, 16\}$.

Beyond good performance, the low number of trainable parameters makes LoRA a practical solution for decentralized fine-tuning of language models, where clients have limited training resources and communication between clients is costly.

3.2 Decentralized Fine-Tuning

We consider a connected network of n clients, denoted by $\mathcal{C} = \{c_1, \ldots, c_n\}$, with edges $\mathcal{E} \subseteq \mathcal{C} \times \mathcal{C}$ representing the communication links between clients. The network collaboratively aims to solve the following optimization problem:

$$\min_{A,B} \left[\ell(W_0 + BA) := \frac{1}{n} \sum_{i=1}^{n} \ell_i(W_0 + BA) \right],$$

where W_0 is the pre-trained model that is shared and fixed across all clients, and the local loss functions $\ell_i : \mathbb{R}^{d\times k} \to \mathbb{R}$ are distributed among n clients and are given in stochastic form:

$$\ell_i(W_0 + BA) = \mathbb{E}_{\xi_i \sim \mathcal{D}_i}[\mathcal{L}_i(W_0 + BA; \xi_i)].$$

Here, the expectation is taken with respect to a randomly selected sample set $\xi_i \sim \mathcal{D}_i$, where $\mathcal{D}_i$ denotes the local data distribution specific to client c_i. Standard empirical risk minimization is an important special case of this problem, when each $\mathcal{D}_i$ presents a finite number m_i of elements

$\{\xi_i^1, \ldots, \xi_i^{m_i}\}$. Then ℓ_i can be rewritten as

$$\ell_i(W_0 + BA) = \frac{1}{m_i} \sum_{j=1}^{m_i} \mathcal{L}_i \left(W_0 + BA; \xi_i^j \right).$$

In this decentralized setting, the clients communicate with each other along the edges $e \in \mathcal{E}$, which means that each client can communicate with its neighboring clients. Furthermore, each edge in the graph is associated with a positive mixing weight, and we denote the mixing matrix by $Q = [q_{ij}] \in \mathbb{R}^{n \times n}$. Additionally, we define:

$$W := W_0 + BA,$$
$$\tilde{\nabla} \mathcal{L}_i(W) := \nabla \mathcal{L}_i(W; \xi_i).$$

3.3 Mixing Matrix

As previously discussed, in our decentralized framework, clients communicate exclusively along the edges of a fixed communication graph that connects n nodes. Each edge in this graph is associated with a positive mixing weight. These weights are collectively represented by the mixing matrix $Q \in \mathbb{R}^{n \times n}$. We assume that the mixing matrix Q is symmetric and doubly stochastic, which is a common assumption in the letreture to ensure the consensus (Koloskova et al., 2020; Ghiasvand et al., 2024a). In this work, we utilize two widely used network topologies, which are described as follows:

- **Ring topology** consists of nodes arranged in a closed-loop structure, where each node communicates only with its immediate neighbors, leading to a sparse mixing matrix Q with nonzero entries corresponding to these direct connections. While this structured and deterministic communication pattern simplifies theoretical analysis, the limited communication range can slow down information diffusion, potentially hindering the overall convergence speed of the learning process. *We use this challenging topology in many parts of our experiment section.*

- **Erdős-Rényi topology** is a random graph model where each edge between nodes exists with an independent probability p_c, but the connectivity structure remains fixed throughout training. The mixing matrix for the Erdős-Rényi topology is defined as $Q = I - \frac{2}{3\lambda_{\max}(L)} L$, where L is the Laplacian matrix of an Erdős-Rényi graph with edge probability p_c. While a larger p_c results in a more connected network, facilitating faster information exchange, a smaller p_c leads to sparser connectivity, which may slow down convergence. *This relationship will be tested for LLMs in the experiment section.*

Algorithm 1 `Dec-LoRA`

```
for communication round t ← 1 to T do
    for clients c_i ∈ C in parallel do
        for local update k ← 1 to K do
            A_i^{(t)+k+1} = A_i^{(t)+k} − η ∇̃_A L_i ( W_i^{(t)+k} )
            B_i^{(t)+k+1} = B_i^{(t)+k} − η ∇̃_B L_i ( W_i^{(t)+k} )
        end for
        Client c_i sends A_i^{(t)+K} and B_i^{(t)+K} to their neighbors
    end for
    At Client c_i:  A_i^{(t+1)} = Σ_j q_ij A_j^{(t)+K}
                    B_i^{(t+1)} = Σ_j q_ij B_j^{(t)+K}
end for
```

4 Proposed Algorithm

We present the `Dec-LoRA` algorithm, described in detail in Alg. 1. At the start of the fine-tuning process, the full model's architecture and initial weights ($A^{(0)} \sim \mathcal{N}(0, \sigma^2)$, $B^{(0)} = \mathbf{0}$) are distributed to all clients in the set $\mathcal{C} = \{c_1, \cdots, c_n\}$.`Dec-LoRA` operates across T communication rounds, where each client performs K local updates on its trainable LoRA parameters in each round.

During a communication round t, each client $c_i \in \mathcal{C}$ initializes its local LoRA matrices with the obtained LoRA matrices from the previous round, $A_i^{(t)}$ and $B_i^{(t)}$, and then performs local training on its local dataset for K local updates:

$$A_i^{(t)+k+1} = A_i^{(t)+k} - \eta \tilde{\nabla}_A \mathcal{L}_i \left(W_i^{(t)+k} \right),$$
$$B_i^{(t)+k+1} = B_i^{(t)+k} - \eta \tilde{\nabla}_B \mathcal{L}_i \left(W_i^{(t)+k} \right),$$

where η is the learning rate, and $A_i^{(t)+k}$ and $B_i^{(t)+k}$ refer to the LoRA matrices for client c_i during communication round t and local update k.

Once the updates are complete, each client transmits its updated parameters, $A_i^{(t)+K}$ and $B_i^{(t)+K}$, to its neighboring clients. The clients then aggregate the parameters received from their neighbors using the mixing matrix Q. The updated

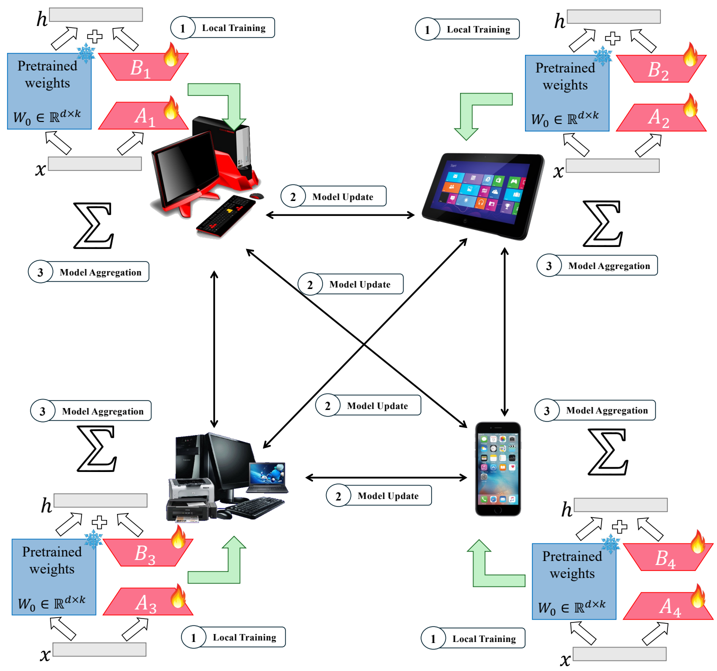

Figure 1: Illustration of the `Dec-LoRA` algorithm. The process includes three stages: (1) local training of low-rank matrices A and B on each client for K iterations using their private data, (2) communication of updated parameters between neighboring clients in the network, and (3) aggregation of received updates by each client using the mixing matrix Q to compute the next round's parameters.

Table 1: A comparative analysis of various decentralized PEFT methods using the RoBERTa-Base model. Highest accuracy is highlighted in **bold**, and the second highest is underlined.

	Method	# Param.	QNLI		SST2		MNLI		QQP		Avg.
			Ring	ER	Ring	ER	Ring	ER	Ring	ER	
$K=1$	`Dec-LoRA`	$0.60M$	**90.99**	**90.81**	**93.81**	93.92	85.37	**85.74**	**88.19**	**88.01**	**89.61**
	`Dec-Adapter`	$2.95M$	90.52	90.54	93.58	**94.38**	**85.45**	84.92	87.92	87.81	89.39
	`Dec-BitFit`	**$0.10M$**	86.47	85.81	92.43	92.78	83.76	84.24	82.39	83.46	86.42
	`Dec-IA3`	$0.65M$	89.36	89.05	92.66	92.55	83.61	83.61	85.86	85.53	87.78
$K=5$	`Dec-LoRA`	$0.60M$	**91.23**	**91.63**	**94.61**	94.27	**85.94**	**85.60**	**85.10**	86.76	**89.39**
	`Dec-Adapter`	$2.95M$	90.72	90.08	93.69	**94.38**	82.28	83.65	81.01	85.01	87.60
	`Dec-BitFit`	**$0.10M$**	88.28	89.42	93.35	93.12	82.28	82.50	80.59	85.35	86.86
	`Dec-IA3`	$0.65M$	90.12	89.97	93.00	93.23	84.89	84.50	84.02	**87.04**	88.35

parameters for client c_i are computed as:

$$A_i^{(t+1)} = \sum_j q_{ij} A_j^{(t)+K},$$

$$B_i^{(t+1)} = \sum_j q_{ij} B_j^{(t)+K}.$$

The steps of the `Dec-LoRA` algorithm are illustrated in Fig. 1.

5 Experiments

We conduct extensive experiments to evaluate the performance of the proposed algorithm across two language models. For the BERT-family models, we utilize RoBERTa-base (Liu et al., 2019), while for large-scale models, we employ LLaMA-2-7B (Touvron et al., 2023). To evaluate `Dec-LoRA`, we consider two topologies: a Ring topology, where each client connects to two neighbors, and an Erdős-Rényi topology. The mixing matrix for the Erdős-Rényi topology is defined as $Q = I - \frac{2}{3\lambda_{\max}(L)} L$, where L is the Laplacian matrix of an Erdős-Rényi graph with edge probability p_c. A larger p_c results in a more connected graph. We perform the experiments on NVIDIA A6000 and V100 GPUs.

Comparative methods. We compare our proposed `Dec-LoRA` method with three widely used PEFT approaches in a decentralized setting:

Table 2: A comparative analysis of centralized LoRA and `Dec-LoRA` with 10 and 20 clients under different ranks, using the RoBERTa-base model.

Rank	# Param.	QNLI			SST2			MNLI			QQP		
		LoRA	Dec-LoRA$_{10}$	Dec-LoRA$_{20}$	LoRA	Dec-LoRA$_{10}$	Dec-LoRA$_{20}$	LoRA	Dec-LoRA$_{10}$	Dec-LoRA$_{20}$	LoRA	Dec-LoRA$_{10}$	Dec-LoRA$_{20}$
2	0.07M	91.84	90.65	89.69	93.12	93.12	92.78	84.89	84.62	84.20	87.75	87.25	87.04
4	0.15M	92.49	90.85	90.32	93.58	93.81	94.15	85.54	85.54	84.50	88.28	87.89	87.15
8	0.30M	91.84	90.88	89.44	93.35	94.84	93.46	86.00	85.21	84.79	88.78	88.23	87.54
Avg.	0.17M	91.06	90.79	89.82	93.35	93.92	93.46	85.48	85.12	84.50	88.27	87.79	87.24

Table 3: Dataset descriptions and statistics.

Task	# Train	# Dev.	Metric
MRPC	3,301	367	F1 Score
SST-2	66,675	674	Accuracy
QNLI	103,695	5,463	Accuracy
QQP	360,210	40,430	Accuracy
MNLI	388,774	9,815	Accuracy

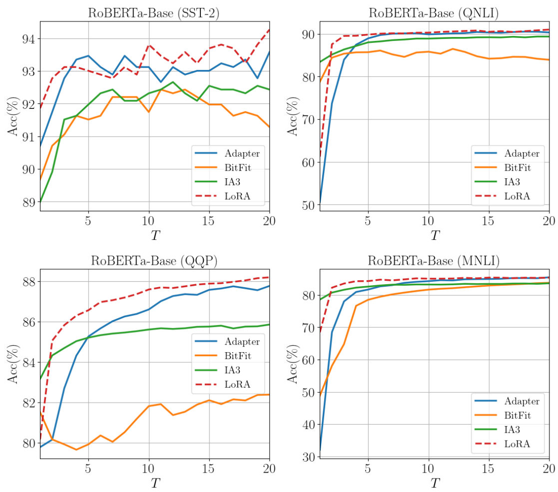

Figure 2: Convergence speed of decentralized PEFT methods using the Ring topology.

Adapter (Houlsby et al., 2019) (`Dec-Adapter`), BitFit (Zaken et al., 2021) (`Dec-BitFit`), and IA3 (Liu et al., 2022) (`Dec-IA3`). These methods are implemented using the Hugging Face PEFT library (Mangrulkar et al., 2022). Additionally, we maintain the default hyperparameter settings for the baseline methods to ensure consistency and generalizability across all tasks.

5.1 Performance on the BERT Family

We conduct experiments using the Generalized Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018), which comprises various natural language understanding tasks. These include sentiment analysis (SST2 (Socher et al., 2013)), similarity and paraphrasing tasks (MRPC, QQP (Dagan et al., 2005)), and natural language inference (MNLI, QNLI (Williams et al., 2017; Rajpurkar et al., 2018)). The evaluation metrics for the GLUE benchmark are detailed in Table 3. We utilize the full training dataset for each task and report the best validation accuracy. Validation accuracies are calculated based on the averaged models of the clients at the end of each communication round. A learning rate of $1e-3$ and a batch size of 32 are applied consistently across all tasks and methods.

5.1.1 Comparative Analysis of Decentralized PEFT methods

In this section, we compare the convergence speeds and accuracies of `Dec-LoRA` with three other methods discussed earlier. Table 1 presents the results after 20 iterations for experiments with $K = 1$, and 10 iterations for experiments with $K = 5$. We set the rank to 16 for `Dec-LoRA` and the bottleneck size to 64 for `Dec-Adapter`. The experiments are conducted using ring and ER topologies with 10 clients. As shown in the table, `Dec-LoRA` achieves the highest average accuracy among these methods, while maintaining a relatively low number of trainable parameters. Additionally, the convergence speed for the Ring topology with $K = 1$ is depicted in Fig. 2. As illustrated, `Dec-LoRA` demonstrates faster convergence compared to the other methods across various tasks.

5.1.2 Impact of Number of Clients, Edge Probabilities, and Number of Local Updates

To illustrate the impact of various parameters during the fine-tuning process, we present results for three methods on the QNLI and MNLI datasets in Fig. 3. As shown, `Dec-LoRA` outperforms the baselines across most settings. The detailed results are as follows.

1. **Fig. 3 (a) and (b):** These plots show the effect of the number of clients on accuracy for

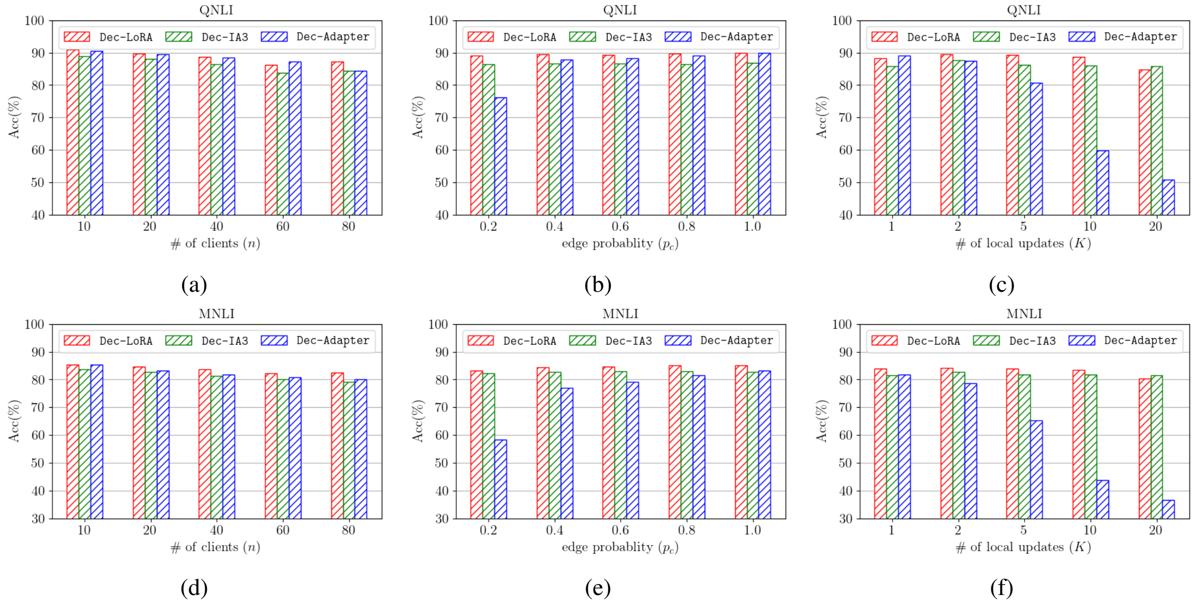

Figure 3: (a) and (d): Effect of the number of clients on accuracy for the Ring topology with $K = 1$. (b) and (e): Effect of edge probability in the Erdős-Rényi topology on accuracy for $K = 5$. (c) and (f): Effect of the number of local updates (K) on accuracy for the Ring topology.

Table 4: Left half: Performance analysis of `Dec-LoRA` with 4-bit quantization for 10 clients across different ranks. Right half: Performance analysis of `Dec-LoRA` under data heterogeneity with 3 clients across different ranks.

Method (Rank)	QNLI		SST2		MRPC		QQP		QNLI		SST2		MNLI		QQP	
	Full	4-bit	Full	4-bit	Full	4-bit	Full	4-bit	i.i.d.	non-i.i.d.	i.i.d.	non-i.i.d.	i.i.d.	non-i.i.d.	i.i.d.	non-i.i.d.
`Dec-LoRA` (2)	90.65	90.35	93.12	94.38	89.31	89.53	87.25	87.32	90.44	89.99	94.84	94.27	85.63	85.39	88.25	86.99
`Dec-LoRA` (4)	90.85	91.01	93.81	93.69	89.20	88.97	87.89	87.65	91.18	90.66	95.18	94.04	85.71	85.58	88.70	88.01
`Dec-LoRA` (8)	90.88	91.16	94.84	93.69	89.16	91.45	88.23	88.42	91.31	89.90	94.61	94.72	86.23	84.15	88.89	88.24
Avg.	90.79	90.84	93.92	93.92	89.22	89.98	87.79	87.80	90.98	90.18	94.88	94.34	85.86	85.04	88.61	87.75

the Ring topology with $T = 20$ and $K = 1$. As expected, the accuracy generally decreases as the number of clients increases across different tasks.

2. **Fig. 3 (c) and (d):** These plots highlight the influence of edge probability in the ER topology on accuracy, with parameters set to $N = 30$, $T = 5$, and $K = 5$. As demonstrated, a more connected network, characterized by a higher edge probability (p_c), leads to improved accuracy.

3. **Fig. 3 (e) and (f):** These figures show the effect of the number of local updates on accuracy for the Ring topology with $N = 30$. In these cases, $K \times T = 20$ for all experiments. While an increase in the number of local updates enhances communication efficiency, it results in lower accuracy when the total gradient computation remains constant.

5.1.3 Comparison of `Dec-LoRA` with Centralized LoRA

We provide a comparative analysis of centralized and decentralized LoRA with 10 and 20 clients across various ranks for the Ring topology, evaluated on four datasets, as presented in Fig. **??**. The results, obtained after 100 communication rounds, indicate that `Dec-LoRA` achieves accuracy levels comparable to centralized LoRA fine-tuning, highlighting its viability as an effective solution for decentralized settings.

5.1.4 `Dec-LoRA` with Quantization

In this section, we evaluate the use of LoRA with 4-bit quantization for the pretrained model (QLoRA) (Dettmers et al., 2024) in a decentralized setting. Specifically, QLoRA leverages 4-bit quantization to compress the base model, making it much more memory efficient, while still allowing for fine-tuning using trainable LoRA adapters. This technique is particularly suited

Table 5: A comparative analysis of centralized LoRA and Dec-LoRA with 10 clients under different ranks, using the LLaMA-2-7B model.

Rank	# Param.	Classfication				Multiple Choice				Generation			
		WIC		BoolQ		COPA		ReCoRD		SQuAD		DROP	
		LoRA	Dec-LoRA$_{10}$	LoRA	Dec-LoRA$_{10}$	LoRA	Dec-LoRA$_{10}$	LoRA	Dec-LoRA$_{10}$	LoRA	Dec-LoRA$_{10}$	LoRA	Dec-LoRA$_{10}$
2	$1.05M$	73.20	72.57	85.9	84.0	87	87	82.4	81.1	89.76	89.39	48.32	44.35
4	$2.10M$	74.61	72.26	85.2	83.5	85	87	81.1	81.2	89.79	89.79	46.56	44.97
8	$4.19M$	73.04	69.44	85.4	83.7	85	89	81.0	81.3	90.11	89.93	47.59	44.99
Avg.	$2.44M$	73.62	71.42	85.5	83.7	86	88	81.5	81.2	89.89	89.70	47.49	44.77

Table 6: A comparative analysis of centralized LoRA and Dec-LoRA with 10 clients, using the LLaMA2-13B and OPT-2.7B models.

Rank	LLaMA-2-13B						OPT-2.7B					
	COPA		ReCoRD		SQuAD		SQuAD		BoolQ		ReCoRD	
	LoRA	Dec-LoRA$_{10}$	LoRA	Dec-LoRA$_{10}$	LoRA	Dec-LoRA$_{10}$	LoRA	Dec-LoRA$_{10}$	LoRA	Dec-LoRA$_{10}$	LoRA	Dec-LoRA$_{10}$
8	92	93	84.2	83.9	92.24	90.88	81.93	79.50	63.1	63.6	77.0	75.8

for decentralized environments where computing resources are often limited, as it enables efficient training of large models on standard GPUs.

For our experiments, we consider a decentralized setup with 10 clients arranged in a Ring topology. The results, presented on the left side of Table 4, show that Dec-LoRA with 4-bit quantization of the pretrained model performs nearly identically to the regular Dec-LoRA. This demonstrates its potential to significantly reduce memory usage in decentralized settings.

5.1.5 Dec-LoRA under Data Heterogeneity

Data heterogeneity occurs when the training data is not identically and independently distributed across clients (non-i.i.d.), causing local models on individual clients to deviate from the global model's optimal state, which can result in slower convergence (Hsieh et al., 2020; Li et al., 2020).

In this section, we assess the performance of Dec-LoRA under the condition of data heterogeneity using three clients, following a setup similar to that in (Sun et al., 2024). For the heterogeneous setting, we partition the data based on class labels. For binary classification tasks, the data is split as $[0.15, 0.85]$, $[0.85, 0.15]$, and $[0.5, 0.5]$, while for three-class classification tasks, the splits are $[0.6, 0.2, 0.2]$, $[0.2, 0.6, 0.2]$, and $[0.2, 0.2, 0.6]$.

The results are presented on the right side of Table 4. As observed, there is a slight drop in performance under the non-i.i.d. setting. A more detailed discussion of this phenomenon can be found in Section 7.

5.2 Performance on the Large-Scale Language Models

Comparison with Other Methods. For large-scale language models, we conduct experiments only on centralized LoRA and Dec-LoRA. Applying Adapters for fine-tuning large-scale models still requires a significant number of trainable parameters. For instance, applying Adapters to LLaMA-2-13B with a bottleneck size of 64—the same as used for the BERT family—would require $50.33M$ trainable parameters, making it impractical for decentralized scenarios. As shown in Table 5, the number of trainable parameters remains relatively small when applying LoRA to LLaMA-2-7B. Additionally, since LLaMA-2-7B does not include bias terms, BitFit cannot be applied, as it updates only the bias parameters.

We evaluate performance using SuperGLUE tasks (Wang et al., 2019) and question-answering generation tasks, including SQuAD (Rajpurkar et al., 2016) and DROP (Dua et al., 2019). For each task, we randomly select 1000 samples for training and 1000 samples for validation, reporting the best validation accuracy. For the experiments involving large-scale language models, we use a learning rate of $1e-4$ and a batch size of 2 across all tasks and methods. All classification tasks within the SuperGLUE benchmark are restructured as

Table 7: The utilized metrics for the SuperGLUE benchmark and generation tasks.

Task Name	Metric
WIC	F1
BoolQ	Accuracy
COPA	Accuracy
ReCoRD	F1
SQuAD	F1
DROP	F1

language modeling tasks using the prompt-based fine-tuning approach outlined in (Malladi et al., 2023). The results shown in Table 5 are obtained after completing 10 communication rounds/epochs. The evaluation metrics are presented in Table 7. Table 5 presents the results for LoRA and `Dec-LoRA` implemented under a Ring topology with 10 clients and 3 local updates, utilizing the LLaMA-2-7B model. As shown, for larger models, `Dec-LoRA` performs comparably to centralized LoRA on most tasks, indicating its effectiveness in decentralized environments. Additional experiments conducted on LLaMA-2-13B and OPT-2.7B (Zhang et al., 2022) are presented in Table 6 under the same setting.

6 Conclusion

In this work, we introduce `Dec-LoRA`, a method for decentralized fine-tuning of LLMs using LoRA. By removing the need for a central server, Dec-LoRA allows efficient and scalable model adaptation in distributed settings while preserving data privacy. We compare `Dec-LoRA` with other popular PEFT methods in a decentralized setting and show that it outperforms them in both accuracy and convergence speed. Our extensive experiments on BERT and LLaMA-2 family models show that `Dec-LoRA` achieves performance comparable to centralized LoRA, even under challenging conditions such as data heterogeneity and quantization constraints. These findings highlight the potential of decentralized fine-tuning as a viable alternative to traditional federated approaches, opening new opportunities for future research in collaborative, serverless adaptation of LLMs.

7 Limitations

As shown in Section 5.1.5, the `Dec-LoRA` algorithm can experience performance degradation under data heterogeneity. This issue tends to become more pronounced as the number of clients and local updates increases. In the context of federated LLMs, methods such as (Babakniya et al., 2023; Yan et al., 2024) attempt to mitigate this challenge. Similarly, research like (Ghiasvand et al., 2024b; Ebrahimi et al., 2024; Ni et al., 2025) aims to address data heterogeneity in decentralized learning settings more generally. Investigating these existing approaches or developing new algorithms to tackle this issue remains a promising avenue for future research.

References

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

Hadi Amini, Md Jueal Mia, Yasaman Saadati, Ahmed Imteaj, Seyedsina Nabavirazavi, Urmish Thakker, Md Zarif Hossain, Awal Ahmed Fime, and SS Iyengar. 2025. Distributed llms and multimodal large language models: A survey on advances, challenges, and future directions. *arXiv preprint arXiv:2503.16585*.

Sara Babakniya, Ahmed Roushdy Elkordy, Yahya H Ezzeldin, Qingfeng Liu, Kee-Bong Song, Mostafa El-Khamy, and Salman Avestimehr. 2023. SLoRA: federated parameter efficient fine-tuning of language models. *arXiv preprint arXiv:2308.06522*.

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. 2024. Federated fine-tuning of large language models under heterogeneous language tasks and client resources. *arXiv preprint arXiv:2402.11505*.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. 2006. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530.

Vishnu Pandi Chellapandi, Liangqi Yuan, Christopher G Brinton, Stanislaw H Żak, and Ziran Wang. 2023. Federated learning for connected and automated vehicles: A survey of existing approaches and challenges. *IEEE Transactions on Intelligent Vehicles*.

Shuangyi Chen, Yue Ju, Hardik Dalal, Zhongwen Zhu, and Ashish J Khisti. Robust federated finetuning of foundation models via alternating minimization of LoRA. In *Workshop on Efficient Systems for Foundation Models II@ ICML2024*.

Yongchao Chen, Jacob Arkin, Yang Zhang, Nicholas Roy, and Chuchu Fan. 2024. Scalable multi-robot collaboration with large language models: Centralized or decentralized systems? In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4311–4317. IEEE.

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, Matt Barnes, and Gauri Joshi. 2023. Heterogeneous lora for federated fine-tuning of on-device foundation models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.

Mohammadjavad Ebrahimi, Uday V Shanbhag, and Farzad Yousefian. 2024. Distributed gradient tracking methods with guarantees for computing a solution to stochastic mpecs. In *2024 American Control Conference (ACC)*, pages 2182–2187. IEEE.

Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. 2023. Fatellm: A industrial grade federated learning framework for large language models. *arXiv preprint arXiv:2310.10049*.

Ghazal Ghajari, Ashutosh Ghimire, Elaheh Ghajari, and Fathi Amsaad. 2025. Network anomaly detection for iot using hyperdimensional computing on nsl-kdd. *arXiv preprint arXiv:2503.03031*.

Sajjad Ghiasvand, Amirhossein Reisizadeh, Mahnoosh Alizadeh, and Ramtin Pedarsani. 2024a. Communication-efficient and decentralized federated minimax optimization. In *2024 60th Annual Allerton Conference on Communication, Control, and Computing*, pages 01–07. IEEE.

Sajjad Ghiasvand, Amirhossein Reisizadeh, Mahnoosh Alizadeh, and Ramtin Pedarsani. 2024b. Robust decentralized learning with local updates and gradient tracking. *arXiv preprint arXiv:2405.00965*.

Sajjad Ghiasvand, Yifan Yang, Zhiyu Xue, Mahnoosh Alizadeh, Zheng Zhang, and Ramtin Pedarsani. 2024c. Communication-efficient and tensorized federated fine-tuning of large language models. *arXiv preprint arXiv:2410.13097*.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Chaoyang He, Emir Ceyani, Keshav Balasubramanian, Murali Annavaram, and Salman Avestimehr. 2022. Spreadgnn: Decentralized multi-task federated learning for graph neural networks on molecular data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 6865–6873.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.

Henry Hexmoor and Ebrahim Maghsoudlou. 2024. Iot with blockchain: A new infrastructure proposal. *Proceedings of 39th International Confer*, 98:15–24.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. 2020. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Björn Johansson, Maben Rabi, and Mikael Johansson. 2010. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3):1157–1170.

David Kempe, Alin Dobra, and Johannes Gehrke. 2003. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 482–491. IEEE.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. 2020. A unified theory of decentralized sgd with changing topology and local updates. In *International conference on machine learning*, pages 5381–5393. PMLR.

Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. 2024. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5260–5271.

Kevin Kuo, Arian Raje, Kousik Rajesh, and Virginia Smith. 2024. Federated lora with sparse communication. *arXiv preprint arXiv:2406.05233*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Baohao Liao, Yan Meng, and Christof Monz. 2023. Parameter-efficient fine-tuning without introducing new latency. *arXiv preprint arXiv:2305.16742*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. 2023. Fine-tuning language models with just forward passes. *arXiv preprint arXiv:2305.17333*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Andrea Manoel, Mirian del Carmen Hipolito Garcia, Tal Baumel, Shize Su, Jialei Chen, Robert Sim, Dan Miller, Danny Karmon, and Dimitrios Dimitriadis. 2023. Federated multilingual models for medical transcript analysis. In *Conference on Health, Inference, and Learning*, pages 147–162. PMLR.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Angelia Nedic and Asuman Ozdaglar. 2009. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61.

Zhou Ni, Masoud Ghazikor, and Morteza Hashemi. 2025. pfedwn: A personalized federated learning framework for d2d wireless networks with heterogeneous data. *arXiv preprint arXiv:2501.09822*.

Haniyeh Ehsani Oskouie, Christina Chance, Claire Huang, Margaret Capetz, Elizabeth Eyeson, and Majid Sarrafzadeh. 2025. Leveraging large language models and topic modeling for toxicity classification. *Workshop on Computing, Networking and Communications (CNC)*, pages 123–127.

Wenqi Qiu, Wu Ai, Huazhou Chen, Quanxi Feng, and Guoqiang Tang. 2022. Decentralized federated learning for industrial iot with deep echo state networks. *IEEE Transactions on Industrial Informatics*, 19(4):5849–5857.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ofir Ben Shoham and Nadav Rappoport. 2023. Federated learning of medical concepts embedding using behrt. *arXiv preprint arXiv:2305.13052*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Ehsan Soltanmohammadi and Neset Hikmet. 2024. Optimizing healthcare big data processing with containerized pyspark and parallel computing: A study on etl pipeline efficiency. *Journal of Data Analysis and Information Processing*, 12(4):544–565.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. 2024. Improving LoRA in privacy-preserving federated learning. *arXiv preprint arXiv:2403.12313*.

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

John N Tsitsiklis. 1984. *Problems in decentralized decision making and computation*. Ph.D. thesis, Massachusetts Institute of Technology.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. 2024. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *arXiv preprint arXiv:2409.05976*.

Stefanie Warnat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian Händler, Peter Pickkers, N Ahmad Aziz, et al. 2021. Swarm learning for decentralized and confidential clinical machine learning. *Nature*, 594(7862):265–270.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Yuxuan Yan, Shunpu Tang, Zhiguo Shi, and Qianqian Yang. 2024. FeDeRA: efficient fine-tuning of language models in federated learning leveraging weight decomposition. *arXiv preprint arXiv:2404.18848*.

Liangqi Yuan, Ziran Wang, Lichao Sun, S Yu Philip, and Christopher G Brinton. 2024. Decentralized federated learning: A survey and perspective. *IEEE Internet of Things Journal*.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhuo Zhang, Yuanhang Yang, Yong Dai, Qifan Wang, Yue Yu, Lizhen Qu, and Zenglin Xu. 2023. Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In *Annual Meeting of the Association of Computational Linguistics 2023*, pages 9963–9977. Association for Computational Linguistics (ACL).

Haodong Zhao, Wei Du, Fangqi Li, Peixuan Li, and Gongshen Liu. 2023. Fedprompt: Communication-efficient and privacy-preserving prompt tuning in federated learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.