Efficient and Differentially Private Federated LLM Fine-Tuning on Heterogeneous Clients

Nan Yan[†]
Wuhan University
Wuhan, China
nanyan@whu.edu.cn

Yuqing Li[*†]
Wuhan University
Wuhan, China
li.yuqing@whu.edu.cn

Xiong Wang
Huazhong University of Science and Technology
Wuhan, China
xiongwang@hust.edu.cn

Jing Chen[†]
Wuhan University
Wuhan, China
chenjing@whu.edu.cn

Wei Wang
The Hong Kong University of Science and Technology
Hong Kong, China
weiwa@cse.ust.hk

Kun He[†]
Wuhan University
Wuhan, China
hekun@whu.edu.cn

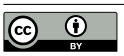
Ruiying Du[†]
Wuhan University
Wuhan, China
duraying@whu.edu.cn

Shuhua Li[†]
Wuhan University
Wuhan, China
shuhuali@whu.edu.cn

Abstract

Federated low-rank adaptation (FedLoRA) allows multiple clients to collaboratively fine-tune large language models (LLMs) on downstream tasks without exposing their private data. To mitigate privacy leakage during aggregation, differential privacy (DP) is widely used to clip and perturb local model updates with noise, yet it can compromise model accuracy due to the inherent privacy-utility trade-off. The performance degradation becomes worse under the FedLoRA setting with the *amplified DP noise impact* and *client heterogeneity* in both model structure and data distribution. In this work, we propose iP-FedLoRA, a privacy-preserving federated fine-tuning framework for heterogeneous clients that strikes a good privacy-utility balance. Specifically, to fully utilize clients' heterogeneous resources, we customize LoRA modules based on their available resources. iP-FedLoRA employs *matrix-wise* differentially private local fine-tuning with *sensitivity-aware* noise allocation and *rank-compensated* LoRA regularization, which effectively alleviates noise impact of low-rank modules and enhances training efficiency. By leveraging noise-resilient knowledge distillation, iP-FedLoRA facilitates heterogeneous LoRA aggregation that selectively prioritizes high-confidence knowledge to filter DP-induced noise, thereby achieving robust knowledge transfer. Through rigorous privacy analysis and extensive experiments, we show that iP-FedLoRA provides privacy guarantees, improves model accuracy by up to 3.8%, and expedites training by 1.37 − 2.23×.

[*]Corresponding author: Yuqing Li.
[†]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University



KDD '26, Jeju Island, Republic of Korea

ACM ISBN 979-8-4007-2259-2/2026/08
https://doi.org/10.1145/3770855.3817914

CCS Concepts

• **Computing methodologies** → **Distributed algorithms**; • **Security and privacy** → **Privacy protections**.

Keywords

Federated LLM Fine-Tuning; Differential Privacy; Data and Model Heterogeneity

ACM Reference Format:
Nan Yan, Yuqing Li, Xiong Wang, Jing Chen, Wei Wang, Kun He, Ruiying Du, and Shuhua Li. 2026. Efficient and Differentially Private Federated LLM Fine-Tuning on Heterogeneous Clients. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3770855.3817914

Resource Availability:
The source code of this paper has been made publicly available at https://doi.org/10.5281/zenodo.20433736.

1 Introduction

Large language models (LLMs), such as GPT-4 [3] and Llama-3 [12], have revolutionized the field of natural language processing (NLP), demonstrating impressive capabilities in a wide array of tasks [50, 51]. LLMs are initially *pre-trained* on vast public datasets, and then *fine-tuned* towards various downstream tasks using domain-specific private data. As LLMs scale up to hundreds of billions of parameters, fine-tuning the full model parameters becomes prohibitively expensive. *Low-rank adaptation* (LoRA) [21] is thus developed as a popular *parameter-efficient* fine-tuning (PEFT) approach that limits fine-tuning to a small portion of parameters. More specifically, with LoRA, the model update matrix is represented as the product of two trainable low-rank matrices. Despite its effectiveness, LoRA often requires collecting large quantities of individual data to fine-tune LLMs, which may contain sensitive information such as personal

images, financial information or medical records, inevitably leading to a severe *privacy leakage* [49]. To break this barrier, practitioners have turned to *federated learning* (FL) [29, 38] as a means to fine-tune LLMs with LoRA, known as FedLoRA, which allows clients to collaboratively fine-tune a shared LLM for a common task without collecting their private data [4]. Clients *freeze* the base model parameters and only perform updates and aggregations on *low-rank* matrices under the coordination of a central server [53].

Although clients' raw training data are not explicitly exposed during federated fine-tuning, adversaries can still exploit the exchanged LoRA parameters to infer sensitive information or even reconstruct the training data samples [15, 32, 40]. *Differential privacy* (DP) [1, 11] offers a promising solution to address such privacy concerns. Specifically, each client can *clip* and *perturb* the sensitive information (e.g., LoRA parameters) with *random noise* before sending them to the (untrusted) server, which provides theoretical guarantees against the identification of private data. Unfortunately, this privacy guarantee comes at the cost of *model accuracy* due to noise perturbation. Even worse, the problem is more salient in FedLoRA: since DP noise is injected directly into the LoRA parameters, it perturbs the trainable adapters, potentially hindering their ability to effectively adapt the base model to target datasets.

The primary challenge in differentially private FedLoRA systems is how to strike an improved *privacy-utility trade-off*. LoRA parameters often vary significantly across low-rank matrices [32]. In particular, down-projection matrices ($\boldsymbol{A}$) exhibit smaller gradient norms than up-projection matrices ($\boldsymbol{B}$), rendering them *highly vulnerable* to DP noise. Intuitively, blindly adding noise to LoRA parameters without accounting for structural asymmetry can overwhelm sensitive features in $\boldsymbol{A}$, severely distorting the model's original update direction. To mitigate the negative impacts of DP noise, it is essential to identify *parameter sensitivity* and then *adaptively* redistribute noise [33]. Yet, existing adaptive DP techniques perform clipping and perturbing either at a fine-grained *scalar level* [44, 45] or at a coarse-grained *layer level* [18, 37]. Though these methods are generally suitable for small-scale models, they become *computationally prohibitive* when applied to LLMs or ignore the structural differences across LoRA matrices, resulting in *excessive noise injection*.

The privacy-utility trade-off of private FedLoRA is further worsened by *client heterogeneity*, an intrinsic feature of real-world FL, which leads to undermined model performance with slow convergence [32, 46]. In practice, clients possess different capabilities in terms of hardware and communication resources, and are capable of training models with capacities that match their resources. To fully utilize clients' resources, several works focus on heterogeneous models (e.g., *LoRA rank*), such as zero-padding [9, 16], singular value decomposition [5, 42], or stacking [39]. However, they often yield sub-optimal performance and incur high communication overhead due to full transmission of LoRA modules. Moreover, the data across clients are non-independent and identically distributed (*non-IID*), which is exacerbated by DP noise [47]. This results in discrepancies in local models that severely diverge from the global model, impairing the privacy-utility trade-off as more communication rounds amplify the accumulated noise. Despite many efforts to improve FL efficiency under *data heterogeneity* [23, 36], their performance is still far from being satisfactory. Our empirical observations show that LoRA modules with lower ranks are more vulnerable to non-IID data due to their limited generalization ability, whereas higher-ranked modules, carrying more information, are susceptible to overfitting. Hence, *jointly addressing* DP and client heterogeneity is vital for improving the privacy-utility trade-off.

To address the dual challenges, we propose iP-FedLoRA, an improved Private FedLoRA framework that strikes a good privacy-utility balance under both LoRA structural asymmetry and client heterogeneity. Specifically, iP-FedLoRA implements matrix-wise differentially private fine-tuning, which rigorously estimates the sensitivity of local LoRA matrices to adaptively allocate DP noise, thereby preserving the utility of specific low-rank modules that are fragile to coarse-grained noise. A rank-compensated update regularization is further performed to prevent clients with smaller LoRA ranks from deviating severely due to DP noise. Moreover, iP-FedLoRA employs noise-resilient knowledge distillation (KD [22]) to transfer knowledge among model-heterogeneous clients. Unlike naive aggregation, the server actively filters out DP-induced noise by selectively transferring only high-confidence knowledge, ensuring robust heterogeneous LoRA aggregation. We *extend* iP-FedLoRA to support *large-scale* FedLoRA systems involving vast numbers of clients. We summarize our main contributions as follows:

- We examine the challenges posed by the amplified DP noise impact and inherent client heterogeneity in current differentially private FedLoRA systems, and propose iP-FedLoRA to enhance the privacy-utility trade-off.
- iP-FedLoRA implements *matrix-wise* DP to adaptively allocate noise based on LoRA matrix sensitivity, thus effectively preserving parameter utility. A *rank-compensated* regularization is also devised to anchor fragile low-rank models against noise-induced drift under non-IID data.
- iP-FedLoRA achieves efficient heterogeneous LoRA aggregation via *noise-resilient* federated distillation. Using correlation-guided knowledge aggregation and confidence-based knowledge selection, we can filter DP noise for *robust* knowledge transfer.
- Extensive experiments validate the superiority of iP-FedLoRA in model performance and convergence. In particular, iP-FedLoRA improves model accuracy by up to 3.8% compared to baselines.

2 Preliminaries and Motivation

2.1 A Primer of FedLoRA

Fine-tuning LLMs with LoRA. Contemporary LLMs are built on the Transformer architecture [41]. A crucial component of this architecture is multi-head attention, which employs query, key, value and output projection matrices to model intricate relationships and capture dependencies within sequences [34]. Specifically, the query, key, and value vectors are split into multiple heads to compute the attention function in parallel, with the outputs of each head concatenated and linearly projected to yield the final output.

LoRA is a popular PEFT method that adapts LLMs to a domain-specific task. It updates the pre-trained model with *low-rank* matrices. Specifically, LoRA freezes the pre-trained weight matrix of LLM parameters $\boldsymbol{W}_0 \in \mathbb{R}^{d\times k}$ and adapts it with the update $\Delta\boldsymbol{W} \in \mathbb{R}^{d\times k}$, defined as the product of two trainable low-rank matrices $\boldsymbol{B} \in \mathbb{R}^{d\times r}$ and $\boldsymbol{A} \in \mathbb{R}^{r\times k}$, i.e., $\Delta\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$. As LoRA *rank* $r \ll \min\{d, k\}$, the number of trainable parameters is reduced by an order of $O(r/\min(d, k))$ compared to full-parameter fine-tuning.

FedLoRA. Consider a federated LoRA system with a central server and N clients $\boldsymbol{N} = \{1, \cdots, N\}$ that collaboratively fine-tune a shared LLM parameterized by $\boldsymbol{W}_0$ without exposing their raw data. Each client i holds a local training dataset $\mathcal{D}_i$ containing $D_i = |\mathcal{D}_i|$ samples, with the local model parameterized by weight matrices $\boldsymbol{W}_i = \{\boldsymbol{W}_{i,j,m}\}$, where j and m are indices for the Transformer layer and weight matrix type (e.g., query, key, value, or output), respectively. Let $l(\boldsymbol{W}_i, x)$ be the loss value computed for model $\boldsymbol{W}_i$ at sample x. The local loss of client i is $l(\boldsymbol{W}_i) = \frac{1}{D_i}\sum_{x\in\mathcal{D}_i} l(\boldsymbol{W}_i, x)$. Under the coordination of the server, clients jointly learn trainable parameters $\boldsymbol{\omega} \triangleq [\boldsymbol{A}, \boldsymbol{B}]$ that minimize the global loss:

$$\min_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{W}) \triangleq \min_{\boldsymbol{\omega}} \frac{1}{N}\sum\nolimits_{i\in\boldsymbol{N}} \frac{D_i}{D} l(\boldsymbol{W}_i), \tag{1}$$

where $\boldsymbol{W} = \boldsymbol{W}_0 + \Delta\boldsymbol{W} = \boldsymbol{W}_0 + \boldsymbol{B}\boldsymbol{A}$ is the global model, and $D = |\cup_{i\in\boldsymbol{N}} \mathcal{D}_i|$ denotes the entire training dataset size.

A common method to solve Eq. (1), such as FedPETuning [53], proceeds in rounds of synchronization. In each round t, the server first randomly selects a subset of clients $\mathcal{P}^t \in \boldsymbol{N}$ and broadcasts global LoRA parameters $\boldsymbol{\omega}^t = [\boldsymbol{A}^t, \boldsymbol{B}^t]$ to them. By setting $\boldsymbol{\omega}_i^{t,0} \leftarrow \boldsymbol{\omega}^t$, each selected client $i \in \mathcal{P}^t$ runs K steps of stochastic gradient descent (SGD) on its local dataset using learning rate η, i.e., $\boldsymbol{\omega}_i^{t,k} = \boldsymbol{\omega}_i^{t,k-1} - \eta\nabla l(\boldsymbol{\omega}_i^{t,k-1}), k = 1, \cdots, K$, and sends the model update $\Delta\boldsymbol{\omega}_i^t = \boldsymbol{\omega}_i^{t,K} - \boldsymbol{\omega}_i^t$ to the server. Upon receiving these updates, the server employs federated aggregation techniques like FedAvg [29] to build a new global model $\boldsymbol{\omega}^{t+1} = [\boldsymbol{A}^{t+1}, \boldsymbol{B}^{t+1}]$ for the next round. The above process is repeated until convergence, where only lightweight LoRA updates are exchanged.

However, there is increasing evidence that the exchanged trainable parameters or updates of clients could be exploited by adversaries to infer their private information, the disclosure of which still has the risk of raw data leakage [15, 32].

2.2 Differentially Private Learning

As the gold standard of privacy protection, DP provides provable guarantees against the identification of private data in a dataset [13, 14]. Our work focuses on *client-level DP* [26, 52] defined in *Definition* 1, a popular DP variant in FL that ensures that adversaries cannot determine whether one client participates in training or not, thus protecting client privacy.

Definition 1 (**Client-Level Differential Privacy** [13]). *A randomized mechanism $\mathcal{M}$ is (ϵ, δ)-DP if for any adjacent datasets $\mathcal{D}, \mathcal{D}'$ differing by adding or removing all samples of any client and for any possible subset of outputs $\mathcal{U} \subseteq Range(\mathcal{M})$, it holds that:*

$$\Pr(\mathcal{M}(\mathcal{D}) \in \mathcal{U}) \leq e^{\epsilon} \cdot \Pr(\mathcal{M}(\mathcal{D}') \in \mathcal{U}) + \delta,$$

where budget ϵ controls privacy guarantee and δ is residual probability, with smaller values indicating stronger protection.

Differentially private FedAvg (DP-FedAvg) has been widely used to achieve client-level DP in FL [26]. Compared to the vanilla FedAvg, DP-FedAvg introduces two critical procedures before sending local updates to the server in each round t:

- *Update clipping*. Each client $i \in \mathcal{P}^t$ clips its model update $\Delta\boldsymbol{\omega}_i^t$ in ℓ_2-norm using the *clipping threshold* S to bound the local update, i.e., $\Delta\widetilde{\boldsymbol{\omega}}_i^t = \Delta\boldsymbol{\omega}_i^t \cdot \min\left(1, \frac{S}{\|\Delta\boldsymbol{\omega}_i^t\|_2}\right)$, bounding its contribution to the global update within a fixed range.

Table 1: Accuracy (%) of private FedLoRA fine-tuning with varying privacy budget ϵ and LoRA rank r.

	Non-DP	$\epsilon = 8$	$\epsilon = 4$	$\epsilon = 2$
$r = 1$	88.0	$86.4_{\downarrow 1.4}$	$85.3_{\downarrow 2.7}$	$83.5_{\downarrow 4.5}$
$r = 16$	90.6	$87.2_{\downarrow 3.4}$	$86.0_{\downarrow 4.6}$	$83.8_{\downarrow 6.8}$
$r = 64$	**91.5**	**87.6**$_{\downarrow 3.9}$	**86.2**$_{\downarrow 5.3}$	**83.7**$_{\downarrow 7.8}$

Table 2: Performance degradation of layer-wise and scalar-wise DP for LoRA fine-tuning compared to Non-DP.

	Non-DP	Layer-wise	Scalar-wise
Accuracy(%)	84.6	$82.6_{\downarrow 2.0}$	$83.8_{\downarrow 0.8}$
Comp. time (hours)	1.18	$1.32_{\uparrow 0.14}$	$1.61_{\uparrow 0.43}$

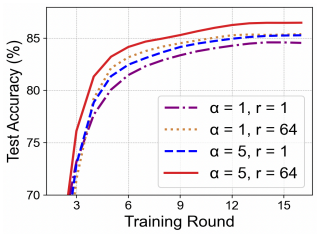


Figure 1: Accuracy with varying data non-IIDness α.

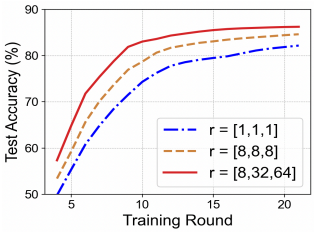


Figure 2: Accuracy with different LoRA rank deployments.

- *Noise perturbation*. To further obscure these contributions, the clipped local updates are then perturbed by the *Gaussian noise* with mean 0 and standard deviation $S\sigma$, i.e., $\Delta\widetilde{\boldsymbol{\omega}}_i^t = \Delta\widetilde{\boldsymbol{\omega}}_i^t + \mathcal{N}(0, S^2\sigma^2)$, where σ controls the scale of noise.

2.3 Motivation

Motivation 1: Structural asymmetry and rank variations in LoRA modules can amplify DP noise impact. While DP inherently compromises model accuracy, this issue is more salient in FedLoRA. In general, LoRA modules possess an intrinsic *structural asymmetry* [17], i.e., up-projection matrices (i.e., $\boldsymbol{B}$) which map extracted features back to the high-dimensional output space, exhibit larger gradient norms than down-projection matrices (i.e., $\boldsymbol{A}$). Standard DP enforces a shared clipping threshold on both matrices, causing sensitive features in $\boldsymbol{A}$ to be easily overwhelmed by noise. This DP noise sensitivity is further *magnified* by the LoRA rank. Table 1 offers a comparative analysis of private FedLoRA by fine-tuning the RoBERTa-base model [25] on QNLI dataset [35]. As rank r increases, the model becomes more susceptible to strict privacy budgets (smaller ϵ). That is, blindly adding noise to LoRA parameters without accounting for structural differences can severely degrade the utility. Prior DP schemes often overlook the *structural nuance*: *scalar-level* DP [44] that captures fine-grained sensitivity is computationally prohibitive for LLMs, while *layer-level* DP [37] ignores the internal $\boldsymbol{A}/\boldsymbol{B}$ disparity in each LoRA module, resulting in excessive noise injection (Table 2). To better balance privacy protection and model performance, a more precise clipping/perturbing granularity that respects LoRA's *structural asymmetry* is required.

Motivation 2: Client heterogeneity adversely impacts the convergence of FedLoRA. Clients' data distributions often vary

Table 3: Main notations.

Symbol	Description
$\boldsymbol{N}, \mathcal{D}_i$	Client set and local dataset of client i
$\boldsymbol{W}_0, \boldsymbol{W}_i$	Frozen backbone and client model
$\boldsymbol{A}_i, \boldsymbol{B}_i, r_i$	LoRA matrices and rank of client i
$\Delta\boldsymbol{\omega}_i^t$	Local LoRA update in round t
$\mathcal{P}^t, K$	Selected clients and local steps
$s_{i,j,m}, H_{i,j,m}$	Matrix-wise threshold and sensitivity
$\sigma_{i,j,m}, \sigma_*$	Matrix noise scale and base noise scale
ϵ, δ	DP privacy budget
μ_1, μ_2, μ_3	Regularization, clipping, and KD factors
$\widehat{\mathcal{D}}, f_i^t$	Public dataset and local logits
f^t, f_s^t	Aggregated and selected knowledge
ϕ, τ	Selection threshold and KD temperature

considerably in labels, features and quantities in real-world scenarios, which can be exacerbated by the randomness of DP noise [47]. This non-IID data distribution causes local model drift and adversely impacts model convergence, worsening the *privacy-utility* trade-off, as more communication rounds lead to more noise injection. As illustrated in Fig. 1 on QNLI dataset, where training data follow the *Dirichlet distribution* $\mathrm{Dir}(\alpha)$ [20], larger performance degradation is observed when data distributions become more divergent (smaller α). Additionally, LoRA modules with lower ranks (smaller r) lack the capacity to generalize under non-IID data, making them much more brittle to DP noise than higher-rank modules.

Furthermore, existing FedLoRA works mostly focus on the model-homogeneous setting [4, 53], where all clients share identical model structures (i.e., LoRA rank). Thus, the rank size has to be limited to accommodate the most *resource-constrained* client, wasting the potential of resource-abundant clients capable of training robust, high-rank models to guide global training. To underscore this point, we refer to Fig. 2 on MNLI dataset [35]. Compared to homogeneous LoRA ranks (e.g., $r = \{1, 1, 1\}$ and $r = \{8, 8, 8\}$), heterogeneous LoRA deployment (e.g., $r = \{8, 32, 64\}$) accelerates the fine-tuning process by adaptively sizing clients' LoRA ranks with available resources. This presents a new challenge: how to effectively *distribute* and *aggregate* LoRA modules of diverse ranks across clients.

3 iP-FedLoRA Design

In this section, we describe iP-FedLoRA, an improved private FedLoRA framework to address the above challenges. Table 3 summarizes the main notations used in this paper.

3.1 Overview

Threat model. We assume that the server and clients are *honest-but-curious*, which is commonly adopted in most existing works on privacy-preserving FL [24, 46]. That is, they will honestly follow the protocol design, yet are curious about a target client's private information (e.g., memberships or data distributions) and want to infer it from the shared messages. This assumption aligns with cross-silo scenarios (e.g., institutional collaboration), where clients are incentivized to cooperatively train a high-utility model rather than disrupting the training process.

Design goal. The goal is to build an improved differentially private FedLoRA framework that achieves the objectives below.

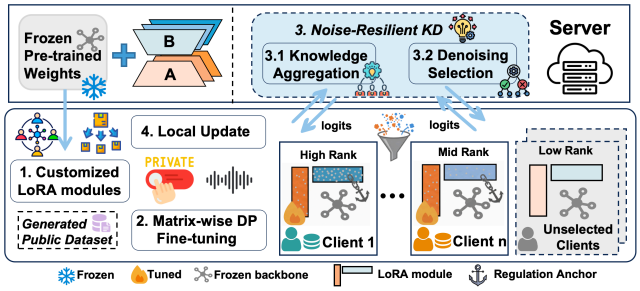


Figure 3: An overview of iP-FedLoRA architecture.

- *Privacy Protection*. It should protect clients' privacy during federated fine-tuning. Neither the server nor the clients can reveal any private information about individual clients.
- *Accuracy*. It should effectively mitigate the adverse impact of DP noise, so as to preserve high model accuracy.
- *Efficiency*. It is expected to be efficient in federated fine-tuning among heterogeneous clients, which often result in undermined training efficiency with slow convergence.

Architecture. We achieve the above goals through matrix-wise private local fine-tuning and KD-based heterogeneous federated aggregation. Fig. 3 depicts the main architecture of iP-FedLoRA, which proceeds in rounds of synchronization.

① *LoRA Module Customization and Client Selection.* The server first sends the backbone $\boldsymbol{W}_0$ and a customized LoRA module $\boldsymbol{\omega}_i^0 = [\boldsymbol{A}_i^0, \boldsymbol{B}_i^0]$ with rank r_i to each client $i \in \boldsymbol{N}$ based on its available resources. At the beginning of each round t, the server randomly selects a subset of clients $\mathcal{P}^t \in \boldsymbol{N}$ as participants.

② *Private Local Fine-Tuning with Update Regularization.* Each selected client $i \in \mathcal{P}^t$ performs matrix-wise differentially private fine-tuning with regularization to obtain the private LoRA update $\Delta\overline{\boldsymbol{\omega}}_i^t$ and update local model $\boldsymbol{\omega}_i^t$.

③ *Heterogeneous Federated LoRA Aggregation.* Each client $i \in \mathcal{P}^t$ computes local knowledge f_i^t in the form of logits for model $\boldsymbol{\omega}_i^t$ using public dataset $\widehat{\mathcal{D}}$ and sends it to the server. The server aggregates $\{f_i^t\}_{i\in\mathcal{P}^t}$ to derive global knowledge f^t, then selects *high-quality* knowledge f_s^t and dispatches it to all clients.

④ *Local Training and Update.* Each client $i \in \boldsymbol{N}$ performs the KD process with global knowledge f_s^t in Eq. (7) and updates the local LoRA module for the next round.

The details of iP-FedLoRA are shown in Alg. 1. Initially, the server *customizes* LoRA modules for clients (line 1). In each round t, it selects participating clients $\mathcal{P}^t$ (line 3), who perform *private local fine-tuning with regularization* for private LoRA update and then send their local knowledge to the server (lines 10-14). The server then *aggregates and selects high-quality knowledge* to distribute to all clients (lines 4-7). Finally, clients conduct the KD process for local model updates (lines 15-17).

3.2 Matrix-Wise Differentially Private Local Fine-Tuning with Update Regularization

During local fine-tuning, private FedLoRA systems typically clip and perturb clients' LoRA updates with random noise before sending them to the server. Yet, current methods often ignore the structural fragility of *low-rank* matrices, which are far more brittle to

Algorithm 1: iP-FedLoRA

Input: Clients $\boldsymbol{N}$, round number T, KD process steps K_d
Output: Private federated fine-tuned model $\boldsymbol{W}$

// Server
Initialize and send $\boldsymbol{W}_0$ and $\boldsymbol{\omega}_i^0 = [\boldsymbol{A}_i^0, \boldsymbol{B}_i^0]$ to each client $i \in \boldsymbol{N}$
for *each round* $t \in \{0, \cdots, T-1\}$ **do**
  Select participating clients $\mathcal{P}^t$ from $\boldsymbol{N}$
  Receive local knowledge f_i^t from selected client $i \in \mathcal{P}^t$
  $f^t \leftarrow$ Conduct knowledge aggregation in Sec. 3.3
  $f_s^t \leftarrow$ Conduct knowledge selection in Sec. 3.3
  Send selected knowledge f_s^t to all clients

// Client $i \in \boldsymbol{N}$
Receive $\boldsymbol{W}_0$ and $\boldsymbol{\omega}_i^0$ from server for LoRA module customization
for *each round* $t \in \{0, \cdots, T-1\}$ **do**
  if $i \in \mathcal{P}^t$ **then**
    $\Delta\widetilde{\boldsymbol{\omega}}_i^t \leftarrow$ Run private local fine-tuning by Alg. 2
    $\boldsymbol{\omega}_i^t \leftarrow \boldsymbol{\omega}_i^t + \Delta\widetilde{\boldsymbol{\omega}}_i^t$
    Update local model $\boldsymbol{W}_i^t = \boldsymbol{W}_0 + \boldsymbol{B}_i^t \boldsymbol{A}_i^t = \boldsymbol{W}_0 + \boldsymbol{\omega}_i^t$
    Calculate and send local knowledge f_i^t to server
  Receive global aggregated knowledge f_s^t from server
  $\boldsymbol{\omega}_i^{t+1} \leftarrow$ `KnowledgeDistill`(f_s^t)
  Update local model $\boldsymbol{W}_i^{t+1} = \boldsymbol{W}_0 + \boldsymbol{B}_i^{t+1} \boldsymbol{A}_i^{t+1}$
Function `KnowledgeDistill`(f_s^t):
  for *each step* $k_d \in \{0, 1, \cdots, K_d - 1\}$ **do**
    for *batch* $\boldsymbol{b} \in \{(\widehat{x}_j, \widehat{y}_j, f_s^t(\widehat{x}_j))\}_{\widehat{x}_j \in \widehat{\mathcal{D}}}$ **do**
      $\boldsymbol{\omega}_i^{t+1} \leftarrow \boldsymbol{\omega}_i^t - \eta \nabla l_{KD}(\boldsymbol{W}_i^t, \boldsymbol{b})$ in Eq. (7)
  return $\boldsymbol{\omega}_i^{t+1}$

noise-induced drift than high-rank ones. We propose a matrix-wise differentially private fine-tuning scheme, where update regularization is also performed to mitigate the noise-induced model drift.

Rank-compensated LoRA regularization. In vanilla FedLoRA, clients optimize local models based solely on private data. Given the compound impact of non-IID distributions and DP noise, local updates can deviate drastically from the global optimum. We identify a critical *reliability gap*: clients with smaller LoRA ranks r_i are more susceptible to being biased by noise-induced model drift, whereas higher-rank clients exhibit greater robustness. Existing scalar regularization methods fail to account for this structural disparity. To bridge this gap, we enforce a *rank-aware regularization* that acts as a rank compensator. For each client $i \in \mathcal{P}^t$, the optimization objective is reformulated to dynamically adjust the constraint strength based on LoRA rank:

$$l\left(\boldsymbol{W}_i^t\right) = l\left(\boldsymbol{W}_i^t\right) + \frac{\mu_1}{2r_i} \left\|\Delta\boldsymbol{W}_i^t - \Delta\boldsymbol{W}_i^{t-1}\right\|_2^2, \tag{2}$$

where μ_1 is a smoothing factor and $\Delta\boldsymbol{W}_i^{t-1}$ is the update from the previous round. We scale the regularization term by $1/r_i$ to impose a *stricter penalty* on low-rank clients, so as to anchor their fragile updates closer to the global and prevent noise-driven divergence. Also, it allows high-rank clients (smaller penalty) with more flexibility to learn complex local features. Our introduced *asymmetric constraint* ensures that all heterogeneous clients contribute reliable updates despite their varying sensitivities to noise.

Matrix-wise adaptive DP mechanism. Due to the structural asymmetry of LoRA, where up-projection matrices $\boldsymbol{B}$ typically exhibit larger gradients than down-projection matrices $\boldsymbol{A}$, standard layer-level DP with a unified threshold injects excessive noise into the vulnerable $\boldsymbol{A}$ matrices. As such, we propose a *matrix-wise* adaptive mechanism to decouple the clipping and perturbation process of $\boldsymbol{A}$ and $\boldsymbol{B}$, where each low-rank matrix is handled based on its specific sensitivity to prevent performance degradation.

Let $\Delta\widetilde{\boldsymbol{\omega}}_i^t$ denote LoRA updates after clipping but before noise perturbation, and $\Delta\overline{\boldsymbol{\omega}}_i^t$ denote the updates after perturbation. Specifically, client i clips each weight matrix m (including A and B) at layer j using an independent threshold $s_{i,j,m}$ as $\Delta\widetilde{\boldsymbol{\omega}}_{i,j,m} \leftarrow \Delta\boldsymbol{\omega}_{i,j,m}^t \min\left(1, \frac{s_{i,j,m}}{\|\Delta\boldsymbol{\omega}_{i,j,m}^t\|_2}\right)$ and injects noise $\sigma_{i,j,m} \leftarrow 2\sqrt{M}\sigma_* s_{i,j,m}$, where M is the total weight matrix number and σ_* is the overall noise scale. To theoretically quantify the impact of this granularity, the expected *mean-square error* of matrix-wise DP is bounded by:

$$\begin{aligned}
&\frac{1}{M}\mathbb{E}\left(\sum_{j,m}\frac{1}{|\boldsymbol{\omega}_{i,j,m}|}\|\Delta\overline{\boldsymbol{\omega}}_{i,j,m}^t - \Delta\boldsymbol{\omega}_{i,j,m}^t\|_F^2\right)\\
\le&\frac{2}{M}\mathbb{E}\left(\sum_{j,m}\frac{1}{|\boldsymbol{\omega}_{i,j,m}|}\left[\|\Delta\widetilde{\boldsymbol{\omega}}_{i,j,m}^t - \Delta\boldsymbol{\omega}_{i,j,m}^t\|_F^2 + \|\Delta\overline{\boldsymbol{\omega}}_{i,j,m}^t - \Delta\widetilde{\boldsymbol{\omega}}_{i,j,m}^t\|_F^2\right]\right)\\
\le&\underbrace{\frac{2}{M}\sum_{j,m}\frac{1}{|\boldsymbol{\omega}_{i,j,m}|}\max\left(0, \|\Delta\boldsymbol{\omega}_{i,j,m}^t\|_2 - s_{i,j,m}\right)^2}_{\text{update clipping error}} + \underbrace{8\sum_{j,m}\sigma_*^2 (s_{i,j,m})^2}_{\text{noise perturbation error}},
\end{aligned} \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius norm and $|\boldsymbol{\omega}_{i,j,m}|$ is the parameter count. Since the optimal $s_{i,j,m}$ differs significantly between low-rank matrices, i.e., $\boldsymbol{A}$ and $\boldsymbol{B}$, a unified threshold fails to minimize the errors, underscoring the necessity for independent thresholds.

Momentum-based dynamic thresholding. Since LoRA updates often exhibit high variance under DP, static clipping thresholds for $\boldsymbol{A}$ and $\boldsymbol{B}$ are ineffective. To address this, we leverage the *post-processing* property of DP [14] to estimate dynamic thresholds based on the momentum of *historical noisy updates*. Specifically, we define the clipping threshold $s_{i,j,m}$ for the m-th matrix at round t as a *privacy-free predictor*:

$$s_{i,j,m} \leftarrow \mu_2 \left\|\boldsymbol{v}_{i,j,m}^t\right\|_2, \tag{4}$$

where $\mu_2 > 0$ is the scaling factor and $\boldsymbol{v}_{i,j,m}^t \leftarrow \gamma \boldsymbol{v}_{i,j,m}^{t-1} + (1-\gamma)\Delta\overline{\boldsymbol{\omega}}_{i,j,m}^{t-1}$ tracks the moving average of the published noisy updates $\Delta\overline{\boldsymbol{\omega}}^{t-1}$. This approach adaptively captures the distinct scales of $\boldsymbol{A}$ and $\boldsymbol{B}$ *without consuming privacy budget*, thereby minimizing noise variance while preserving informative update directions.

Sensitivity-aware noise allocation. We next align the noise scale with the estimated sensitivity. Given that $\mathcal{D}$ and $\mathcal{D}'$ are neighboring datasets differing by adding or removing all samples of any client, the ℓ_2-sensitivity of client i's m-th LoRA matrix is

$$H_{i,j,m} = \max_{\mathcal{D},\mathcal{D}'} \left\|\Delta\boldsymbol{\omega}_{i,j,m}^t(\mathcal{D}) - \Delta\boldsymbol{\omega}_{i,j,m}^t(\mathcal{D}')\right\|_2 \le 2s_{i,j,m}. \tag{5}$$

Inspired by [44] and in line with Eq. (5), noise scale $\sigma_{i,j,m}$ added to $\Delta\overline{\boldsymbol{\omega}}_{i,j,m}^t$ for privacy protection needs to satisfy $\sum_{j,m}\frac{H_{i,j,m}^2}{\sigma_{i,j,m}^2} \le \frac{1}{\sigma_*^2}$. Hence, we set $\sigma_{i,j,m} = \sqrt{M}\sigma_* H_{i,j,m}$ to ensure that "sparse" matrices (i.e., $\boldsymbol{A}$) receive less noise, while "dense" matrices (i.e., $\boldsymbol{B}$) are accommodated with higher thresholds, effectively preserving LoRA's structural integrity under the overall (ϵ, δ)-DP guarantee.

The whole private local fine-tuning process is detailed in Alg. 2. Every selected client performs local LoRA update with regularization (lines 2-11). Subsequently, the adaptive clipping threshold is

Algorithm 2: Private local fine-tuning for client i

Input: Round t, overall noise scale σ_*, local steps K, LoRA rank r_i, total number of matrices M, initial threshold, $s_{init} = 1$, momentum $\boldsymbol{v}_i = \mathbf{0}$
Output: Private fine-tuned LoRA update $\Delta\overline{\boldsymbol{\omega}}_i^t$

1 $\Delta \boldsymbol{W}_i^{t,0} \leftarrow \Delta \boldsymbol{W}_i^{t-1} = \boldsymbol{B}_i^{t-1}\boldsymbol{A}_i^{t-1}$
2 **for** *each step* $k \in \{0, \cdots, K-1\}$ **do**
3 Sample data samples $\boldsymbol{b}^{t,k} \in \mathcal{D}_i$
4 $l(\boldsymbol{W}_i^{t,k-1}) \leftarrow l(\boldsymbol{W}_i^{t,k-1}) + \frac{\mu_1}{2r_i}\|\Delta \boldsymbol{W}_i^{t,k-1} - \Delta \boldsymbol{W}_i^{t-1}\|_2^2$
5 $g_i^{t,k} \leftarrow \frac{1}{|\boldsymbol{b}^{t,k}|}\sum_{(x,y)\in \boldsymbol{b}^{t,k}} \nabla l(\boldsymbol{W}_i^{t,k-1}, x, y)$
6 **if** $k = 0$ **then** $\hat{g}_i^{t,k} = g_i^{t,k}$
7 **else** $\hat{g}_i^{t,k} \leftarrow \gamma \hat{g}_i^{t,k-1} + (1-\gamma) g_i^{t,k}$
8 $\boldsymbol{\omega}_i^{t,k} \leftarrow \boldsymbol{\omega}_i^{t,k-1} - \eta \hat{g}_i^{t,k}$
9 $\Delta\boldsymbol{\omega}_i^t \leftarrow \{\Delta \boldsymbol{A}_i^t, \Delta \boldsymbol{B}_i^t\} = \boldsymbol{\omega}_i^{t,K} - \boldsymbol{\omega}_i^{t,0}$
10 **for** *each weight matrix* m *at each layer* j **do**
11 **if** $t = 0$ **then** $s_{i,j,m} \leftarrow s_{init}$
12 **else** $s_{i,j,m} \leftarrow \mu_2 \|\boldsymbol{v}_{i,j,m}^t\|_2$
13 $H_{i,j,m} \leftarrow 2s_{i,j,m}, \sigma_{i,j,m} \leftarrow \sqrt{M}\sigma_* H_{i,j,m}$
  // Update clipping
14 $\Delta\overline{\boldsymbol{\omega}}_{i,j,m}^t \leftarrow \Delta\boldsymbol{\omega}_{i,j,m}^t \min(1, s_{i,j,m}/\|\Delta\boldsymbol{\omega}_{i,j,m}^t\|_2)$
  // Noise perturbation
15 $\Delta\overline{\boldsymbol{\omega}}_{i,j,m}^t \leftarrow \Delta\overline{\boldsymbol{\omega}}_{i,j,m}^t + \mathcal{N}(0, (\sigma_{i,j,m})^2)$
16 $\boldsymbol{v}_{i,j,m}^{t+1} \leftarrow \gamma \boldsymbol{v}_{i,j,m}^t + (1-\gamma)\Delta\overline{\boldsymbol{\omega}}_{i,j,m}$
17 **return** $\Delta\overline{\boldsymbol{\omega}}_i^t$

calculated, followed by update clipping and noise perturbation for each LoRA update matrix (lines 12–17).

3.3 Noise-Resilient Heterogeneous Aggregation

Heterogeneous LoRA ranks across clients create a *structural barrier* against direct parameter aggregation. This is exacerbated by the injected matrix-wise DP noise, which corrupts parameter utility and renders naive averaging catastrophic. To bridge this gap, iP-FedLoRA employs a *noise-resilient* federated distillation scheme. Unlike standard KD [10], our approach is specifically tailored to filter out DP-induced noise and enforce consensus among model-heterogeneous clients. The overall process is integrated into Alg. 1.

KD-based heterogeneous LoRA aggregation. Instead of exchanging noise-perturbed LoRA matrices [16, 39], we leverage KD as a *semantic bridge* that facilitates knowledge transfer over clients with heterogeneous LoRA. In particular, clients project private knowledge onto an unlabeled public dataset $\widehat{\mathcal{D}}$, which is constructed via generative synthesis in Appendix C. At a high level, each selected client $i \in \mathcal{P}^t$ computes local knowledge in the *form of logits* for model $\boldsymbol{W}_i^t$, i.e., $f_i^t(x) = \text{Forward}(x; \boldsymbol{W}_0 + B_i^t A_i^t), \forall x \in \widehat{\mathcal{D}}$. This knowledge is subsequently sent to the server for aggregation, and redistributed to clients for distillation, formulated as $\min_{\boldsymbol{\omega}} \mathbb{E}_{x_j \sim \widehat{\mathcal{D}}} \tau^2 D_{KL}(\psi(f_i^t/\tau)\|\psi(f^t/\tau))$, where the global knowledge is $f^t = \frac{1}{|\mathcal{P}^t|}\sum_{i=1}^{|\mathcal{P}^t|} f_i^t$, D_{KL} is the Kullback-Leibler divergence, τ is the temperature, and $\psi(\cdot)$ is the activation function.

Correlation-guided knowledge aggregation. Nevertheless, model structure heterogeneity can result in disparate confidence levels in clients' local knowledge. Low-rank clients are more easily impacted by DP noise, producing volatile logits, whereas high-rank clients attain highly stable predictions. Simple averaging would allow noisy clients to pollute global knowledge. To mitigate this, we devise a weighted aggregation scheme based on *knowledge correlation*. The intuition is that valid knowledge should exhibit consensus, while DP-induced errors are random and uncorrelated.

Specifically, for each local knowledge f_i^t, the server calculates the *Pearson correlation coefficient* with the ensemble, i.e., $\rho_i^t = \frac{\sum_{\widehat{x}_j}(f_i^t(\widehat{x}_j)-\overline{f_i^t})(R(\widehat{x}_j)-\overline{R})}{\sqrt{\sum_{\widehat{x}_j}(f_i^t(\widehat{x}_j)-\overline{f_i^t})^2}\sqrt{\sum_{\widehat{x}_j}(R(\widehat{x}_j)-\overline{R})^2}}$, where $f_i^t(\widehat{x}_j)$ is the logit for sample $\widehat{x}_j$ from $\widehat{\mathcal{D}}$, $\overline{f_i^t} = \sum_{\widehat{x}_j} f_i^t(\widehat{x}_j)/|\widehat{\mathcal{D}}|$, and $R(\widehat{x}_j) = \sum_{i=1}^{|\mathcal{P}^t|} f_i^t(\widehat{x}_j)/|\mathcal{P}^t|$ is the average knowledge with $\overline{R} = \sum_{\widehat{x}_j} R(\widehat{x}_j)/|\widehat{\mathcal{D}}|$. Then the aggregation weight for knowledge f_i^t is $\lambda_i^t = e^{\rho_i^t}/\sum_{i=1}^{|\mathcal{P}^t|} e^{\rho_i^t}$. In other words, we *prioritize* reliable clients while down-weighting those whose predictions diverge due to excessive noise or limited rank, yielding the aggregated knowledge:

$$f^t(\widehat{x}_j) = \textstyle\sum_{i\in\mathcal{P}^t} \lambda_i^t \times f_i^t(\widehat{x}_j). \tag{6}$$

Confidence-based denoising selection. Even with weighted aggregation, the global knowledge may still carry accumulated noise, manifesting as *high-entropy* (ambiguous) predictions that can mislead local training. To prevent this "negative transfer", the server filters out DP-induced noise by selectively transferring only high-confidence knowledge. For each sample $\widehat{x}_j$, the server calculates the maximum probability $\phi_j = \max(\psi(f^t(\widehat{x}_j)))$ as the prediction. Only predictions exceeding a confidence threshold ϕ are retained and distributed to clients, formalized as $f_s^t(\widehat{x}_j) = f^t(\widehat{x}_j) \cdot \mathbf{1}_{\{\phi_j > \phi\}}, \forall \widehat{x}_j \in \widehat{\mathcal{D}}$. This denoising selection ensures that clients learn exclusively from *trustworthy* signals that have survived the DP perturbation. Upon receiving f_s^t, each client $i \in \boldsymbol{N}$ calculates *pseudo-labels* $\widehat{y}_j = \arg\max_{\text{labels}} f_s^t(\widehat{x}_j)$ and updates $\boldsymbol{W}_i^t$ using *cross-entropy* loss l_{CE} and *KL divergence* D_{KL} for each batch $\boldsymbol{b} \in \{(\widehat{x}_j, \widehat{y}_j, f_s^t(\widehat{x}_j))\}_{\widehat{x}_j \in \widehat{\mathcal{D}}}$, i.e.,

$$\begin{aligned} l_{KD}(\boldsymbol{W}_i^t, \boldsymbol{b}) = \frac{1}{|\boldsymbol{b}|}\sum\nolimits_{\widehat{x}_j \in \boldsymbol{b}} \Big(&\mu_3 l_{CE}((\widehat{x}_j, \widehat{y}_j), \boldsymbol{W}_i^t) \\ &+ (1-\mu_3) D_{KL}\big(\psi(f_s^t(\widehat{x}_j)/\tau)\|\psi(f_i^t(\boldsymbol{W}_i^t, \widehat{x}_j)/\tau)\big)\Big). \end{aligned} \tag{7}$$

3.4 Extension to Large-Scale FedLoRA Systems

iP-FedLoRA can also be extended to support *large-scale* FedLoRA systems through *client clustering*, *intra-cluster* private local fine-tuning and *inter-cluster* heterogeneous aggregation.

Resource-aware client clustering. Prior clustering methods [10] often rely on data similarity, potentially raising privacy concerns. To avoid this, the server clusters clients into C groups $\mathcal{A}_1, \cdots, \mathcal{A}_C$ based on *resource capabilities* (e.g., data volume, computing power, memory, communication), using *fuzzy membership* [6]. Each client is assigned to the cluster with the highest membership. A *unique* LoRA module $\boldsymbol{\omega}_c = [\boldsymbol{A}_c, \boldsymbol{B}_c]$ is customized for each cluster $\mathcal{A}_c$, with higher-rank modules allocated to resource-abundant clusters. This maintains a uniform LoRA rank within clusters while allowing for inter-cluster heterogeneity. Details of resource-aware client clustering can refer to Appendix A.

Intra-cluster private local fine-tuning. In each round t, the server selects clients $\mathcal{P}_c^t$ from each cluster $\mathcal{A}_c$, where a specific client p_c^t is designated for aggregation. Each selected client $i \in \mathcal{P}_c^t$ performs private fine-tuning with update regularization and sends its private LoRA update $\Delta\overline{\boldsymbol{w}}_i^t$ to the *cluster client* p_c^t. Then, p_c^t employs FedAvg [29] for homogeneous aggregation within the

Table 4: Statistics of the datasets.

Dataset	# Train	# Validation	# Test	Task	Metric
MRPC	3,301	367	408	Paraphrase Identification	F1 score
SST-2	66,675	674	872	Sentiment Analysis	Accuracy
QNLI	103,695	1,048	5,463	QA/NLI	Accuracy
QQP	360,210	3,639	40,430	Duplicate Question Detection	Accuracy
MNLI	388,774	3,928	9,815	Natural Language Inference	Accuracy

cluster, i.e., $\boldsymbol{\omega}_c^t \leftarrow \boldsymbol{\omega}_c^t + \frac{1}{|\mathcal{P}_c^t|}\sum_{i\in\mathcal{P}_c^t}\Delta\overline{\boldsymbol{\omega}}_i^t$ to update the cluster LoRA module $\boldsymbol{\omega}_c^t$ and cluster model $\boldsymbol{W}_c^t$.

Inter-cluster heterogeneous federated aggregation. Each cluster client p_c^t computes knowledge f_c^t on cluster model $\boldsymbol{W}_c^t$ and sends it to the server. The server then aggregates them to derive global knowledge f^t, selects *high-quality* knowledge f_s^t and dispatches it to cluster clients. Finally, each client p_c^t performs KD process with f_s^t to update cluster LoRA module $\boldsymbol{\omega}_c^{t+1}$ and distributes it to all clients in $\mathcal{A}_c$ for local update.

3.5 Privacy and Convergence Analysis

We now present the formal privacy guarantee of iP-FedLoRA, where the proof sketch is given in Appendix B.1.

Theorem 1 (**Privacy Guarantee**). *Let M and I_i denote the number of weight matrices and the number of participating rounds of client i. Given the privacy budget (ϵ, δ), sampling ratio $q = \frac{|\boldsymbol{b}|}{|\mathcal{D}_i|}$ with batch $\boldsymbol{b}$ and integer $\beta > 1 + \frac{log(1/\epsilon)}{\delta}$, if noise scales $\{\sigma_{i,j,m}\}$ for each LoRA matrix m of client i in layer j satisfy $\sum_{j,m}\frac{H_{i,j,m}^2}{\sigma_{i,j,m}^2} \le \frac{1}{\sigma_*^2}$ with σ_* such that $\frac{I_i q\beta}{2\sigma_*^2} + \frac{log(1/\delta)}{\beta-1} \le \epsilon$, iP-FedLoRA achieves (ϵ, δ)-DP for each client.*

We next provide the convergence result under several commonly used assumptions [52], with the proof in Appendix B.2.

Assumption 1 (**Smoothness**). *The local loss function $l(\cdot)$ is L-smooth, such that $\|\nabla l(x) - \nabla l(y)\| \le L\|x - y\|, \forall x, y \in \mathbb{R}$.*

Assumption 2 (**Bounded Gradient**). *There exists $G > 0$ such that local gradient is bounded, i.e., $\|\nabla l(\boldsymbol{W}_i^{t,k})\| \le G$.*

Assumption 3 (**Bounded Variance**). *There exist two constants $\sigma_l > 0$ and $\sigma_g > 0$ such that the variance of every local gradient is bounded for any client i, i.e., $\mathbb{E}_{x\sim\mathcal{D}_i}[\|\nabla l(\boldsymbol{W}_i, x) - \nabla l(\boldsymbol{W}_i)\|^2] \le \sigma_l^2$, and the global variance of the local gradient is bounded, i.e., $\|\nabla l(\boldsymbol{W}_i) - \nabla\mathcal{L}(\boldsymbol{W})\|^2 \le \sigma_g^2$.*

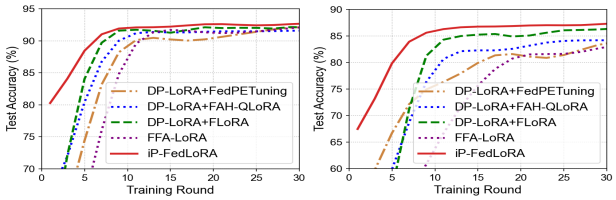
Theorem 2 (**Convergence**). *Under the Assumptions 1-2, for any noise scale in Theorem 1, we have the following convergence result for iP-FedLoRA:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{W}^t)\|^2\right] \le \underbrace{O\left(\frac{1}{\eta KT} + \eta M + \eta^2 K(1+K)\right)}_{\text{from FedAvg}} + \underbrace{O\left(\frac{Mqr(\epsilon + 2log(\frac{1}{\delta}))}{\epsilon^2\eta KN}\right)}_{\text{from matrix-level DP}}.$$

4 Experiments

4.1 Evaluation Setup

Implementation details. We conduct experiments using PyTorch 2.2.1 and CUDA 12.2 on a GPU server equipped with six NVIDIA GeForce RTX 4090 GPUs, where one node serves as the server and the rest simulate 10 clients. Results for the larger-scale iP-FedLoRA involving 50 clients are also presented in Appendix E. Consistent with previous studies [53], we add LoRA to the multi-head attention module and feed-forward network of each Transformer layer.



(a) SST-2 (RoBERTa-base) **(b) MNLI (Llama-3.2-3B)**

Figure 4: Comparison of test accuracy versus global rounds.

Datasets and models. We evaluate iP-FedLoRA on three widely used LLMs and five real-world datasets in Table 4. We fine-tune RoBERTa-base [25] and DeBERTa-v3-base [19] on MRPC, SST-2, and QNLI [35], as well as Llama-3.2-3B [30] on QQP and MNLI [35]. To circumvent the often unrealistic assumption of possessing in-domain real-world public data, we employ a synthetic data generation strategy for data-free KD [27, 54]. Specifically, we leverage the exceptional generation capability of a pre-trained LLM deployed on the cloud server, guided by the task descriptions as detailed in Appendix C. These synthetic samples constitute the public dataset $\widehat{\mathcal{D}}$, ensuring that the server operates without access to real data distributions or private client data, while maintaining the semantic relevance required for KD.

Hyper-parameters. We use the AdamW optimizer and batch size $|\boldsymbol{b}| = 32$. We set privacy parameters $\epsilon = 4$, $\delta = 1e-5$, regularization factor $\mu_1 = 3.2$ and clipping factor $\mu_2 = 1.1$. Details of privacy accounting are provided in Appendix D. We study data heterogeneity using a Dirichlet distribution [20] with $\alpha = 5$ and set client selection ratio to 0.5. For model heterogeneity, we vary the LoRA rank $r \in \{1, 8, 16, 32, 64\}$ and focus on average accuracy. For KD-based heterogeneous LoRA aggregation, we set the balance coefficient $\mu_3 = 0.4$, selection threshold $\phi = 0.6$ and temperature $\tau = 5$.

Baselines. We choose the following baselines for comparisons.

- *DP-LoRA* [49] applies DP-SGD to LoRA for achieving privacy-preserving fine-tuning.
- *FedPETuning* [53] conducts parameter-efficient federated fine-tuning on homogeneous models.
- *FAH-QLoRA* [16] enables heterogeneous federated LoRA aggregation via utilizing *zero-padding* and *truncation*.
- *FLoRA* [39] *stacks* local LoRA matrices for heterogeneous aggregation, enabling rank-flexible global updates.
- *FFA-LoRA* [32] applies DP to FedLoRA by only *freezing non-zero matrices* and tuning zero-initialized ones.

4.2 End-to-End Performance

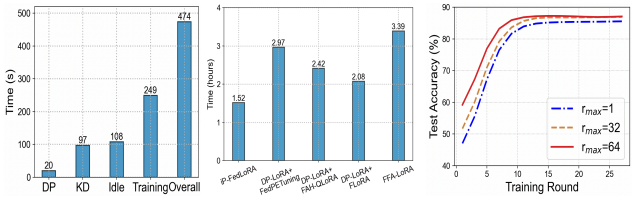
iP-FedLoRA improves model accuracy. Table 5 summarizes the accuracy of iP-FedLoRA and baselines. DP-LoRA+FedPETuning shows lower accuracy due to homogeneous LoRA aggregation. In contrast, DP-LoRA+FAH-QLoRA and DP-LoRA+FLoRA improve accuracy up to 1.2% and 2.2%, respectively, through assigning various LoRA rank modules. FFA-LoRA neglects the variance of noise

Table 5: Accuracy (%) or F1 score (%) of iP-FedLoRA and the baselines on different datasets and LLMs.

Method	RoBERTa-base			DeBERTa-v3-base			Llama-3.2-3B	
	MRPC	SST-2	QNLI	MRPC	SST-2	QNLI	QQP	MNLI
DP-LoRA+FedPETuning	$85.9_{\pm 1.02}$	$92.0_{\pm 0.42}$	$84.2_{\pm 1.22}$	$85.5_{\pm 1.52}$	$92.5_{\pm 0.27}$	$86.5_{\pm 0.58}$	$85.1_{\pm 0.18}$	$85.1_{\pm 0.45}$
DP-LoRA+FAH-QLoRA	$85.7_{\pm 0.40}$	$92.2_{\pm 0.12}$	$85.2_{\pm 1.72}$	$85.9_{\pm 0.85}$	$92.9_{\pm 0.95}$	$86.8_{\pm 0.56}$	$86.3_{\pm 0.24}$	$86.1_{\pm 0.27}$
DP-LoRA+FLoRA	$86.2_{\pm 0.82}$	$92.3_{\pm 1.01}$	$85.9_{\pm 0.21}$	$86.5_{\pm 0.97}$	$93.0_{\pm 0.58}$	$87.3_{\pm 1.24}$	$87.3_{\pm 0.22}$	$87.2_{\pm 0.83}$
FFA-LoRA	$85.3_{\pm 1.00}$	$92.3_{\pm 1.37}$	$85.2_{\pm 0.32}$	$85.8_{\pm 1.24}$	$92.7_{\pm 0.15}$	$87.0_{\pm 0.43}$	$86.4_{\pm 0.66}$	$85.8_{\pm 0.40}$
iP-FedLoRA	**87.1**$_{\uparrow 1.8}$	**93.3**$_{\uparrow 1.3}$	**87.5**$_{\uparrow 3.3}$	**87.0**$_{\uparrow 1.5}$	**93.5**$_{\uparrow 1.0}$	**89.0**$_{\uparrow 2.5}$	**88.9**$_{\uparrow 3.8}$	**87.8**$_{\uparrow 2.7}$

Table 6: Ablation study on components of iP-FedLoRA.

DP	Data Hetero.	Model Hetero.	Acc.(%)
DP-SGD	✗	✗	84.8
matrix-wise DP	✗	✗	86.5
matrix-wise DP	**LoRA regularization**	truncation	86.6
matrix-wise DP	freeze non-zero matrices	**noise-resilient KD**	87.2
matrix-wise DP	**LoRA regularization**	**noise-resilient KD**	**87.5**



(a) Time breakdown. (b) Overall time cost. Figure 5: Resource cost versus time.

Figure 6: Impact of maximum rank.

Table 7: Accuracy comparison of standalone regularization and heterogeneous aggregation modules with standard DP.

Method	QNLI	QQP	Method	QNLI	QQP
FedProx	84.5	85.8	FAH-QLoRA	85.0	86.1
FedDyn	84.7	86.1	FLoRA	85.9	87.2
LoRA regularization	**85.3**	**86.9**	**noise-resilient KD**	**86.8**	**87.8**

Table 8: Comparison of computational and communication costs.

Method	Comp.(s)	Acc.(%)	Method	Comm.(MB)
DP-SGD	18.3	84.2	FLoRA	1.18
matrix-wise DP	**20.5**$_{\uparrow 2.2}$	**87.5**$_{\uparrow 3.3}$	**noise-resilient KD**	**0.08**$_{\downarrow 14.7\times}$

impact on different LoRA modules and struggles with model heterogeneity. iP-FedLoRA achieves a 1.3% – 3.8% accuracy gain by combining matrix-wise adaptive DP and addressing client heterogeneity in data distribution and models.

iP-FedLoRA expedites convergence. A key concern with iP-FedLoRA is whether its accuracy improvement compromises model convergence. Fig. 4 presents model accuracy versus training rounds. iP-FedLoRA converges 1.8× and 2.5× faster than baselines on RoBERTa-base and Llama-3.2-3B.

4.3 Ablation Study

Component-wise evaluation of iP-FedLoRA. To evaluate the effectiveness of each key component in iP-FedLoRA, we conduct an ablation study using RoBERTa-base on QNLI dataset. As shown in Table 6, replacing DP-SGD with our matrix-wise DP alone improves accuracy by 1.7%. Further gains arise from handling heterogeneity: combining matrix-wise DP with "freeze non-zero matrices" (FFA-LoRA) and our noise-resilient heterogeneous LoRA aggregation yields 87.2%, while our LoRA regularization paired with noise-resilient aggregation achieves the best (+2.7% over DP-SGD).

Comparison of regularization methods. Table 7 shows that our rank-compensated LoRA regularization outperforms FedProx [23] and FedDyn [2] by 0.8%-1.1%. This is because it adapts to LoRA rank heterogeneity and prevents overfitting for low-rank clients while preserving informative updates for high-rank ones.

Comparison of heterogeneous LoRA aggregation methods. From Table 7, compared to heterogeneous federated LoRA aggregation baselines, our noise-resilient aggregation improves accuracy by 0.6%-1.8%. This gain is achieved by effectively filtering DP-polluted knowledge and learning high-quality knowledge across clients.

4.4 Resource Cost

Breakdown of iteration time in iP-FedLoRA. Fig. 5(a) shows the iteration time of iP-FedLoRA on RoBERTa-base, where "idle" is sync waiting. Matrix-wise DP adds only 4.2% overhead, while noise-resilient heterogeneous aggregation takes 20.5% of the total time, which remains acceptable.

Overall time comparison with baselines. Fig. 5(b) presents the comparison results of overall time consumption. Compared to the baselines, iP-FedLoRA speeds up training to 1.37 – 2.23× in terms of wall-clock time. This improvement is attributed to the *far fewer convergence rounds* than baselines.

Comparison of single component. We analyze *per-round cost* to verify the practicality in real deployments in Table 8. Matrix-wise DP increases computation by only 2.2s but yields 3.3% accuracy gains over DP-SGD, and noise-resilient KD cuts communication cost by 14.7× compared with FLoRA, showing its efficiency.

4.5 Sensitivity Analysis

We conduct a sensitivity analysis for iP-FedLoRA by fine-tuning RoBERTa-base on QNLI dataset.

Impact of maximum LoRA rank. Basically, a higher maximum rank r indicates a larger disparity in clients' resource availabilities and an increased heterogeneity. From Fig. 6, iP-FedLoRA performs better as r increases, achieving a 1.2% accuracy improvement.

Impact of privacy budget. Table 9 presents model accuracy with varying privacy budget ϵ. As privacy protection is enhanced (i.e., smaller ϵ), model accuracy generally declines for both methods, but iP-FedLoRA exhibits less degradation under stricter privacy guarantees, particularly at $\epsilon = 2$, showing the effectiveness of our matrix-aware DP method.

Table 9: Impact of privacy budget ϵ with non-IIDness α.

Method	non-IID	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$
DP-LoRA +FedPETuning	$\alpha = 1$	$83.1_{\pm 0.31}$	$84.0_{\pm 0.11}$	$85.0_{\pm 0.95}$
	$\alpha = 5$	$83.5_{\pm 0.19}$	$84.7_{\pm 0.82}$	$86.0_{\pm 0.54}$
	$\alpha = 10$	$83.9_{\pm 0.21}$	$85.2_{\pm 0.14}$	$86.8_{\pm 0.10}$
iP-FedLoRA	$\alpha = 1$	**86.1**$_{\uparrow 3.0}$	**86.5**$_{\uparrow 2.5}$	**86.7**$_{\uparrow 1.7}$
	$\alpha = 5$	**86.7**$_{\uparrow 3.2}$	**87.0**$_{\uparrow 3.0}$	**87.8**$_{\uparrow 1.8}$
	$\alpha = 10$	**87.0**$_{\uparrow 3.1}$	**87.9**$_{\uparrow 2.7}$	**88.4**$_{\uparrow 1.6}$

Table 10: Membership inference attack performance vs ϵ.

Method	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = \infty$
Loss Attack	0.504	0.508	0.524	0.582
Neighbour Attack	0.518	0.522	0.564	0.641
LiRA	0.525	0.535	0.626	0.755

Impact of data non-IIDness. Typically, a smaller α suggests more heterogeneous data. Table 9 shows that under the same privacy budget ϵ, iP-FedLoRA improves model accuracy up to 3.2% compared to baseline across varying data non-IIDness.

4.6 Robustness Analysis

Resistance to membership inference attacks. We assess iP-FedLoRA's resistance to three common membership inference attacks, i.e., Loss Attack [48], Neighbor Attack [28], LiRA [7], on the Llama-3.2-3B model for QQP task. Table 10 measures the attack success rate via AUC across varying privacy budgets. With strong DP protection (e.g., $\epsilon = 4$), all attacks show near-random guess performance (AUC≈0.5), while removing DP ($\epsilon = \infty$) significantly enhances attack efficacy, with LiRA achieving the highest AUC of 0.755. These results validate our robustness against privacy attacks.

5 Related Work

Federated LoRA fine-tuning. LoRA has been recognized as one of the most representative PEFT methods for LLMs [21]. With increasing privacy concerns, FL [29] has emerged as a solution that enables clients to jointly fine-tune LLMs without exposing private data. Zhang *et al.* explore PEFT methods for FL [53]. Xu *et al.* develop an efficient federated fine-tuning method using perturbed inferences [43]. However, they still suffer from gradient leakage and heterogeneity challenges. iP-FedLoRA complements them with matrix-wise DP fine-tuning and noise-resilient aggregation.

Differentially private fine-tuning. DP is commonly adopted as a rigorous privacy notion for measuring privacy risk [13]. Yu *et al.* present DP-LoRA, a centralized private fine-tuning scheme by applying DP-SGD to LoRA [49]. Sun *et al.* develop FFA-LoRA to improve LoRA in privacy-preserving FL [32]. Wang *et al.* propose a relevance-based adaptive differentially private fine-tuning framework to balance privacy and utility [37]. However, these solutions overlook the structural asymmetry inherent in LoRA matrices. Applying adaptive DP at the coarse-grained layer level forces structurally disparate LoRA matrices to share clipping thresholds, resulting in suboptimal noise allocation.

Heterogeneity in federated fine-tuning. Federated fine-tuning faces unique challenges posed by client heterogeneity [5]. Babakniya *et al.* propose SLoRA to address the limitations of LoRA in heterogeneous data [4]. Chen *et al.* design a federated dual-adapter teacher framework for heterogeneous multi-modal FL [8]. Gao *et al.* propose FAH-QLoRA for dynamic LoRA rank adaptation and quantization, using zero-padding and truncation to enable heterogeneous aggregation [16]. However, these methods often suffer from accuracy loss and lack of flexibility. In contrast, iP-FedLoRA develops a noise-resilient KD scheme that selectively transfers high-confidence knowledge, thereby mitigating heterogeneity issue without propagating DP noise during distillation.

6 Conclusion

In this paper, we present iP-FedLoRA, an improved private federated fine-tuning framework for LLMs on heterogeneous clients. To achieve better privacy-utility balance, iP-FedLoRA employs matrix-wise differentially private local fine-tuning with rank-compensated regularization, while at the same time developing noise-resilient federated LoRA aggregation for robust knowledge transfer among model-heterogeneous clients. Extensive evaluations corroborate the effectiveness and superiority of iP-FedLoRA over baselines in model accuracy and convergence.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China under grant 62302343, the Key R&D Program of Hubei Province under grant No. 2024BAB018, the Hubei Provincial Science and Technology Innovation Base (Platform) Program Project under grant No. 2025CSA112, and the NSFC/RGC CRS Grants (Ref. #CRS_HKUST601/24 and Ref. #CRS_PolyU501/23).

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proc. ACM CCS*. 308–318.
[2] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. 2021. Federated learning based on dynamic regularization. In *Proc. ICLR*.
[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, and Ilge Akkaya et al. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).
[4] Sara Babakniya, Ahmed Elkordy, Yahya Ezzeldin, Qingfeng Liu, Kee-Bong Song, MOSTAFA EL-Khamy, and Salman Avestimehr. 2023. SLoRA: Federated Parameter Efficient Fine-Tuning of Language Models. In *Proc. NeurIPS*.
[5] Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. 2024. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In *Proc. NeurIPS*. 14457–14483.
[6] James C Bezdek, Robert Ehrlich, and William Full. 1984. FCM: The fuzzy c-means clustering algorithm. *Computers & geosciences* (1984), 191–203.
[7] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. In *Proc. IEEE S&P*. 1897–1914.
[8] Haokun Chen, Yao Zhang, Denis Krompass, Jindong Gu, and Volker Tresp. 2024. FedDAT: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proc. AAAI*. 11285–11293.
[9] Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, and Gauri Joshi. 2024. Heterogeneous LoRA for federated fine-tuning of on-device foundation models. In *Proc. EMNLP*. 12903–12913.
[10] Yongheng Deng, Ju Ren, Cheng Tang, Feng Lyu, Yang Liu, and Yaoxue Zhang. 2023. A hierarchical knowledge transfer framework for heterogeneous federated learning. In *Proc. IEEE INFOCOM*. 1–10.
[11] Minxin Du, Xiang Yue, Sherman SM Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. 2023. DP-Forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proc. ACM CCS*. 2665–2679.
[12] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).
[13] Cynthia Dwork. 2006. Differential privacy. In *Proc. ICALP*. 1–12.
[14] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3–4 (2014),

211–407.
[15] Ying Gao, Yuxin Xie, Huanghao Deng, and Zukun Zhu. 2025. Gradient Inversion Attack in Federated Learning: Exposing Text Data through Discrete Optimization. In *Proc. COLING*. 2582–2591.
[16] Zhidong Gao, Zhenxiao Zhang, Yuanxiong Guo, and Yanmin Gong. 2025. Federated Adaptive Fine-Tuning of Large Language Models with Heterogeneous Quantization and LoRA. In *Proc. IEEE INFOCOM*. 1–10.
[17] Yongchang Hao, Yanshuai Cao, and Lili Mou. 2024. Flora: Low-Rank Adapters Are Secretly Gradient Compressors. In *Proc. ICML*.
[18] Jiyan He, Xuechen Li, Da Yu, Huishuai Zhang, Janardhan Kulkarni, Yin Tat Lee, Arturs Backurs, Nenghai Yu, and Jiang Bian. 2023. Exploring the limits of differentially private deep learning with group-wise clipping. In *Proc. ICLR*.
[19] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *Proc. ICLR*.
[20] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335* (2019).
[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proc. ICLR*.
[22] Daliang Li and Junpu Wang. 2019. FedMD: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581* (2019).
[23] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. In *Proc. MLSys*. 429–450.
[24] Yuqing Li, Nan Yan, Jing Chen, Xiong Wang, Jianan Hong, Kun He, Wei Wang, and Bo Li. 2025. FedPHE: A Secure and Efficient Federated Learning via Packed Homomorphic Encryption. *IEEE TDSC* 22, 5 (2025), 5448–5463.
[25] 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* (2019).
[26] Jiating Ma, Yipeng Zhou, Qi Li, Quan Z Sheng, Laizhong Cui, and Jiangchuan Liu. 2025. The power of bias: Optimizing client selection in federated learning with heterogeneous differential privacy. *IEEE TDSC* (2025).
[27] Xinge Ma, Jin Wang, and Xuejie Zhang. 2026. LLM-Driven Synthetic Text Generation for Privacy-Preserving Federated Learning. *IEEE Transactions on Audio, Speech and Language Processing* 34 (2026), 921–935.
[28] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, et al. 2023. Membership inference attacks against language models via neighbourhood comparison. In *Proc. ACL*. 11330–11343.
[29] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proc. AISTATS*. 1273–1282.
[30] Meta. 2024. Llama-3.2-3B-Instruct. https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct.
[31] Ilya Mironov. 2017. Rényi differential privacy. In *Proc. IEEE CSF*. 263–275.
[32] Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. 2024. Improving LoRA in Privacy-preserving Federated Learning. In *Proc. ICLR*.
[33] Anvith Thudi, Hengrui Jia, Casey Meehan, Ilia Shumailov, and Nicolas Papernot. 2024. Gradients look alike: Sensitivity is often overestimated in DP-SGD. In *Proc. USENIX Security*. 973–990.
[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, et al. 2017. Attention is all you need. In *Proc. NeurIPS*.
[35] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proc. ICLR*.
[36] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. 2020. Optimizing federated learning on non-iid data with reinforcement learning. In *Proc. IEEE INFOCOM*. 1698–1707.
[37] Naiyu Wang, Shen Wang, Meng Li, Longfei Wu, Zijian Zhang, Zhitao Guan, and Liehuang Zhu. 2025. Balancing Differential Privacy and Utility: A Relevance-Based Adaptive Private Fine-Tuning Framework for Language Models. *IEEE TIFS* 20 (2025), 207–220.
[38] Xiong Wang, Yuxin Chen, Yuqing Li, Xiaofei Liao, Hai Jin, and Bo Li. 2023. FedMoS: Taming Client Drift in Federated Learning with Double Momentum and Adaptive Selection. In *Proc. IEEE INFOCOM*. 791–800.
[39] 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. In *Proc. NeurIPS*.
[40] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *Proc. IEEE INFOCOM*. 2512–2520.
[41] Duo Wu, Xianda Wang, Yaqi Qiao, Zhi Wang, Junchen Jiang, Shuguang Cui, and Fangxin Wang. 2024. NetLLM: Adapting Large Language Models for Networking. In *Proc. ACM SIGCOMM*. 661–678.
[42] Fei Wu, Jia Hu, Geyong Min, and Shiqiang Wang. 2025. Adaptive rank allocation for federated parameter-efficient fine-tuning of language models. *IEEE Trans. Comput.* 75 (2025), 1650–1663.
[43] Mengwei Xu, Dongqi Cai, Yaozong Wu, Xiang Li, and Shangguang Wang. 2024. FwdLLM: Efficient Federated Finetuning of Large Language Models with Perturbed Inferences. In *Proc. USENIX ATC*. 579–596.
[44] Zhiying Xu, Shuyu Shi, Alex X Liu, Jun Zhao, and Lin Chen. 2020. An adaptive and fast convergent approach to differentially private deep learning. In *Proc. IEEE INFOCOM*. 1867–1876.
[45] Rui Xue, Kaiping Xue, Bin Zhu, Xinyi Luo, Tianwei Zhang, Qibin Sun, and Jun Lu. 2024. Differentially private federated learning with an adaptive noise mechanism. *IEEE TIFS* 19 (2024), 74–87.
[46] Nan Yan, Yuqing Li, Jing Chen, Xiong Wang, Jianan Hong, Kun He, and Wei Wang. 2024. Efficient and Straggler-Resistant Homomorphic Encryption for Heterogeneous Federated Learning. In *Proc. IEEE INFOCOM*. 791–800.
[47] Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. 2023. PrivateFL: Accurate, differentially private federated learning via personalized data transformation. In *Proc. USENIX Security*. 1595–1612.
[48] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Proc. CSF*. 268–282.
[49] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. 2022. Differentially private fine-tuning of language models. In *Proc. ICLR*.
[50] Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *Proc. ICML*.
[51] Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Pan, and Lidong Bing. 2024. Sentiment Analysis in the Era of Large Language Models: A Reality Check. In *Proc. NAACL*. 3881–3906.
[52] Xinwei Zhang, Xiangyi Chen, Mingyi Hong, Zhiwei Steven Wu, and Jinfeng Yi. 2022. Understanding clipping for federated learning: Convergence and client-level differential privacy. In *Proc. ICML*.
[53] 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 *Proc. ACL*. 9963–9977.
[54] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. 2021. Data-free knowledge distillation for heterogeneous federated learning. In *Proc. ICML*. 12878–12889.

A Resource-Aware Client Clustering

The server groups clients into clusters based on *resource availability* rather than private data attributes. Define the *collective resource feature* of client i as $\boldsymbol{\chi}_i = [\chi_{i,1}, \chi_{i,2}, \chi_{i,3}, \chi_{i,4}]$, which captures the data volume, computing power, memory capacity and communication capacity, respectively. To address the feature scale problem, *Min-Max* normalization is performed on each feature such that $\widetilde{\chi}_{i,k} = \frac{\chi_{i,k} - \min_j \chi_{j,k}}{\max_j \chi_{j,k} - \min_j \chi_{j,k}}$, where $\chi_{i,k}$ is the k-th feature of χ_i. The server randomly selects C resource features from all clients to serve as initial *clustering centers* $v_1, v_2, \cdots, v_C$. For each client i, its *fuzzy membership* [6] for every cluster center v_c is calculated as $u_{i,c} = 1/\sum_{j=1}^{C} \left[\frac{d(\boldsymbol{\chi_i}, \boldsymbol{v_c})}{d(\boldsymbol{\chi_i}, \boldsymbol{v_j})}\right]^{\frac{2}{m-1}}$, which reflects the probability of i being assigned to cluster $\mathcal{A}_c$. Here, $d(\boldsymbol{\chi_i}, v_c) = \sqrt{\sum_k w_k (\widetilde{\chi}_{i,k} - v_{j,k})^2}$, $m > 1$ is the fuzzy factor and w_k is the weight of the k-th resource feature. The client is then added to the *highest-membership* cluster. For each cluster $\mathcal{A}_c$, update the cluster center v_j as $\frac{1}{|\mathcal{A}_c|} \sum_{i \in \mathcal{A}_c} \boldsymbol{\chi_i}$. This process is repeated until all cluster centers no longer change. Finally, iP-FedLoRA computes a *resource score* $Q_c = \sum_k w_k v_{c,k}$ for each cluster and customizes a unique LoRA module for it. In particular, a client with a higher score will be assigned to a larger rank r_c, thus achieving flexible LoRA rank adaptation.

B Proofs of Theorems

B.1 Proof for Theorem 1

Thanks to the *post-processing* property of DP [14], it suffices to show that accurate matrix-wise DP local fine-tuning satisfies (ϵ, δ)-DP in our iP-FedLoRA, since the subsequent modules like LoRA

regulation and heterogeneous federated LoRA aggregation incur no extra privacy loss. To establish this privacy guarantee, we leverage the techniques of *Rényi differential privacy* (RDP) to analyze the privacy cost of the *composition* of Gaussian noise mechanisms for all M LoRA matrices. The results that we obtain via RDP are later converted to DP. We first introduce several key definitions and lemmas as follows.

Definition 2 (**Rényi Divergence** [31]). *Given two probability distributions P and Q, the Rényi divergence between P and Q with $\beta > 1$ is defined by $D_\beta(P||Q) \triangleq \frac{1}{\beta-1} log\, \mathbb{E}_{x\sim Q}\left(\frac{P(x)}{Q(x)}\right)^\beta$.*

Definition 3 (**RDP** [31]). *A randomized mechanism $\mathcal{M}: D \leftarrow \mathbb{R}^d$ is (β,ϵ)-RDP if for any two neighboring datasets $\mathcal{D}, \mathcal{D}'$, it holds that $D_\beta(\mathcal{M}(\mathcal{D})||\mathcal{M}(\mathcal{D}'))\le\epsilon$.*

Lemma 1 (**RDP of Gaussian Mechanism** [31]). *If the sensitivity of a function f meets $H_f = 1$, then the Gaussian Mechanism $\mathcal{G}_{\sigma_*}(f) = f(\mathcal{D}) + \mathcal{N}(0,\sigma_*^2)$ satisfies $\left(\beta, \frac{\beta}{2\sigma_*^2}\right)$-RDP.*

Lemma 2 (**From RDP to DP** [31]). *If a randomized mechanism $\mathcal{M}$ is (β,ϵ)-RDP, it satisfies $\left(\epsilon + \frac{log(1/\delta)}{\beta-1}, \delta\right)$-DP for any $\delta \in (0,1)$.*

Lemma 3 (**Sequential Composition of RDP** [31]). *Suppose $\mathcal{M}_1 : \mathcal{D} \to \mathcal{R}_1$ satisfy (β,ϵ_1)-RDP and $\mathcal{M}_2 : \mathcal{R}_1 \times \mathcal{D} \to \mathcal{R}_2$ satisfy (β,ϵ_2)-RDP, then the composition mechanism defined as $\mathcal{M} := (\mathcal{M}_1(\mathcal{D}), \mathcal{M}_2(\mathcal{D}))$, satisfies $(\beta, \epsilon_1+\epsilon_2)$-RDP.*

Lemma 4 (**Parallel Composition of RDP** [31]). *Given a collection of disjoint subsets of the data $\mathcal{D}_i \in \mathcal{D}$, for any $i = 1,\cdots,N$. Let mechanism $\mathcal{M}$ satisfies (β,ϵ)-RDP, then the combination of these mechanisms $(\mathcal{M}(\mathcal{D}_1), \mathcal{M}(\mathcal{D}_2),\cdots,\mathcal{M}(\mathcal{D}_N))$ satisfies (β,ϵ)-RDP.*

Given these lemmas, we then formally present the proof.

Proof. Although the clipping threshold relies on historical data, the privacy accounting remains rigorous under adaptive composition for RDP. Since the noise scale dynamically follows the sensitivity, the data-dependent term cancels exactly in the RDP calculation, yielding a constant privacy cost.

Conditional sensitivity. Let h^{t-1} denote the history of all DP-protected updates prior to round t. By Eq. (4), $s^t_{i,j,m}$ is solely post-processed from h^{t-1}. For any fixed h^{t-1}, $s^t_{i,j,m}$ is a deterministic scalar independent of the current dataset $\mathcal{D}$. According to Eq. (5), the conditional ℓ_2-sensitivity is bounded as $H^t_{i,j,m} \mid h^{t-1} = \max_{\mathcal{D}\sim\mathcal{D}'} \|\text{clip}(\Delta\boldsymbol{\omega}^t_{i,j,m}(\mathcal{D}), s^t_{i,j,m}) - \text{clip}(\Delta\boldsymbol{\omega}^t_{i,j,m}(\mathcal{D}'), s^t_{i,j,m})\|_2 \le 2s^t_{i,j,m}$.

Conditional RDP per matrix. Given h^{t-1}, the noise scale is $\sigma^t_{i,j,m} = \sqrt{M}\sigma_* H^t_{i,j,m} = 2\sqrt{M}\sigma_* s^t_{i,j,m}$. By Lemma 1, the conditional RDP cost for matrix m in layer j is

$$\epsilon_{t,j,m}(\beta) \mid h^{t-1} \le \frac{\beta(H^t_{i,j,m} \mid h^{t-1})^2}{2(\sigma^t_{i,j,m})^2} = \frac{\beta(2s^t_{i,j,m})^2}{2\left(2\sqrt{M}\sigma_* s^t_{i,j,m}\right)^2} = \frac{\beta}{2M\sigma_*^2}.$$

Notice that $s^t_{i,j,m}$ cancels exactly, making the conditional RDP cost a constant independent of h^{t-1} and the current dataset.

Conditional RDP per round. Applying Lemma 3 across all M matrices, the total conditional cost for round t is constant, i.e.,

$$\epsilon_t(\beta) \mid h^{t-1} \le \sum_{j,m} \frac{\beta}{2M\sigma_*^2} = \frac{\beta}{2\sigma_*^2}.$$

Adaptive composition and DP conversion. As the per-round RDP cost is constant for any history h^{t-1}, adaptive composition [31] safely allows sequential summation. For I_i participation rounds with sampling ratio q, the total privacy guarantee for client i is $\left(\beta, \frac{I_i q\beta}{2\sigma_*^2}\right)$-RDP. Applying Lemma 2 yields

$$\frac{I_i q\beta}{2\sigma_*^2} + \frac{log(1/\delta)}{\beta-1} \le \epsilon,$$

which is exactly the condition in Theorem 1. □

B.2 Proof for Theorem 2

Proof. According to Assumption 1 and Eq. (5), it holds that

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\boldsymbol{W}^{t+1})] &\le \mathcal{L}(\boldsymbol{W}^t) + \langle\nabla\mathcal{L}(\boldsymbol{W}^t), \boldsymbol{W}^{t+1}-\boldsymbol{W}^t\rangle + \frac{L}{2}\|\boldsymbol{W}^{t+1}-\boldsymbol{W}^t\|^2 \\ &\le \mathcal{L}(\boldsymbol{W}^t) + \underbrace{\left\langle\nabla\mathcal{L}(\boldsymbol{W}^t), \mathbb{E}\left[\frac{1}{N}\sum_{i\in\boldsymbol{N}}\Delta\widetilde{\boldsymbol{\omega}}^t_i\right]\right\rangle}_{T_1} + \underbrace{\frac{L}{2}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i\in\boldsymbol{N}}\Delta\widetilde{\boldsymbol{\omega}}^t_i\right\|^2\right]}_{T_2} + \frac{LM\sigma_*^2}{2N}.\end{aligned} \tag{8}$$

We simplify the term $T_1 \le \eta\left\langle\nabla\mathcal{L}(\boldsymbol{W}^t), \mathbb{E}\left[\nabla\mathcal{L}(\boldsymbol{W})\right]\right\rangle$, and bound the term $T_2 \le \frac{LMK\eta^2G^2}{2}$ with Assumption 2. Taking the sum of t from 1 to T in Eq. (8), we can derive that

$$\begin{aligned}&\frac{\mathbb{E}\left[\mathcal{L}(\boldsymbol{W}^{T+1})\right] - \mathbb{E}\left[\mathcal{L}(\boldsymbol{W}^1)\right]}{\eta T} \\ &\le \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\langle\nabla\mathcal{L}(\boldsymbol{W}^t), \mathbb{E}[\nabla\mathcal{L}(\boldsymbol{W})]\rangle\right] + \frac{L\eta MKG^2}{2} + \frac{LM\sigma_*^2}{2\eta N}.\end{aligned} \tag{9}$$

Given that $\langle a,b\rangle = -\frac{1}{2}\|a\|^2 - \frac{1}{2}\|b\|^2 + \frac{1}{2}\|a-b\|^2, \forall a,b$, we convert the first part of Eq. (9) into $-\frac{K}{2T}\sum_{t=1}^T\mathbb{E}\left[\left\|\nabla\mathcal{L}(\boldsymbol{W}^t)\right\|^2\right] - \frac{G^2}{2KT} + \frac{1}{2T}\sum_{t=1}^T\mathbb{E}\left[\|\sqrt{K}\nabla\mathcal{L}(\boldsymbol{W}^t) - \frac{1}{\sqrt{K}}\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{W})]\|^2\right]$. Inspired by [52], we then scale the last term for a single round t, we have

$$\begin{aligned}&\frac{1}{2T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\sqrt{K}\nabla\mathcal{L}(\boldsymbol{W}^t) - \frac{1}{\sqrt{K}}\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{W})]\right\|^2\right] \\ &\le \frac{1}{2TN}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla l(\boldsymbol{W}_i^t) - \nabla l(\boldsymbol{W}_i^{t,k})\right\|^2\right] \\ &\le \frac{L^2K}{2T}\left(5K\eta^2(\sigma_l^2 + 6K\sigma_g^2) + 30K^2\eta^2\left\|\nabla\mathcal{L}(\boldsymbol{W}^t)\right\|^2\right),\end{aligned} \tag{10}$$

where the first inequality comes from Jensen's inequality, the second inequality comes from Assumption 3.

When $\eta \le \frac{1}{\sqrt{60}LK}$, combining Eq. (9) and Eq. (10) yields

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\boldsymbol{W}^t)\right\|^2\right] &\le \frac{4\left(\mathbb{E}\left[\mathcal{L}(\boldsymbol{W}^1)\right] - \mathbb{E}[\mathcal{L}(\boldsymbol{W}^{T+1})]\right)}{\eta KT} \\ &\quad + 2G^2(L\eta M - 1) + 10L^2K\eta^2(\sigma_l^2 + 6K\sigma_g^2) + \frac{2LM\sigma_*^2}{\eta KN} \\ &= O\left(\frac{1}{\eta KT} + \eta M + \eta^2K + \eta^2K^2 + \frac{Mq\beta r}{2\left(\epsilon - \frac{log(\frac{1}{\delta})}{\beta-1}\right)\eta KN}\right)\end{aligned}$$

Applying $\beta = 1 + 2\frac{log(1/\delta)}{\epsilon}$, we have $\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^T\|\nabla\mathcal{L}(\boldsymbol{W}^t)\|^2\right] \le O\left(\frac{1}{\eta KT} + \eta M + \eta^2K + \eta^2K^2 + \frac{Mqr(\epsilon+2log(\frac{1}{\delta}))}{\epsilon^2\eta KN}\right)$, which completes the proof. □

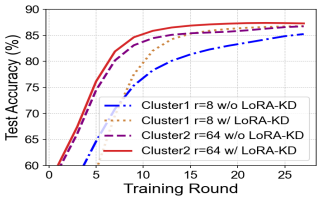


Figure 7: Ablation study of InterLoRA.

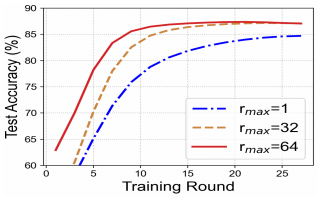


Figure 8: Impact of maximum LoRA rank.

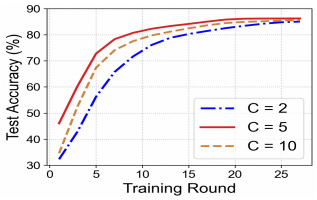


Figure 9: Impact of cluster number.

Table 11: Impact of generator selection on QNLI.

Generator	Acc.(%)
GPT-4o	87.5
Llama-3-8B-Instruct	87.5
Qwen3-1.7B	87.2

C Public Dataset Generation

Unlike prior works that rely on existing public datasets [10], which are often unavailable or suffer from distribution shifts, iP-FedLoRA constructs the public dataset $\widehat{\mathcal{D}}$ via generative synthesis. The server leverages GPT-4o to generate diverse, unlabeled samples based on task-specific prompts. For instance, for the SST-2 sentiment analysis task, we use the prompt: "Generate a diverse list of 100 sentences that express either positive or negative opinions about movies, similar to Rotten Tomatoes reviews." We set the generation temperature to 1.0 to ensure high diversity and generate 20% samples of each task as the public dataset. This process is not only cost-effective but also ensures that the distillation process is driven by the semantic-related task rather than by a specific external dataset.

We further clarify the robustness of this public dataset generation strategy to different generators. Our confidence-based denoising in Sec. 3.3 filters out low-quality or ambiguous pseudo-labels, reducing the impact of generator capability. As shown in Table 11, QNLI accuracy remains stable even with a compact 1.7B generator, suggesting that smaller open-source LLMs are sufficient for synthetic data generation.

D Privacy Accounting

To ensure strict (ϵ, δ)-DP guarantees, we calibrate the noise parameters to the privacy budget. Specifically, given a target budget ϵ (eg., $\epsilon = 4$) and the total number of communication rounds, we first numerically derive the required base noise scale σ_* using the standard RDP accountant [31]. Following Theorem 1, the noise scale for each LoRA matrix m of client i in layer j is then calculated as $\sigma_{i,j,m} = 2\sqrt{M}\sigma_* s_{i,j,m}$, where M denotes the total number of matrices and $s_{i,j,m}$ is the adaptive clipping threshold dynamically updated via Eq. (4). This ensures that the noise magnitude for each LoRA matrix scales automatically with the adaptive clipping threshold, maintaining the constant signal-to-noise ratio required by σ_* to satisfy the privacy constraint.

E Results for Large-Scale iP-FedLoRA

To verify the scalability of iP-FedLoRA, we further extend experiments to 50 clients using RoBERTa-base on QNLI, where the resource-aware client clustering scheme is employed with client selection ratio of 0.1.

Table 12: Accuracy (%) comparison of large-scale deployments on different datasets.

Method	SST-2	QNLI	QQP
DP-LoRA+FedPETuning	$91.5_{\pm 0.31}$	$83.6_{\pm 1.22}$	$84.9_{\pm 1.02}$
DP-LoRA+FAH-QLoRA	$91.8_{\pm 0.45}$	$84.7_{\pm 2.21}$	$85.0_{\pm 0.82}$
DP-LoRA+FLoRA	$92.0_{\pm 0.12}$	$85.2_{\pm 0.74}$	$85.8_{\pm 0.51}$
FFA-LoRA	$91.8_{\pm 0.58}$	$84.5_{\pm 0.32}$	$85.3_{\pm 1.00}$
iP-FedLoRA	**92.8**$_{\uparrow 1.3}$	**86.5**$_{\uparrow 2.9}$	**87.4**$_{\uparrow 2.5}$

Table 13: Inter-cluster heterogeneous LoRA aggregation.

Methods	SST-2	QNLI	QQP
HETLoRA	$92.3_{\pm 0.08}$	$85.2_{\pm 0.51}$	$86.1_{\pm 0.42}$
InterLoRA	**92.5**$_{\uparrow 0.2}$	**86.8**$_{\uparrow 1.6}$	**87.8**$_{\uparrow 1.7}$

E.1 End-to-End Performance

We report the model performance when fine-tuning RoBERTa-base on SST-2 and QNLI, and Llama-3.2-3B on QQP, respectively. From Table 12, we observe that iP-FedLoRA still outperforms the baselines in large-scale deployments with an accuracy improvement by 1.3% – 2.9%. Table 13 shows that our inter-cluster heterogeneous aggregation (InterLoRA) significantly outperforms HETLoRA, increasing model accuracy by 0.2%-1.7%. This is because HETLoRA is unstable under truncating and padding techniques, resulting in performance loss.

E.2 Ablation Study

We explore the necessity of InterLoRA from the perspective of clusters by fine-tuning RoBERTa-base on QNLI. As seen in Fig. 7, without InterLoRA, high-rank clusters attain higher accuracy, while with InterLoRA, lower-rank clusters exhibit greater improvements in both convergence speed and model accuracy. This indicates that knowledge from high-rank clusters can enhance the model performance of low-rank ones.

E.3 Sensitivity Analysis.

We conduct a sensitivity analysis by fine-tuning RoBERTa-base on QNLI. Fig. 8 shows that iP-FedLoRA is robust to the maximum LoRA rank r_{max}, especially for more model-heterogeneous settings (i.e., larger r_{max}). Fig. 9 reports the impact of the number of clusters C. A small C slows convergence, whereas a large C reduces the number of clients per cluster, which degrades knowledge quality and results in suboptimal KD. Choosing $C \approx 5$ provides a good trade-off for preserving model accuracy and convergence speed.