One-Shot Parameter-Efficient Federated Tuning for LLMs via Heterogeneous Knowledge Alignment

Anonymous ACL submission

Abstract

Fine-tuning large language models (LLMs) on decentralized data offers opportunities while also posing challenges, especially concerning data privacy and reducing overhead. Although federated learning (FL) combined with parameter-efficient methods like low-rank adaptation (LoRA) has shown promise, current approaches often necessitate multiple communication rounds to mitigate client drift, resulting in significant communication and computation overhead. To address these challenges, we propose a novel one-shot parameter-efficient federated tuning (**OnePeFT**) framework for LLMs that views global model aggregation as heterogeneous knowledge alignment. In this framework, each client applies LoRA to its local model while training only the adapters on domain-specific data, then uploads the adapters to the server with one-round communication. The server uses a novel SVD-based aggregation for low-rank reparameterization to create a global initialization. The global adapter is refined via distillation with a public task-agnostic dataset, aligning shared semantics across clients to reduce bias and enhance generalization and domain-specific performance. Extensive experiments on LLaMA3-8B and Qwen2-7B show that OnePeFT achieves the state-of-the-art performance while significantly reducing communication overhead up to 20×.

1 Introduction

Recent advances in large language models (LLMs) (Guo et al., 2025; Yang et al., 2024b; Achiam et al., 2023) have demonstrated impressive performance across a wide range of tasks, such as question answering and problem solving. To further adapt LLMs to domain-specific scenarios while preserving data privacy, recent studies (Kuang et al., 2024; Chen et al., 2023; Zhang et al., 2023) have explored fine-tuning LLMs using federated learning (FL) on decentralized data. Given the high resource demands of full-model tuning in this setting,

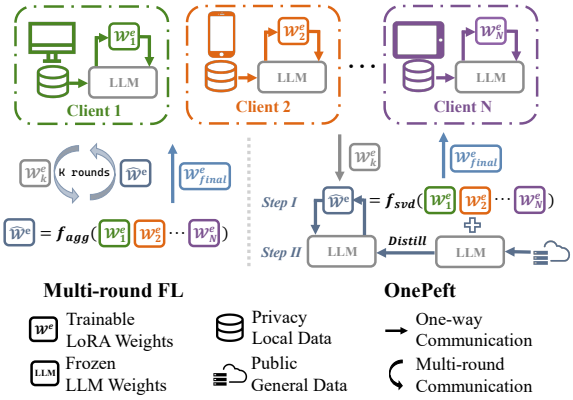


Figure 1: Comparison between multi-round FL-based LLM finetune and OnePeFT. OnePeFT achieves one-shot communication by SVD-based aggregation for initialization, followed by heterogeneous knowledge alignment via distillation on a task-agnostic dataset.

parameter-efficient fine-tuning (PEFT) methods-especially LoRA (Hu et al., 2022)—have become a practical choice for local adaptation. Its low communication cost and relatively low local computational burdens make LoRA particularly appealing for federated LLM tuning.

However, current FL-based LLM tuning methods (Sun et al., 2024; Cho et al., 2024; Zhang et al., 2023) still face a key challenge: **high communication frequency**. Most approaches rely on multiple rounds of communication, as gradual aggregation is needed to mitigate client drift and optimize the global model. Although PEFT methods such as LoRA reduce computational and communication costs per round, the overall overhead remains significant due to the high number of communication rounds. Moreover, frequent communication amplifies privacy risks, exposing the system to attacks such as man-in-the-middle interception (Wang et al., 2020) and gradient-based data reconstruction (Yin et al., 2021).

In response to these challenges, one-shot FL methods (Zhang et al., 2022a,b; Dai et al., 2024) have emerged as a promising solution. By leveraging techniques such as distribution reconstruction,

pseudo-sample generation, and knowledge distillation, they aim to achieve single-round communication for federated model aggregation. These methods have shown feasibility in lightweight models and simple tasks. However, scaling them to LLM fine-tuning remains difficult. Because the high dimensionality of LLM, combined with the complexity of semantic modeling tasks, makes it challenging to generate pseudo data that is both semantically coherent and logically consistent. The knowledge transfer process is then prone to accumulated cognitive bias, ultimately degrading model performance. These limitations make existing one-shot FL approaches ill-suited for fine-tuning LLMs.

To tackle these challenges, we propose a novel one-shot federated LLM fine-tuning framework that views global model aggregation as a process of heterogeneous knowledge alignment, addressing the cognitive bias accumulation problem inherent in existing approaches. Unlike prior one-shot FL methods that rely on generating pseudo samples to approximate client knowledge, we directly align the semantic representations encoded in LoRA adapters using a task-agnostic public dataset. This dataset consists of publicly available, general-purpose text that captures broad linguistic patterns. Since it is entirely decoupled from client data, it does not violate the privacy guarantees of FL.

As shown in Figure 1, in this framework, server cannot access local data and clients collaboratively fine-tune the model without data sharing. Each client incorporates LoRA into its local model, freezing the original LLM parameters and fully training the LoRA adapters on local data to capture domain-specific linguistic patterns. After local training, only the adapters are uploaded to server. The server first performs an *Singular Value Decomposition (SVD)-based Aggregation* to integrate these adapters into a global initialization. While this merges heterogeneous client knowledge, it may cause parameter drift and semantic inconsistency due to differences in local adapter knowledge. To address this, we introduce a *Heterogeneous Knowledge Alignment* stage, where the global LoRA adapter is further refined via distillation using a public, task-agnostic dataset. Instead of generating pseudo samples, we exploit the shared semantics embedded in this general-purpose data to align the global adapter with the diverse knowledge encoded in the client adapters. This process effectively mitigates aggregation bias and ensures semantic coherence, thereby improving both generalization and domain-specific performance.

Based on our experiments, the proposed **One**-shot **P**arameter-**e**fficient **F**ederated **T**uning for LLMs (**OnePeFT**) cuts communication costs by 10×–20× and computation by up to 54.4% versus existing methods. Despite the low cost, OnePeFT maintains competitive performance across all tasks and improves the generalization ability by up to 4.46%, offering a practical FL solution. Our main contributions can be summarized as follows:

- We propose a one-shot federated LLM fine-tuning framework OnePeFT that requires only a single communication round to achieve state-of-the-art performance. To the best of our knowledge, this is the first work of one-shot LLM federated fine-tuning.
- We propose an *SVD-based Aggregation* method and a *Heterogeneous Knowledge Alignment* strategy to integrate and align client knowledge into a global LoRA adapter, improving domain performance and generalization.
- Extensive experiments on LLaMA3-8B and Qwen2-7B demonstrate that our method achieves superior or competitive performance compared to the baseline methods while significantly reducing communication costs by 10×–20×.

2 Related Work

2.1 Parameter Efficient Fine-Tuning

The prohibitive computational cost of direct fine-tuning for ever-growing LLMs has driven the emergence of parameter-efficient fine-tuning (PEFT) methods. Existing approaches fall into two categories: (1) *selective parameter modification* through head tuning (Wei et al., 2021), bias fine-tuning (Bu et al.), or parameter subset optimization (Zaken et al., 2022); and (2) *module augmentation* that introduces lightweight trainable components like adapters (Houlsby et al., 2019), prompts (Lester et al., 2021), prefixes (Li and Liang, 2021), or low-rank matrices (Hu et al., 2022). Among these, LoRA has emerged as one of the most widely adopted PEFT methods. By optimizing a low-rank decomposition of weight updates, LoRA reduces the number of trainable parameters to less than 1% of full fine-tuning while achieving comparable performance.

2.2 One-Shot Federated Learning

One-shot federated learning (OFL) reduces the high communication and privacy costs of tradi-

tional FL by limiting knowledge transfer to a single round. Existing OFL methods fall into three main categories: (1) *Parameter Learning* via clustering (Dennis et al., 2021), layer-wise aggregation (Su et al., 2023), or Fisher-based regularization (Liu et al., 2024b); (2) *Knowledge Distillation* with ensemble-based techniques (Dai et al., 2024; Zhang et al., 2022a); and (3) *Generative Models* using GANs (Kasturi and Hota, 2023), VAEs (Heinbaugh et al., 2023), or diffusion models (Yang et al., 2024c) to synthesize data. However, these methods face challenges when scaling to LLMs: (i) the high dimensionality of LLM parameters amplifies fusion bias during single-round aggregation; (ii) the complexity of semantic modeling makes it difficult to generate pseudo data that is both semantically coherent and logically consistent, resulting in drift.

2.3 PEFT for Federated Learning with LLM

PEFT methods reduce computational and communication costs by freezing most LLM parameters and fine-tuning only a small subset or lightweight modules. Some studies (Sun et al., 2022; Zhang et al., 2023) have conducted a comprehensive empirical study evaluating various representative PEFT methods in terms of performance, privacy preservation, and resource constraints. Given the effectiveness and ease of implementation of LoRA, recent research has increasingly focused on its role in FL. For instance, FedJudge (Yue et al., 2024) explores LoRA-based federated fine-tuning in the Chinese legal domain. SLoRA(Babakniya et al., 2023) and FeDeRA (Yan et al., 2024) explore different initialization strategies for LoRA weights to accelerate model convergence. HETLORA (Cho et al., 2024) introduces rank-adaptive LoRA for heterogeneous clients. FFA-LoRA (Sun et al., 2024) mitigates LoRA aggregation error by freezing the low-rank matrix in LoRA's decomposition. However, these methods typically require many communication rounds to achieve satisfactory performance, limiting their practicality—especially in privacy-sensitive LLM fine-tuning where communication efficiency is essential.

3 Methodology

We consider a typical FL scenario with a total number of N clients, denoted as$\{C_1, ..., C_N\}$, each possessing its corresponding local private, non-iid dataset $\{\mathcal{D}_1, ..., \mathcal{D}_N\}$. The framework of our proposed method, OnePeFT, is illustrated in Figure 3.

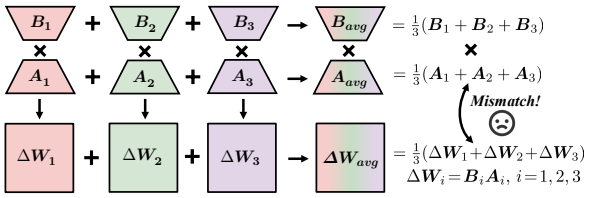


Figure 2: Aggregating LoRA parameters via weighted averaging may lead to inconsistency due to separately averaging the two low-rank matrices.

It consists of three key stages of training: *Client Update*, *SVD-based Aggregation*, and *Heterogeneous Knowledge Alignment*. The following sections will detail the components and processes of the proposed method. Moreover, we present pseudocode for OnePeFT in Appendix D.

3.1 Client Update

In the first stage, low-rank adaptation matrices are inserted as adapter into each transformer block of the LLM. The core idea of LoRA is to constrain the weight updates in the model through two low-rank decomposition matrices. More formally, the weight update is represented as:

$$\boldsymbol{W}_0 + \Delta\boldsymbol{W} = \boldsymbol{W}_0 + \boldsymbol{B}\boldsymbol{A}. \quad (1)$$

Here, the updates are applied on $\boldsymbol{A} \in \mathbb{R}^{r\times k}$ and $\boldsymbol{B} \in \mathbb{R}^{d\times r}$, where $r \ll min(d, k)$.

Each client C_i freeze the pre-trained LLM parameters and update only the inserted adapter. The fine-tuning is carried out via instruction tuning on local data $\mathcal{D}_i$. The optimization objective of the client update stage can be formulated as:

$$\mathcal{L}_i = \max_{\Theta} \sum_{(x,y)\in\mathcal{D}_i} \sum_{t=1}^{|y|} \log\left(p_{\Phi_0+\Delta\Phi(\Theta)}\left(y_t|x, y_{<t}\right)\right), \quad (2)$$

where x and y represent the *Instruction Input* and *Instruction Output*, respectively. Specifically, y_t denotes the t-th token in y, and $y_{<t}$ indicates all preceding tokens before y_t. Φ_0 refers to the frozen pre-trained parameters of the LLM, while Θ denotes the trainable parameters introduced by LoRA, with $|\Theta| \ll |\Phi_0|$.

After local training, only the LoRA adapter Θ are uploaded to the central server for subsequent knowledge alignment. Since $|\Theta| \ll |\Phi_0|$, the communication overhead is significantly reduced.

3.2 SVD-based Aggregation

After receiving the LoRA adapters uploaded by the clients, the server integrates them into a global initialization to serve as the starting point for subsequent alignment. The traditional model aggregation method, *FedAvg*, updates the global model

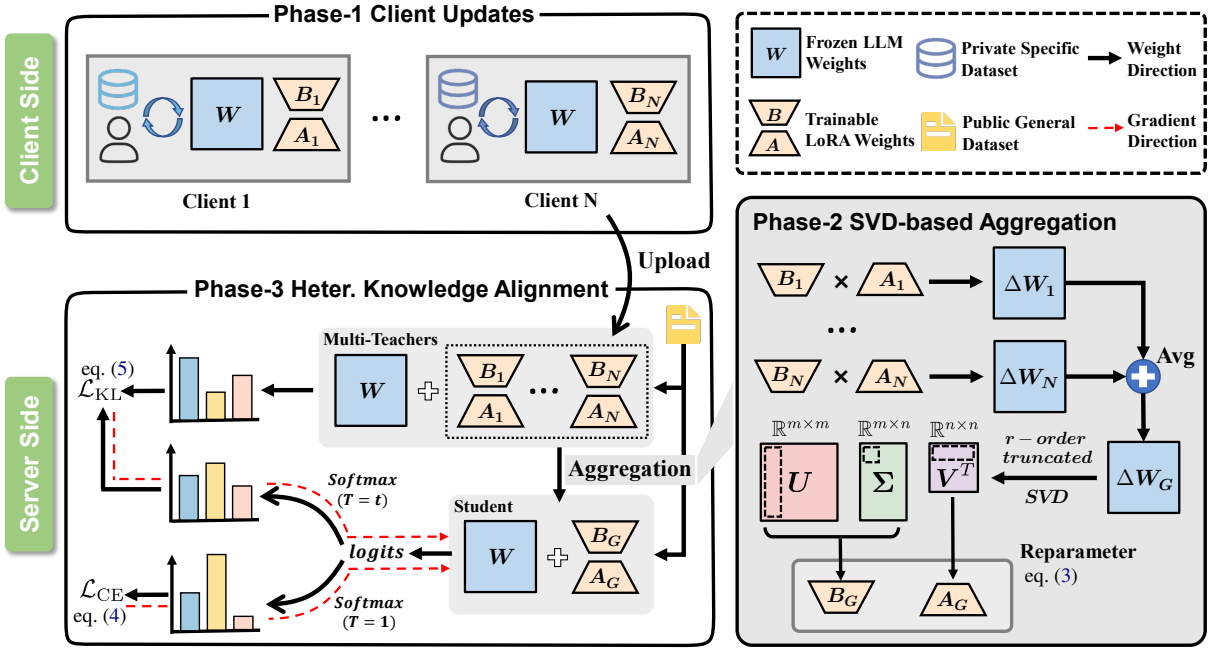


Figure 3: Framework of OnePeFT. ❶ Clients perform local LoRA-based instruction tuning with frozen LLM backbones (Sec. 3.1). ❷ Upon receiving the client LoRA adapters, the server performs SVD-based aggregation to mitigate model drift and initialize a global adapter (Sec. 3.2). ❸ The global adapter is aligned with heterogeneous client LoRA adapters via distillation, enhancing domain knowledge while preserving generalization. (Sec. 3.3).

by weighted averaging of local models, *i.e.*, $\mathcal{W} = \sum_{i=1}^{N} \lambda_i \mathcal{W}_i$, where λ_i is the weight of client i, reflects the data proportion of client i. This approach has been widely used in previous works on fine-tuning LLMs in FL scenarios (Zhang et al., 2023; Yue et al., 2024; Yan et al., 2024).

However, we argue that directly applying *FedAvg* to the LoRA adapter overlooks the core idea behind it, which jointly optimizes two low-rank matrices. As shown in Figure 2, after using *FedAvg* to aggregate the low-rank matrices, the produced $\boldsymbol{B}_{\text{avg}}\boldsymbol{A}_{\text{avg}}$ is inconsistent with ideal update $\sum_{i=1}^{N}\lambda_i \Delta \boldsymbol{W}_i$, potentially undermining convergence. To mitigate this mismatch, FFA-LoRA freezes $\boldsymbol{A}$ and averages only $\boldsymbol{B}$ across clients. However, this strategy implicitly restricts the global optimization space to the linear subspace spanned by the frozen $\boldsymbol{A}$ and depends heavily on initialization. As a result, FFA-LoRA lacks the flexibility to fully capture client features and may suffer from suboptimal generalization under heterogeneous data.

Perform SVD on LoRA parameters. Therefore, to mitigate the impact of the locally quadratic nature of LoRA, we reformulate the global update into a locally linear task. Specifically, we approximate the weight updates $\Delta \boldsymbol{W}_i \approx \boldsymbol{B}_i \boldsymbol{A}_i$ and then perform *FedAvg* over these approximatations to obtain the optimal global model update, *i.e.*, $\Delta \boldsymbol{W}_{\text{avg}} = \sum_{i=1}^{N} \lambda_i \Delta \boldsymbol{W}_i$. Since $\Delta \boldsymbol{W}_{\text{avg}}$ has the same dimension as the original weight matrix, it needs to be further decomposed into $\boldsymbol{B}_{\text{avg}}, \boldsymbol{A}_{\text{avg}}$.

We utilize Singular Value Decomposition (SVD) to perform low-rank reparameterization. Given a matrix $\boldsymbol{M} \in \mathbb{R}^{m\times n}$, SVD factorizes it as $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where $\boldsymbol{U} \in \mathbb{R}^{m\times m}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}$ is a diagonal matrix with singular values in descending order, and $\boldsymbol{V}^T \in \mathbb{R}^{n\times n}$. Specifically, we approximate $\Delta \boldsymbol{W}_{\text{avg}}$ by preserving its most significant information through a low-rank factorization, *i.e.*, $\Delta \boldsymbol{W}_{\text{avg}} \approx \tilde{\boldsymbol{B}}_{\text{avg}} \tilde{\boldsymbol{A}}_{\text{avg}}$, where $\tilde{\boldsymbol{B}}_{\text{avg}} \in \mathbb{R}^{m\times r}$, $\tilde{\boldsymbol{A}}_{\text{avg}} \in \mathbb{R}^{r\times n}$ and $r \ll \min(m, n)$. This approximation is obtained by retaining the top-r singular values of $\boldsymbol{\Sigma}$ with the corresponding vectors of $\boldsymbol{U}$ and $\boldsymbol{V}$. The final matrices are then constructed as:

$$\begin{aligned} \tilde{\boldsymbol{B}}_{\text{avg}} &= \boldsymbol{U}_{[1:m,1:r]}\boldsymbol{\Sigma}_{[1:r,1:r]}, \\ \tilde{\boldsymbol{A}}_{\text{avg}} &= \boldsymbol{V}^T_{[1:n,1:r]}. \end{aligned} \tag{3}$$

It is worth noting that the computational overhead introduced by the SVD-based aggregation and reparameterization is very small, taking up less than 1% of each client's local training time, which is acceptable in practic.

3.3 Heterogeneous Knowledge Alignment

Although the SVD-based aggregation provides a well-initialized global adapter $\tilde{\boldsymbol{B}}_{\text{avg}}$ and $\tilde{\boldsymbol{A}}_{\text{avg}}$, directly merging client adapters may still lead to parameter drift and sub-optimal performance. Traditional FL requires multiple communication rounds

for convergence. However, such high communication cost may be impractical, while the LoRA adapters are at an increased risk of being attacked. To mitigate these issues, we propose a heterogeneous knowledge alignment strategy that performs effective alignment between the global and client-specific LoRA adapters via multi-teacher distillation within a single communication round. In this stage, client-uploaded LoRA adapters serve as lightweight teachers from which the server distills the global adapter.

While traditional Knowledge Distillation (KD) (Hinton et al., 2015) relies on a proxy dataset to transfer knowledge from the teacher to student, FL imposes strict privacy constraints that limit data sharing. To address this, We utilize a public, task-agnostic dataset (e.g., Alpaca) for distillation, which is independent of any client's private data. Though not domain-specific, its diverse instruction-response pairs and broad linguistic coverage allow it to serve as a semantically meaningful medium for aligning the knowledge encoded in client adapters. This setup implicitly transfers domain knowledge through general data, enabling the global adapter to inherit domain-specific capabilities without violating privacy constraints. Our approach aligns with recent studies (Dong et al., 2024), which show that domain-specific tuning followed by general-data adaptation can effectively transfer knowledge. By aligning the global adapter $\tilde{\boldsymbol{B}}_{\text{avg}}$ and $\tilde{\boldsymbol{A}}_{\text{avg}}$ with client-specific knowledge, our method enables efficient transfer of domain-specific insights.

The loss during the KD process consists of two parts. The first part is the cross-entropy loss, which is used to enhance the model's general capabilities:

$$\mathcal{L}_{\text{CE}} = -\sum_{(x,y)\in\mathcal{D}_{\text{pub}}} \sum_{t=1}^{|y|} \log\left(q_\theta\left(y_t|x, y_{<t}\right)\right). \quad (4)$$

This formula is similar to Equation (2), where $\mathcal{D}_{\text{pub}}$ represents a general dataset, distinct from the domain-specific datasets $\{\mathcal{D}_1, ..., \mathcal{D}_N\}$ of individual clients. Minimizing the cross-entropy loss allows the model to learn general knowledge from the general public dataset. Here q_θ is the predicted distribution based on the frozen pre-trained parameters of the LLM and the global LoRA adapter. By doing so, the model retains its ability to perform well on general tasks.

The second part of the loss is the KL divergence, which incorporates the distributions from all the clients. It encourages the student model to align its output distribution with those of the teacher models, thereby enhancing its domain-specific capability. The KL divergence is formulated as:

$$\mathcal{L}_{\text{KL}} = -\sum_{(x,y)\in\mathcal{D}_{\text{pub}}} \sum_{t=1}^{|y|} \sum_{y_t\in V} \left(\sum_{i=1}^{N} \lambda_i p_i(y_t|x, y_{<t})\right) \log \frac{\sum_{i=1}^{N} \lambda_i p_i(y_t|x, y_{<t})}{q_\theta(y_t|x, y_{<t})}, \quad (5)$$

where p_i denotes the predicted distribution based on the frozen pre-trained parameters of the LLM and the client i's LoRA adapter.

We integrate the above losses to form the full objective:

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{CE}} + (1-\alpha)\mathcal{L}_{\text{KL}}, \quad (6)$$

where α controls the trade-off between these two components, balancing the model's generalization ability (via $\mathcal{L}_{\text{CE}}$) and domain-specific knowledge transfer (via $\mathcal{L}_{\text{KL}}$).

4 Experiments

4.1 Experimental Setup

Datasets. In our experiments, we train and evaluate LLM on three NLP tasks: math problem-solving, code generation, and legal document analysis.

- For **math problem-solving**, we use the GSM-8K dataset (Cobbe et al., 2021), a grade school math problem dataset released by OpenAI.
- For **code generation**, we fine-tune the model on the Rosetta-Alpaca dataset (Chaudhary, 2023), and evaluate it on the HumanEvalX benchmark (Zheng et al., 2023) that requires the model to generate code solutions for given problems.
- For **legal document analysis**, we collect datasets for five **Chinese** legal NLP tasks from publicly available legal benchmarks. All datasets are split into training and test sets.
- For the **public dataset** used in OnePeFT's distillation, we adopt Alpaca (Taori et al., 2023) for math and code tasks, and Alpaca-GPT4-zh (LlamaFactory, 2023) for the legal task, due to their diverse, high-quality instruction-following samples in English and Chinese. To evaluate generalization, we also test on HELM (Liang et al., 2022) tasks and the Alpaca-GPT4-zh test set.

Although the public datasets are not tailored to our target domains, their broad linguistic coverage and general-purpose nature make them effective proxies for aligning the global adapter with client-specific knowledge. Details of the datasets and client data partitioning are provided in Appendix A.

Methods	C++			Java			GO			Python		
	α=0.1	α=0.3	α=0.7	α=0.1	α=0.3	α=0.7	α=0.1	α=0.3	α=0.7	α=0.1	α=0.3	α=0.7
LLaMA3-8B												
Zero-shot	33.32	33.32	33.32	35.79	35.79	35.79	24.27	24.27	24.27	37.16	37.16	37.16
Local	33.40	34.19	34.22	36.02	36.64	37.01	26.21	26.85	27.68	37.71	37.56	38.63
FedPETuning	34.15	35.62	35.89	37.85	38.71	39.06	28.18	28.20	29.97	38.95	40.52	41.63
FeDeRA	34.22	35.77	35.82	37.93	38.65	39.17	28.31	28.95	29.70	38.66	40.13	41.78
FFA-LoRA	33.67	36.91	37.92	37.31	**39.81**	39.85	26.52	28.01	30.19	38.87	**41.59**	42.06
OnePeFT	**35.60**	**37.08**	**37.95**	**38.12**	39.53	**40.49**	**29.76**	**30.28**	**31.13**	**39.58**	41.18	**42.75**
Qwen2-7B												
Zero-shot	40.24	40.24	40.24	45.73	45.73	45.73	35.67	35.67	35.67	48.93	48.93	48.93
Local	40.51	40.66	40.95	46.07	46.43	46.47	35.71	35.77	36.16	48.94	49.10	49.21
FedPETuning	40.96	41.24	41.98	47.50	48.29	48.79	37.13	37.74	38.05	49.48	49.57	49.77
FeDeRA	41.32	41.25	41.78	47.82	48.31	48.76	37.41	37.69	38.13	49.61	49.68	49.74
FFA-LoRA	41.42	43.69	43.89	48.15	49.03	49.31	**38.05**	37.89	38.43	**50.95**	50.91	51.43
OnePeFT	**42.05**	**43.72**	**44.12**	**48.61**	**49.27**	**49.61**	37.56	**39.48**	**40.13**	50.74	**51.29**	**51.83**

Table 1: Pass@1 performance comparison on HumanEvalX across different non-i.i.d. settings. **Bold** is the best.

Models	Methods	ACC ↑
LLaMA3-8B	Centralized	56.03% (739/1319)
	Few-shot CoT	43.59% (575/1319)
	Local	47.99% (633/1319)
	FedPETuning	50.72% (669/1319)
	FeDeRA	51.71% (682/1319)
	FFA-LoRA	**54.59% (720/1319)**
	OnePeFT	54.13% (714/1319)
Qwen2-7B	Centralized	82.87% (1093/1319)
	Few-shot CoT	79.38% (1047/1319)
	Local	79.91% (1054/1319)
	FedPETuning	80.36% (1060/1319)
	FeDeRA	79.83% (1053/1319)
	FFA-LoRA	81.50% (1075/1319)
	OnePeFT	**82.49% (1088/1319)**

Table 2: Accuracy (%) comparison on the GSM-8K dataset. **Bold** is the best.

Baselines. We compare OnePeFT with representative FL-based LLM fine-tuning methods, including FedPETuning (Zhang et al., 2023), FeDeRA (Yan et al., 2024), and FFA-LoRA (Sun et al., 2024). We also include the original pre-trained model, a centrally trained model, and client-specific models for reference.

Implementation Details. In this paper, all methods utilize LLaMA3-8B (Grattafiori et al., 2024) and Qwen2-7B (Yang et al., 2024a) as the pre-trained LLM. The number of clients varies by task, as detailed in Appendix A. During training, the LoRA rank is set to 8, and the LoRA scaling factor is set to 16. The balance factor α is set to 0.5. Due to space constraints, a full description of the experimental setup is provided in Appendix B.

Evaluation Metrics. The evaluation metrics for math problem-solving and code generation follow previous work (Kuang et al., 2024; Wu et al., 2024), while legal document analysis uses task-specific metrics for each subtask, detailed in Appendix C.

4.2 Domain-Specific Experimental Results

Results on Math Problem-Solving. We present the results on GSM-8K in Table 2. For LLaMA3-8B, FFA-LoRA achieves the best accuracy, while OnePeFT performs competitively performance with 10× less communication. For Qwen2-7B, OnePeFT outperforms all baselines. Although centralized training yields the best accuracy, it requires full data sharing, which is often impractical. Moreover, we observe larger gains from fine-tuning on LLaMA3-8B than on Qwen2-7B, suggesting that fine-tuning is more effective when the pre-trained model is less aligned with the target domain.

Results on Code Generation. Table 1 shows Pass@1 results across different programming languages under various non-i.i.d. settings. For both pre-trained LLM, OnePeFT consistently achieves best or close to best performance across all settings, particularly under the most challenging setting with α=0.1, with only a single round of communication. Although FFA-LoRA attains slightly higher scores in a few specific cases, OnePeFT demonstrates a more balanced and robust performance overall. These results highlight the effectiveness of our method in enabling efficient federated fine-tuning for code generation.

Results on Legal Document Analysis. Since Qwen2-7B offers stronger chinese understanding, and the legal data focuses on chinese texts, we fine-tune only on Qwen2-7B. As shown in Table 3, Zero-shot performance remains lowest, highlighting the difficulty of applying LLMs to legal tasks. While all baselines improve legal performance, they sacrifice general ability. In contrast, OnePeFT achieves the best general-domain performance (72.08) while maintaining competitive or leading results on legal

Methods	DRC	JE		LEE	JP		DS	General
	BertScore	Accuracy	BertScore	Accuracy	Accuracy	BertScore	BertScore	BertScore
Zero-shot	77.83	42.99	74.99	37.29	33.63	70.29	69.53	71.04
Local	78.81	42.80	75.51	42.63	40.94	71.03	70.14	69.11
FedPETuning	81.84	42.48	74.95	58.71	67.55	69.84	71.44	65.24
FeDeRA	82.96	43.52	75.39	58.62	68.78	72.58	72.10	63.19
FFA-LoRA	84.37	45.31	**78.28**	62.07	71.56	**74.84**	73.32	67.25
OnePeFT	**85.03**	**47.22**	76.53	**64.25**	**72.24**	73.70	**73.50**	**72.08**

Table 3: Performance comparison on five legal tasks and a general-domain task using Qwen2-7B. **Bold** is the best.

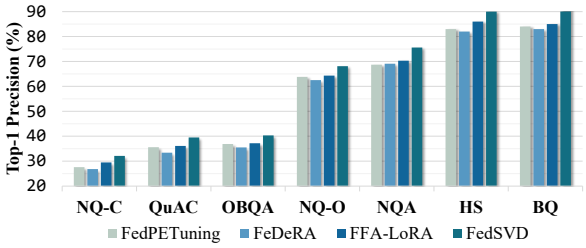


Figure 4: Performance comparison on the HELM benchmark after fine-tuning LLaMA3-8B on GSM-8K.

tasks. These results demonstrate the effectiveness of our method in balancing domain adaptation and generalization.

4.3 General Ability Experimental Results

To evaluate general capability retention after domain-specific fine-tuning, we assess all methods on the HELM benchmark. As shown in Figure 4, most baselines exhibit noticeable performance drops—particularly FeDeRA. In contrast, OnePeFT, aided by distillation on general data, consistently outperforms all baselines across tasks. This highlights that our distillation strategy not only adapts well to target domains but also mitigates catastrophic forgetting of general abilities.

4.4 Computation and Communication Costs

As shown in Table 4, we compare the communication and computation efficiency of different methods. Since FeDeRA and FedPETuning share the same costs, only the latter is report. Although both FedPETuning and OnePeFT fine-tune 20.19M parameters, OnePeFT completes training in a single round, reducing communication cost from 3088.4MB to just 154.4MB—a 20× reduction. Compared to FFA-LoRA, which transmits only half of the parameters but requires 20 rounds, our method still achieves lower overall cost. In terms of computation, OnePeFT also shows significant savings. While it introduces server-side distillation, the overall computation cost is greatly reduced by eliminating costly local training on clients. Specifically, OnePeFT lowers total computation to 3.24×10^9 GFLOPs, a 54% reduction

Methods	Trainable Param.	Comm. Round	Comm. Costs	Comp. Costs
FedPETuning	20.19	20	3088.4	7.11×10^9
FFA-LoRA	11.01	20	1544.2	7.09×10^9
OnePeFT	20.19	1	**154.4**	$\mathbf{3.24\times10^9}$

Table 4: Communication and computation efficiency comparison of different methods using Qwen2-7B.

SVD	HKA	ACC	Pass@1	Top-1 Acc.
		45.94%	33.96	57.17
✓		47.08%	34.58	56.91
	✓	52.99%	37.71	61.99
✓	✓	**54.13%**	**38.08**	**62.16**

Table 5: Ablation study on the key components of OnePeFT using LLaMA3-8B

compared to 7.11×10^9 for FedPETuning. Although FFA-LoRA halves the number of trainable parameters, the LoRA parameters make up only a small portion of the full model, resulting in limited computation savings. These results highlight the efficiency of OnePeFT, making it well-suited for FL.

4.5 Ablation Study

To better understand the contributions of each component in OnePeFT, we conduct an ablation study using LLaMA3-8B, as shown in Table 5. Removing both SVD-based aggregation (SVD) and Heterogeneous Knowledge Alignment (HKA) yields the lowest performance, indicating that directly averaging client-trained adapters leads to parameter drift and sub-optimal results. Introducing SVD alone improves performance, confirming its effectiveness in aggregating LoRA parameters. Similarly, using only HKA brings notable gains, showing its role in aligning the global adapter with client knowledge. Combining both achieves the best results, demonstrating that SVD provides a better initialization for the alignment process and that the two components are complementary.

Hyperparameter Analysis. We study the impact of LoRA rank r and the balance factor α on the GSM-8K dataset. As shown in Figure 5, we vary one hyperparameter while keeping others fixed to the default settings (refer to Section 4.1). Increasing r initially improves both domain-specific

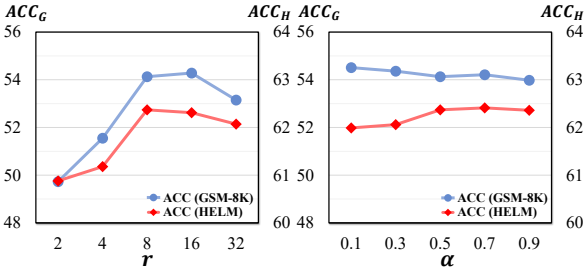


Figure 5: Hyperparameter analysis on GSM-8K and HELM datasets with varying hyperparameters.

Public Dataset	#Samples	ACC
Alpaca	52,002	54.13%
Dolly	15,011	53.83%
InstructionWild	52,190	55.12%

Table 6: Aligning LoRA adapters fine-tuned on GSM-8K using different public datasets with LLaMA3-8B.

and general performance, as a larger rank captures richer subspaces. However, further increasing r degrades performance, likely due to overparameterization failing to capture additional useful subspaces. As for the balance factor α, the performance remains stable across different values, suggesting that *Heterogeneous Knowledge Alignment* is robust and not sensitive to the choice of α.

Impact of Different Public Datasets. To investigate whether the choice of public datasets affects the alignment performance, we additionally experiment with InstructWild (Ni et al., 2023) and Dolly (Conover et al., 2023) as distillation datasets. As shown in Table 6, the choice of public dataset has slight impact on alignment quality, suggesting that datasets with broad linguistic coverage and general-purpose instructions can serve as effective proxies.

4.6 Visualization

SVD-based aggregation. To verify the superiority of our SVD-based aggregation, we visualize the differences between the aggregated weights and the assumed global weight. Specifically, we first compute the weight update matrices from the aggregated LoRA parameters, and then decompose each weight matrix into a magnitude vector m and a normalized directional matrix V following (Liu et al., 2024a), based on which we compute their differences. As shown in Figure 6, SVD-based aggregation consistently yields smaller variations across all LoRA weight matrices, indicating reduced aggregation error and mitigated model drift.

Heterogeneous Knowledge Alignment. To explore whether domain-specific knowledge can be transferred via general datasets, we sample 100 queries from both domain-specific and general

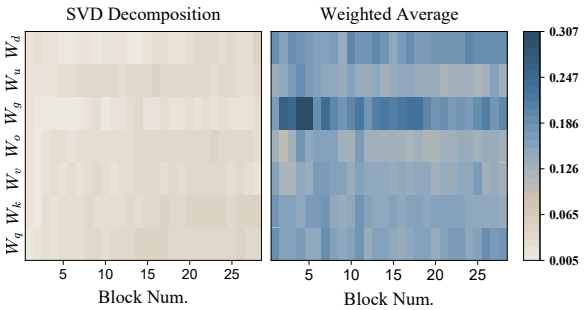
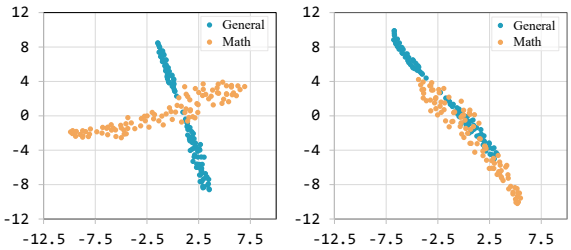


Figure 6: Directional variation of different aggregation methods w.r.t. the assumed global weight (lower is better). Magnitude results are in Appendix E.



(a) Qwen2-7B (b) Qwen2-7B (GSM-8K)

Figure 7: t-SNE visualization of 15th-layer representations for math and general data: (a) original Qwen2-7B, (b) Qwen2-7B fine-tuned on GSM-8K.

datasets. We then extracted the hidden representations from the middle layer (15th) of the model and visualized them using t-SNE (Van der Maaten and Hinton, 2008). As shown in Figure 7, the original model shows nearly orthogonal distributions with minimal overlap. After domain-specific fine-tuning, both distributions align in direction with increased overlap, indicating that domain features have been embedded into the general data representation, resulting in a more unified semantic space.

5 Conclusion

In this paper, we propose OnePeFT, a novel one-shot FL framework for LLM fine-tuning. We interpret the global model aggregation as a process of aligning heterogeneous knowledge from clients. To this end, we introduce an SVD-based aggregation strategy to mitigate parameter drift and obtain a well-initialized global LoRA adapter. Furthermore, we perform knowledge alignment between the global and client LoRA adapters via distillation on a task-agnostic dataset, thereby enhancing domain-specific performance while preserving the generalization. Experiments on LLaMA3-8B and Qwen2-7B demonstrate that OnePeFT achieves competitive performance with significantly lower communication cost, highlighting its effectiveness for real-world LLM fine-tuning.

Limitations

One limitation of this work lies in the computational burden on clients. Although we leverage LoRA for efficient fine-tuning and reduce the entire training process to a single communication round, each client is still required to host the full LLM during local training. This can pose practical challenges for deployment in resource-constrained environments. We believe future work could explore lightweight alternatives or collaborative strategies to further lower the local cost, making federated fine-tuning more accessible and scalable across diverse edge devices.

Ethical Considerations

We propose OnePeFT, a one-shot parameter-efficient federated tuning framework for LLMs, designed to leverage private data while safeguarding user privacy. By reducing communication to a single round, OnePeFT minimizes the risk of privacy leakage and exposure to potential adversarial attacks. All training data used in this work are sourced from open-source NLU and NLG projects, strictly adhering to their license terms and public benchmark guidelines. This research contributes to the development of privacy-preserving LLM adaptation, promoting ethical and socially responsible use of federated learning technologies.

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*.

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*.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Differentially private bias-term only fine-tuning of foundation models, 2023. In *URL https://openreview. net/forum*.

CAIL. 2020. Cail2020. https://github.com/china-ai-law-challenge/CAIL2020.

CAIL. 2022. Cail2022. https://github.com/china-ai-law-challenge/CAIL2020.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation. GitHub repository.

Chaochao Chen, Xiaohua Feng, Jun Zhou, Jianwei Yin, and Xiaolin Zheng. 2023. Federated large language model: A position paper. *arXiv e-prints*, pages arXiv–2307.

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 *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12903–12913.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. Databricks Blog.

Rong Dai, Yonggang Zhang, Ang Li, Tongliang Liu, Xun Yang, and Bo Han. 2024. Enhancing one-shot federated learning through data and ensemble co-boosting. In *The Twelfth International Conference on Learning Representations*.

Don Kurian Dennis, Tian Li, and Virginia Smith. 2021. Heterogeneity for the win: One-shot federated clustering. In *International conference on machine learning*, pages 2611–2620. PMLR.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How abilities in large language models are affected by supervised fine-tuning data composition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 177–198.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Clare Elizabeth Heinbaugh, Emilio Luz-Ricca, and Huajie Shao. 2023. Data-free one-shot federated learning under very high statistical heterogeneity. In *The Eleventh International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. Lawyer llama technical report. *arXiv preprint arXiv:2305.15062*.

Anirudh Kasturi and Chittaranjan Hota. 2023. Osgan: One-shot distributed learning using generative adversarial networks. *The Journal of Supercomputing*, 79(12):13620–13640.

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.

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.

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 965–978. IEEE.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Hongcheng Liu, Yusheng Liao, Yutong Meng, and Yuhao Wang. 2023. Lawgpt. https://github.com/LiuHC0428/LAW_GPT.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024a. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

Xiang Liu, Liangxi Liu, Feiyang Ye, Yunheng Shen, Xia Li, Linshan Jiang, and Jialin Li. 2024b. Fedlpa: One-shot federated learning with layer-wise posterior aggregation. *Advances in Neural Information Processing Systems*, 37:81510–81548.

LlamaFactory. 2023. Alpaca_gpt4_zh. https://huggingface.co/datasets/llamafactory/alpaca_gpt4_zh.

Jinjie Ni, Fuzhao Xue, Kabir Jain, Mahir Hitesh Shah, Zangwei Zheng, and Yang You. 2023. Instruction in the wild: A user-based instruction dataset. https://github.com/XueFuzhao/InstructionWild.

Shangchao Su, Bin Li, and Xiangyang Xue. 2023. One-shot federated learning without server-side training. *Neural Networks*, 164:203–215.

Guangyu Sun, Matias Mendieta, Taojiannan Yang, and Chen Chen. 2022. Exploring parameter-efficient fine-tuning for improving communication efficiency in federated learning.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. 2024. Improving lora in privacy-preserving federated learning. In *The Twelfth International Conference on Learning Representations*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Derui Wang, Chaoran Li, Sheng Wen, Surya Nepal, and Yang Xiang. 2020. Man-in-the-middle attacks against machine learning classifiers via malicious generative models. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2074–2087.

Colin Wei, Sang Michael Xie, and Tengyu Ma. 2021. Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. *Advances in Neural Information Processing Systems*, 34:16158–16170.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Feijie Wu, Zitao Li, Yaliang Li, Bolin Ding, and Jing Gao. 2024. Fedbiot: Llm local fine-tuning in federated learning without full model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3345–3355.

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

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024b. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Mingzhao Yang, Shangchao Su, Bin Li, and Xiangyang Xue. 2024c. Feddeo: Description-enhanced one-shot federated learning with diffusion models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6666–6675.

Feng Yao, Chaojun Xiao, Xiaozhi Wang, Zhiyuan Liu, Lei Hou, Cunchao Tu, Juanzi Li, Yun Liu, Weixing Shen, and Maosong Sun. 2022. Leven: A large-scale chinese legal event detection dataset. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 183–201.

Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. 2021. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16337–16346.

Linan Yue, Qi Liu, Yichao Du, Weibo Gao, Ye Liu, and Fangzhou Yao. 2024. Fedjudge: Federated legal large language model. In *International Conference on Database Systems for Advanced Applications*, pages 268–285. Springer.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, et al. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325*.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9.

Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. 2022a. Dense: Data-free one-shot federated learning. *Advances in Neural Information Processing Systems*, 35:21414–21428.

Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. 2022b. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10174–10183.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *8th International Conference on Learning Representations, ICLR 2020*, pages 26–30, Addis Ababa, Ethiopia.

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).

Qinkai Zheng, Xiao Xia, Xu Zou, Yuxiao Dong, Shan Wang, Yufei Xue, Lei Shen, Zihan Wang, Andi Wang, Yang Li, et al. 2023. Codegeex: A pre-trained model for code generation with multilingual benchmarking on humaneval-x. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5673–5684.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. Jec-qa: a legal-domain question answering dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9701–9708.

A Datasets

As described in Table 7, we fine-tune the LLM on three NLP tasks and perform distillation using either a Chinese or an English general dataset, depending on the task. This section briefly introduces all the datasets used in our experiments.

GSM-8K. GSM-8K is a high-quality and linguistically diverse dataset of grade school math word problems released by OpenAI (Cobbe et al., 2021), commonly used to evaluate the mathematical reasoning ability of LLMs. The dataset consists of 7,473 training samples and 1,319 testing samples. Since it is not divided into categories and all samples are of relatively similar length and complexity, we split the GSM-8K training dataset ensuring i.i.d. across three clients.

Rosetta-Alpaca. The Rosetta-Alpaca dataset (Chaudhary, 2023) consists of 7,969 code generation samples across nine different programming languages. The dataset is used for fine-tuning models on code generation tasks, and the model's performance is evaluated on the HumanEvalX benchmark, detailed in Appendix C. According to Table 7, we split Rosetta-Alpaca in non-i.i.d. style. To simulate data heterogeneity across clients, we adopt the Dirichlet distribution $Dir(\alpha)$ (Li et al., 2022) to generate non-i.i.d. data splits, where a smaller α indicates higher data heterogeneity.

Legal Datasets. The Legal Document Analysis datasets, as detailed in Table 7, cover five legal NLP subtasks, each assigned to a specific client. These datasets were sourced from publicly available legal benchmarks and open-source instruction datasets. These include Legal Text Summarization (CAIL, 2020, 2022), LEVEN (Yao et al., 2022), Legal Question Answering (Zhong et al., 2020), Lawyer LLaMA (Huang et al., 2023), LawGPT-zh (Liu et al., 2023), and DISC-LawLLM (Yue et al., 2023). All datasets are transformed into "input-output" pairs, which are then split into training and test sets. In the following, we provide a detailed description of the five task categories.

- **Document Reading Comprehension(DRC):** Given a case description or legal document, answer the given questions to assess LLM's ability to understand legal texts.
- **Judicial Examination(JE):** Provide answers and explanations for legal exam questions, evaluating the LLM's knowledge retention and analytical reasoning in the context of legal assessments.
- **Legal Element Extraction(LEE):** Divided into two parts: event detection and element extraction. The LLM is tasked with labeling legal cases or extracting key entities, assessing its proficiency in identifying and extracting legal elements.
- **Judgment Prediction(JP):** Divided into two parts: legal case classification and case judgment prediction. The LLM is required to classify cases or predict judicial outcomes, evaluating its ability to comprehend and reason about legal cases.
- **Document Summarization(DS):** To generate summaries for the provided legal documents or public opinion reports, assessing the LLM's ability to condense and extract key information from legal texts.

Alpaca. The Alpaca dataset (Taori et al., 2023), introduced by Stanford's CRFM, comprises 52,002 instruction-following demonstrations generated using OpenAI's text-davinci-003 model. This dataset is designed to facilitate instruction-tuning for language models, enhancing their ability to follow diverse instructions. In our study, we utilize the Alpaca dataset as a general dataset during the Task-Agnostic Distillation phase for math problem-solving and code generation tasks.

Alpaca-GPT4-zh. The Alpaca-GPT4-zh dataset (LlamaFactory, 2023) is a Chinese-language instruction-following dataset. It contains 43,937 instruction-output pairs, formatted similarly to the original Alpaca dataset, but with outputs generated by GPT-4. This dataset is specifically designed to improve the instruction-following capabilities of LLMs in Chinese. In our study, we utilize the Alpaca-GPT4-zh dataset as a general dataset during the Task-Agnostic Distillation phase for the legal document analysis task.

B Complete Experimental Setup

Base LLM. We adopt LLaMA3-8B (Grattafiori et al., 2024) and Qwen2-7B (Yang et al., 2024a) as the pre-trained large language models for all experiments.

Hyperparameters. For training, the LoRA rank is set to 8 with a scaling factor of 16, and the balance factor α is set to 0.5. The number of communication rounds for other baseline is set to 20, whereas for OnePeFT, only a single round of parameter upload is performed, fixing the communication round to 1. The local training epoch for baseline is set to 2, while for OnePeFT, the

Task	Training Dataset	#train	#client	Partition Strategy	Avg. Input Length	Avg. Output Length	Test Dataset	#test	LICENSE
Math Problem Solving	GSM-8K	7,473	3	i.i.d.	235.3	288.3	GSM-8K	1,319	MIT License
Code Generation	Rosetta-Alpaca	7,969	8	Non-i.i.d.	1383.4	1381.1	HumanEvalX	656	Apache-2.0
Legal Document Analysis	DRC	34,677	5	Task-Specific	601.7	59.0	DRC	3,853	Unspecified
	JE	18,946			212.9	311.2	JE	2,105	Unspecified
	LEE	1,9030			153.9	19.0	LEE	2,022	Unspecified
	JP	15,606			430.0	255.9	JP	1,733	Unspecified
	DS	5,659			1187.3	176.8	DS	627	Unspecified
Public Dataset	Alpaca	52,002	—	—	59.8	270.3	HELM	—	CC BY-NC 4.0
	Alpaca-GPT4-zh	43,937			21.3	227.9	Alpaca-GPT4-zh	4,881	Apache-2.0
	Dolly	15,011			424.7	358.1	HELM	—	CC BY-SA 3.0
	InstructionWild	52,190			72.1	676.67	HELM	—	Unspecified

Table 7: Detailed statistics of datasets for LLM training and evaluation in our experiments.

local training epochs is set to 8, and the distillation epochs is set to 3. We adopt the Adam optimizer and search for the optimal learning rate over $\{1\times10^{-5}, 4\times10^{-5}, 8\times10^{-5}, 1\times10^{-4}, 2\times10^{-4}\}$. The momentum coefficients (β_1, β_2) are set to $(0.9, 0.95)$. All other optimizer-related hyperparameters follow the default settings. During training, the batch size per device is set to 1, the gradient accumulation step is 8, and the maximum sequence length is 2048.

Environment The experiments are conducted in the following environment:

- **Operating System:** Ubuntu 20.04.1
- **CPU:** AMD Ryzen Threadripper PRO 5945WX
- **GPU:** NVIDIA GeForce RTX 3090 Ti

C Evaluation Metrics

In this section, we introduce the evaluation metrics used to assess the fine-tuning performance of the LLMs. As shown in Table 7, we apply various datasets, and here we describe the metrics used to measure the model's performance on each of these datasets.

GSM-8K. We use the GSM-8K test set to evaluate the performance of a LLM in solving math problems. The dataset consists of "questions" and their corresponding "ground truth" answers. To assess correctness, we measure the accuracy by calculating the rate at which the LLM provides the correct answer to a given question. We follow the Chain of Thought (CoT) (Wei et al., 2022) approach by preparing a set of sample questions (i.e., few-shot prompting) and prompting the LLM to generate step-by-step solutions. The generated answers are then extracted and compared to the ground truth to compute the correctness rate.

HumanevalX. This task evaluates code autofill performance using a dataset consisting of 164 test samples across five programming languages (Zheng et al., 2023). For our evaluation, we focus on four languages (C++, GO, Java, and Python), as JavaScript is not included in the training dataset. Each test sample contains the following components: "task id," "prompt" (task description with partial code), "entry point" (function to be implemented), "canonical solution" (a reference solution), and "test" (a unit test to evaluate whether the generated code produces the correct output for the given input). In this task, we treat the "prompt" as the input and generate twenty versions of code using the given model. The generated codes are then compiled, and we check if they pass the corresponding unit tests. Let c represent the number of correct codes that pass the test. The Pass@k metric is calculated by determining the proportion of test samples for which at least k of the generated code versions pass the unit tests:

$$\text{Pass@k} = \mathbb{E}_{\text{problems}}\left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}\right]$$

Legal Datasets. For legal document analysis, we split the collected datasets into training and test sets, using the test set for evaluation. The model performance across different tasks is assessed using task-specific metrics. For Document Reading Comprehension (DRC), Judgment Prediction (JP), Document Summarization (DS), and General, we use BERTScore (Zhang et al., 2020), which measures the similarity between the generated and reference texts. Legal Element Extraction (LEE) is evaluated based on accuracy, determining whether the extracted elements are correct. Judicial Examination (JE) is assessed using both BERTScore to

Algorithm 1 OnePeFT

Input: Pretrained model Φ_0, clients N, local datasets $\{\mathcal{D}_i\}_{i=1}^{N}$, public general dataset $\mathcal{D}_{\text{pub}}$, LoRA rank r, balance factor α.

Output: Global model $\Phi_{\text{final}} = \Phi_0 + \Delta W_{\text{final}}$.

Client Side:

`/* Phase 1: Client Updates */`

foreach client $i = 1, 2, \ldots, N$ in parallel **do**

  Initialize local LoRA parameters $\Theta_i \leftarrow \{B_i, A_i\}$

  Train Θ_i on $\mathcal{D}_i$ by maximizing Eq.(2)

  Upload Θ_i to central server

end

Server Side:

`/* Phase 2: SVD-based Aggregation */`

Compute local update: $\Delta W_i \leftarrow B_i A_i$

Compute weighted average: $\Delta W_{\text{avg}} \leftarrow \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{\sum_j |\mathcal{D}_j|} \Delta W_i$

Perform SVD decomposition: $U \Sigma V^T \leftarrow \text{SVD}(\Delta W_{\text{avg}})$

Extract low-rank approximation:

$\tilde{B}_{\text{avg}} \leftarrow U_{[1:m,1:r]} \Sigma_{[1:r,1:r]}$

$\tilde{A}_{\text{avg}} \leftarrow V^T_{[1:n,1:r]}$

Construct global LoRA: $\Theta_{\text{global}} \leftarrow \{\tilde{B}_{\text{avg}}, \tilde{A}_{\text{avg}}\}$

`/* Phase 3: Heterogeneous Knowledge Alignment */`

Optimize combined loss on $\mathcal{D}_{\text{pub}}$:

while not converged **do**

  Compute cross-entropy loss $\mathcal{L}_{\text{CE}}$ ▷ Eq.(4)

  Compute KL divergence $\mathcal{L}_{\text{KL}}$ ▷ Eq.(5)

  Update parameters: $\Theta_{\text{final}} \leftarrow \arg\min_{\Theta(\Theta_{\text{global}})} (\alpha \mathcal{L}_{\text{CE}} + (1-\alpha)\mathcal{L}_{\text{KL}})$

end

return $\Phi_{\text{final}} = \Phi_0 + \tilde{B}_{\text{final}} \tilde{A}_{\text{final}}$.

measure the quality of explanations and accuracy to verify the correctness of the answers. These metrics together provide a comprehensive evaluation of the model's performance on various aspects of legal document analysis.

HELM. HELM (Liang et al., 2022) is a benchmark that encompasses a wide range of NLP tasks, used to evaluate the general capabilities of fine-tuned models. We upload the well-trained models to the benchmark and evaluate them on general question-answering tasks, which include seven datasets: BoolQ, NarrativeQA, Natural Questions (closed-book), Natural Questions (open-book), QuAC, HellaSwag, and OpenbookQA. For different tasks, the evaluation metrics vary: exact match for HellaSwag and OpenbookQA; quasi-exact match for BoolQ; and F1 score for the remaining tasks.

D Pseudocode

To facilitate understanding and implementation, we present the pseudocode of the OnePeFT framework in Algorithm 1, detailing the key procedures including *Client Updates*, *SVD-based Aggregation* and *Heterogeneous Knowledge Alignment*. In the pseudocode, the notation $x(x_{\text{init}})$ indicates that x is initialized with x_{init}.

E Additional Experimental Results

E.1 Visualization

We present the magnitude variations in Figure 8, where the SVD-based aggregation consistently results in smaller variations across all LoRA weight matrices, indicating reduced aggregation errors and mitigated model drift.

Additionally, we visualize the hidden representations from the 15th layer of the model fine-tuned

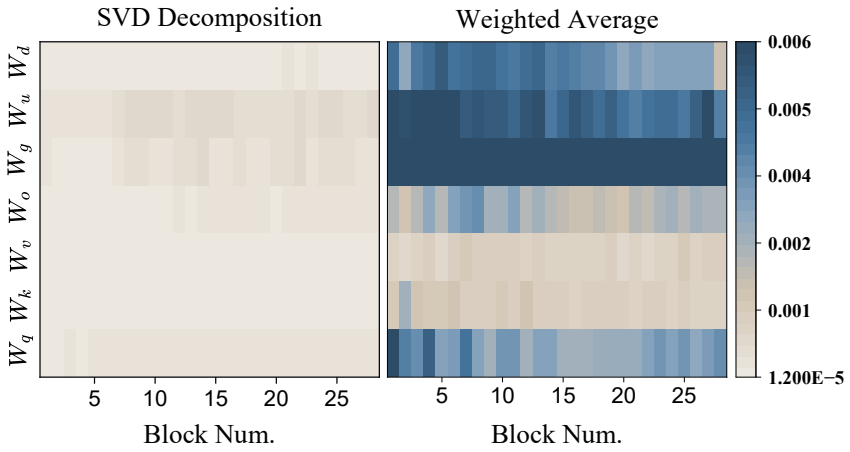
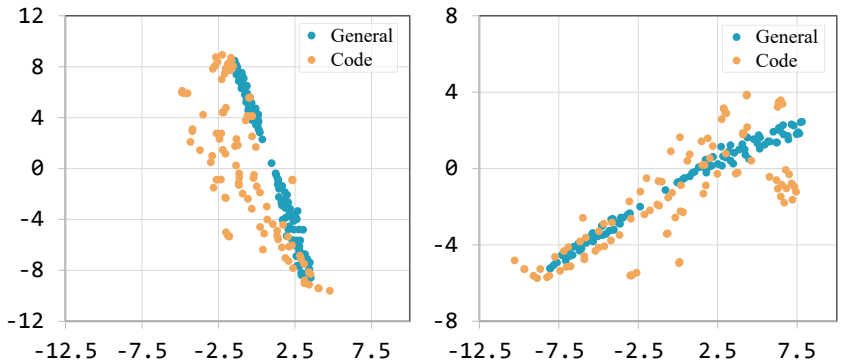
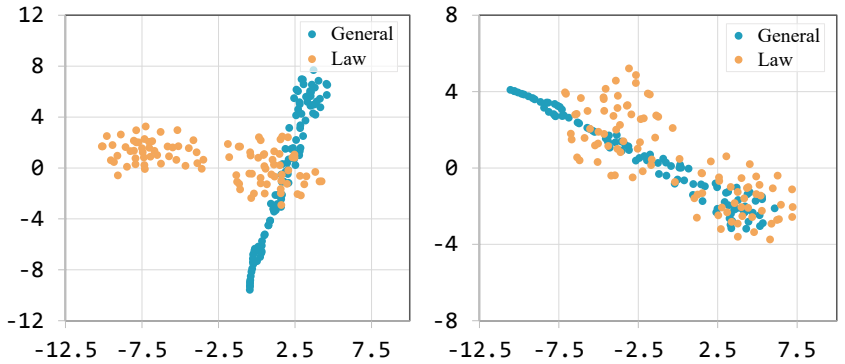


Figure 8: Magnitude variation of different aggregation methods w.r.t. the assumed global weight (lower is better).



(a) Qwen2-7B (b) Qwen2-7B (Rosetta-Alpaca)

Figure 9: t-SNE visualization of 15th-layer representations for code and general data: (a) original Qwen2-7B, (b) Qwen2-7B fine-tuned on Rosetta-Alpaca.



(a) Qwen2-7B (b) Qwen2-7B (Judicial Examination)

Figure 10: t-SNE visualization of 15th-layer representations for legal and general data: (a) original Qwen2-7B, (b) Qwen2-7B fine-tuned on Judicial Examination.

on the code and legal domains. As show in Figures 9 and 10, after domain-specific fine-tuning, domain features are embedded into the representation space of general data, resulting in a more unified semantic space. This demonstrates that domain-specific knowledge encoded in the client-side LoRA adapters can be effectively injected into the global adapter through general data during distillation.

E.2 General Ability Experiment

In addition to evaluating general capability retention after domain-specific fine-tuning on the math dataset in Section 4.3, we further conduct the same evaluation on models fine-tuned with code data.

As shown in Figure 11, most baselines exhibit

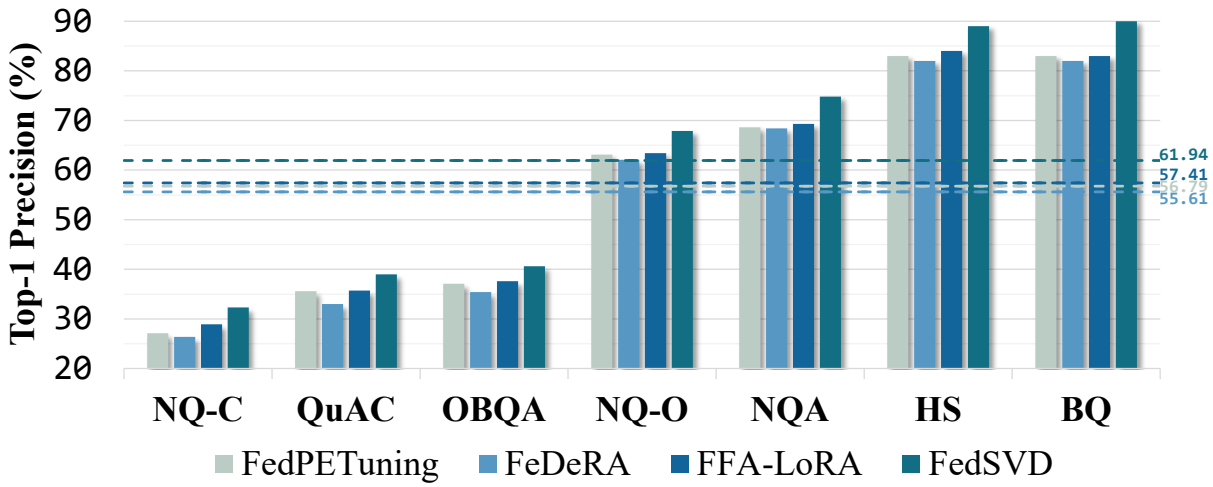


Figure 11: Performance comparison on the HELM benchmark after fine-tuning LLaMA3-8B on Rosetta-Alpaca. The dashed line represents the overall performance.

noticeable performance degradation—particularly FeDeRA. In contrast, OnePeFT consistently outperforms all baselines across tasks. This demonstrates that our distillation strategy not only adapts effectively to the target domain but also alleviates catastrophic forgetting of general capabilities.