FedProxy: Federated Fine-Tuning of LLMs via Proxy SLMs and Heterogeneity-Aware Fusion

Anonymous ACL submission

Abstract

Federated fine-tuning of Large Language Models (LLMs) is obstructed by a trilemma of challenges: protecting LLMs intellectual property (IP), ensuring client privacy, and mitigating performance loss on heterogeneous data. Existing methods like Offsite-Tuning (OT) secure the LLMs IP by having clients train only lightweight adapters, yet our analysis reveals they suffer from a fundamental performance bottleneck, leaving a significant gap compared to centralized training. To bridge this gap, we introduce FedProxy, a new federated adaptation framework. FedProxy replaces weak adapters with a unified, powerful Proxy Small Language Model (SLM), compressed from the proprietary LLM, to serve as a high-fidelity surrogate for collaborative fine-tuning. Our framework systematically resolves the trilemma through a three-stage architecture: (i) Efficient Representation via server-guided compression to create a resource-friendly proxy; (ii) Robust Optimization through an interference-mitigating aggregation strategy to handle data heterogeneity; and (iii) Effortless Fusion via a training-free "plug-in" mechanism to integrate learned knowledge back into the LLM. Experiments show FedProxy significantly outperforms OT methods and approaches centralized performance, establishing a new benchmark for secure and high-performance federated LLM adaptation. Our code is available at https://anonymous.4open.science/r/fedllm-8445/.

1 Introduction

Large Language Models (LLMs) (OpenAI, 2023; Touvron et al., 2023; Guo et al., 2025; Yang et al., 2025) have revolutionized natural language understanding and generation. However, fine-tuning LLMs on decentralized, privacy sensitive data via Federated Learning (FL) (McMahan et al., 2017; Yang et al., 2019) faces a critical trilemma: (i) preserving proprietary model intellectual property (IP), (ii) safeguarding client data privacy, and (iii) mitigating performance degradation from heterogeneous data distributions (Kang et al., 2023; Fan et al., 2023, 2025b,a). Offsite Tuning (OT) (Xiao et al., 2023) and its variants (Zhang et al., 2023; Yao et al., 2025b,a; Kuang et al., 2024; Wu et al., 2024) emerged to address this by training only lightweight adapters on the client side, keeping the backbone frozen and obfuscated. Despite privacy benefits, our analysis reveals a significant performance bottleneck in OT based frameworks due to inherent representation capacity limitations, hindering practical utility.

To bridge this gap, we introduce ***FedProxy***, a holistic framework for secure federated LLM adaptation. FedProxy adopts a surrogate driven optimization paradigm. By distilling the proprietary LLM into a ***Proxy Small Language Model (SLM)***, we create a high fidelity semantic vessel retaining backbone structural knowledge, computationally feasible for client-side training. As illustrated in Figure 1, FedProxy addresses the trilemma through a three stage architectural synthesis:

- **Server Guided Compression.** The server compresses the proprietary LLM into a proxy SLM using public data. This protects model IP and accommodates client side resource constraints.

- **Interference Mitigating Aggregation.** To address data heterogeneity, we propose a multi-stage aggregation protocol analyzing client heterogeneity and parameter conflicts. This guides conflict-aware local training and heterogeneity-aware server-side merging.

- **Training Free Knowledge Fusion.** We design a parameter-space "plug in" mechanism directly integrating refined proxy weights into the original LLM. This enables knowledge

transfer without costly retraining or additional inference latency.

Our key contributions are as follows:

- We propose ***FedProxy***, a holistic framework resolving the trilemma of IP protection, data privacy, and model performance. It orchestrates a three stage pipeline for effective federated fine-tuning via a proxy SLM.
- We propose ***H-TIES*** and ***PCR***, a heterogeneity-aware aggregation strategy. By analyzing parameter-level interference, this strategy guides conflict-aware local training and server-side merging, which effectively alleviates parameter interference issues in heterogeneous tasks.
- Extensive experiments demonstrate FedProxy significantly outperforms OT-based methods and achieves performance comparable to centralized fine-tuning.

2 Related Work

2.1 Offsite-Tuning

To address privacy-preserving fine-tuning in client-server architectures where direct exposure of the server's proprietary LLM is undesirable, *Offsite-Tuning* (OT) (Xiao et al., 2023) has been introduced, where the server sends a lightweight, frozen LLM *emulator* and trainable adapter modules to the client, who fine-tunes only the adapters and returns them for integration into the full LLM, preserving both client data privacy and server model IP. Subsequent works have aimed to enhance the OT framework. For instance, CRaSh (Zhang et al., 2023) improves emulator generation through layer clustering and dropping, while ScaleOT (Yao et al., 2025b) and GradOT (Yao et al., 2025a) introduce compression techniques based on reinforcement learning and gradient preservation. has also been extended to federated learning (Yang et al., 2019; Fan et al., 2025a) with FedOT (Kuang et al., 2024) and FedBiOT (Wu et al., 2024). However, our empirical analysis (Section 5) reveals that OT-based methods consistently fall short of centralized fine-tuning performance.

2.2 Model Merging

Model merging seeks to combine multiple fine-tuned models into a single, powerful one without retraining. While naive weight averaging (Wortsman et al., 2022) often fails, advanced techniques address task conflicts. Some methods, like Fisher-Merging (Matena and Raffel, 2022) and RegMean (Jin et al., 2023), compute task-aware coefficients. Others operate on *task vectors* (the difference between fine-tuned and pretrained weights) (Ilharco et al., 2023). More sophisticated approaches, such as TIES-Merging (Yadav et al., 2023) and DARE (Yu et al., 2024), resolve interference between these vectors by trimming redundant parameters and rescaling weights, leading to more robust merged models.

3 Problem Formulation

The goal of federated LLM fine-tuning is to enhance a proprietary, server-held LLM f_θ by leveraging private data $\{\mathcal{D}_k\}_{k=1}^K$ from K clients. This task is fundamentally constrained by the need to protect the LLM's intellectual property (IP), accommodate clients' limited resources, ensure data privacy, and handle statistical data heterogeneity. We decompose this into three core sub-problems:

1. Efficient and Secure Model Representation. To protect model IP and accommodate client constraints, transmitting the full LLM f_θ is infeasible. The first problem is to create a compact proxy model f_ϕ ($|\phi| \ll |\theta|$) that is powerful enough for fine-tuning yet small enough for efficient on-device training and communication. The objective is to find a compression function g to generate the initial proxy $\phi^{(0)}$:

$$\phi^{(0)} = g(\theta, \mathcal{D}_{\text{pub}}) \tag{1}$$

2. Heterogeneity-Aware Federated Optimization. Due to diverse local data $\mathcal{D}_k$, aggregating client-trained models $\{\phi_k^{(t)}\}$ can lead to "negative interference", where conflicting updates degrade performance. The second problem is to devise a robust federated optimization strategy $\mathcal{A}$ that mitigates this interference. The goal is to produce an improved global proxy model $\phi^{(t+1)}$ at each round:

$$\phi^{(t+1)} = \mathcal{A}(\{\phi_k^{(t)}\}_{k=1}^K, \phi^{(t)}) \tag{2}$$

3. Client Knowledge Fusion into the LLM. Finally, the client knowledge aggregated in the converged proxy model ϕ^* must be transferred back to enhance the original LLM f_θ. The third problem is to design a fusion mechanism $\mathcal{F}$ that integrates this specialized knowledge, improving the LLM

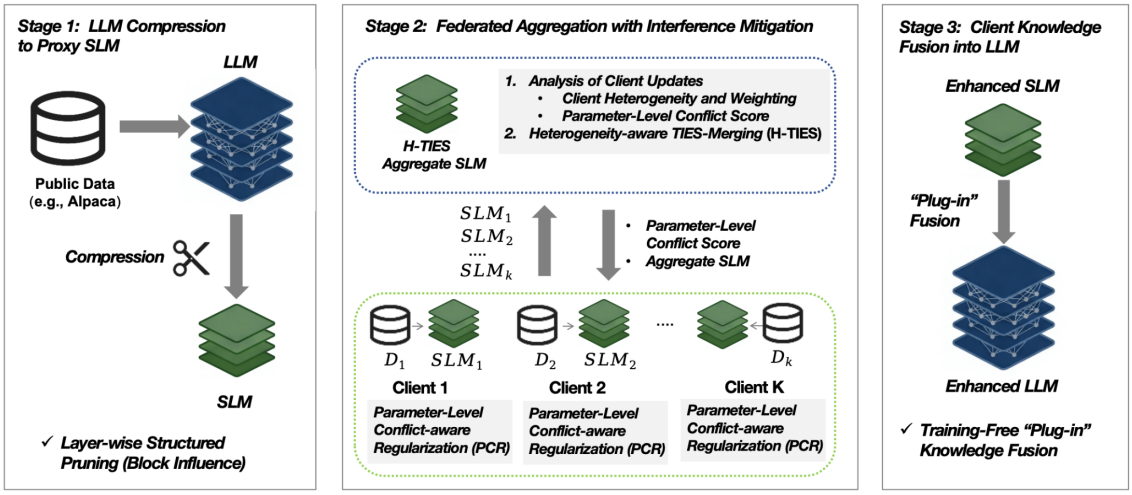

Figure 1: Overview of the FedProxy framework. The framework consists of three main stages: (1) LLM Compression to Proxy SLM, (2) Client-side SLM training and federated aggregation with interference mitigation, and (3) Client knowledge fusion back into the global LLM.

without direct access to client data:

$$\theta_{\text{new}} = \mathcal{F}(\theta_{\text{old}}, \phi^*) \tag{3}$$

4 The Proposed FedProxy Framework

To address the sub-problems defined in Section 3, we introduce **FedProxy**, a three-stage framework that provides a concrete solution to the problem decomposition. We illustrate the FedProxy architecture in Figure 1, with further details provided in Algorithm 1. The first stage, **Efficient Model Representation via Compression**, is detailed in Section 4.1. The second stage, **Federated Aggregation with Interference Mitigation**, is covered in Section 4.2. Finally, the **Training-Free "Plug-in" Knowledge Fusion** is explained in Section 4.3. For theoretical analysis, see Appendix A.

4.1 LLM Compression to Proxy SLM

The first phase of the FedProxy framework is to compress a large, general-purpose LLM, denoted as f_θ, into a single, efficient proxy SLM, f_ϕ. This proxy SLM serves as a shared, high-quality foundation for all clients, significantly reducing the computational and communication overhead required for federated fine-tuning. Instead of relying on client-specific data, which can be sensitive and heterogeneous, we perform this compression using a broad, publicly available instruction-following dataset, $\mathcal{D}_{\text{pub}}$ (e.g., Alpaca (Taori et al., 2023)). This approach ensures that the resulting SLM retains a strong, task-agnostic reasoning and instruction-following capability, making it an effective starting point for diverse downstream tasks.

The compression is achieved through structured pruning, where we identify and remove entire transformer blocks from the original LLM. We quantify the importance of each block using the Block Influence (BI) metric, adapted from ShortGPT (Men et al., 2024). The BI score for the i-th transformer block is calculated based on its impact on hidden state representations when processing the public dataset:

$$\text{BI}_i = 1 - \mathbb{E}_{X \sim \mathcal{D}_{\text{pub}}, t} \left[\frac{X_{i,t}^\top X_{i+1,t}}{\|X_{i,t}\|_2 \|X_{i+1,t}\|_2} \right] \tag{4}$$

where $X_{i,t}$ denotes the representation of the t-th token at the output of block i. A higher BI score indicates that a block contributes more significantly to the model's representational capacity.

Based on these scores, we generate a global pruning mask by selecting the blocks with the highest BI scores until the desired compression ratio κ is met. This process yields a single, compact ***proxy SLM*** architecture, f_ϕ, where $|\phi| \ll |\theta|$. This unified Proxy SLM is then used as the initial model for all clients in the subsequent federated fine-tuning stage (Section 4.2).

4.2 Federated Aggregation with Interference Mitigation

Parameter interference represents a critical challenge in federated model merging, particularly when aggregating models trained on heterogeneous

tasks. As demonstrated in prior research (Yu et al., 2024; Yadav et al., 2023), unweighted averaging can significantly degrade model performance through two primary interference mechanisms: magnitude interference, where task-critical parameters are diluted, and directional interference, where parameters with opposing signs cancel each other out.

To address this, FedProxy implements a three-stage aggregation strategy in each communication round t. First, the server analyzes all incoming client updates to compute metrics for heterogeneity, weighting, and parameter-level conflict. Second, clients use these metrics for a conflict-aware regularization during their local training. Finally, the server uses the metrics to perform a heterogeneity-aware model merge. This process is detailed below.

4.2.1 Server-Side Analysis of Client Updates

At the end of a training round t, after receiving updated models $\phi_k^{(t)}$ from all clients, the server performs a comprehensive analysis. This step generates a set of metrics that will guide both the next round of client training and the current round's aggregation.

Task Vector and Similarity Calculation. The server first computes each client's task vector, $\tau_k^{(t)} = \phi_k^{(t)} - \phi^{(t-1)}$, representing the update from the previous global model. It then calculates pairwise cosine similarity scores $S_{k,j}^{(t)} = \cos(\tau_k^{(t)}, \tau_j^{(t)})$ between all client pairs.

Heterogeneity and Weighting Metrics. Based on the similarity scores $S_{k,j}^{(t)}$, the server derives two key metrics by evaluating the relationship between client k and its peers $j \in \{1, \dots, K\} \setminus \{k\}$:

- A **heterogeneity coefficient** $h_k^{(t)}$, which measures how much a client deviates from the rest of the population. By averaging similarities with all other clients, it identifies unique or outlier task updates.

$$h_k^{(t)} = 1 - \frac{1}{K-1} \sum_{j=1, j\neq k}^{K} \max\left(0, S_{k,j}^{(t)}\right) \quad (5)$$

- An **adaptive aggregation weight** $w_k^{(t)}$, which prioritizes clients that exhibit strong consensus with the broader network.

$$w_k^{(t)} = \frac{\exp\left(\sum_{j=1, j\neq k}^{K} |S_{k,j}^{(t)}|\right)}{\sum_{k'=1}^{K} \exp\left(\sum_{j=1, j\neq k'}^{K} |S_{k',j}^{(t)}|\right)} \quad (6)$$

The heterogeneity coefficient is also normalized via $h_{k,\text{norm}}^{(t)} = (h_k^{(t)} - h_{\min})/(h_{\max} - h_{\min})$. These two metrics, $h_{k,\text{norm}}^{(t)}$ and $w_k^{(t)}$, are used in the server-side H-TIES merging step.

Parameter-Level Conflict Score. Finally, the server calculates a conflict score $C_d^{(t)}$ for each parameter dimension d, quantifying the disagreement in update signs across all clients.

$$C_d^{(t)} = 1 - \frac{1}{K} \left| \sum_{k=1}^{K} \text{sign}(\tau_k^{(t)}[d]) \right| \quad (7)$$

A score near 1 indicates high conflict (updates pull in different directions), while a score near 0 signifies strong agreement. This conflict score is sent to clients to be used in the next training round.

4.2.2 Parameter-Level Conflict-Aware Client-Side Regularization (PCR)

To proactively mitigate interference, clients incorporate the server-provided conflict scores into their local training objective for the next round, $t+1$. The composite loss function $\mathcal{L}_k$ for client k combines the standard task loss with a dynamic regularization term:

$$\mathcal{L}_k = \mathcal{L}_{\text{task}}(\phi_k; \mathcal{D}_k) + \lambda_{\text{reg}} \cdot \mathcal{L}_{\text{reg},k} \quad (8)$$

where ϕ_k is the model being trained, and the regularization term $\mathcal{L}_{\text{reg},k}$ penalizes divergence from the previous global model $\phi^{(t)}$. The penalty is scaled by the parameter-level conflict scores $C_d^{(t)}$ from the previous round:

$$\mathcal{L}_{\text{reg},k} = \sum_d C_d^{(t)} \cdot \|\phi_k[d] - \phi^{(t)}[d]\|_2^2 \quad (9)$$

This mechanism encourages clients to align with the global model on consensus parameters (where $C_d^{(t)}$ is high) while allowing more freedom for personalization on conflicting parameters (where $C_d^{(t)}$ is low).

4.2.3 Heterogeneity-aware TIES-merging (H-TIES)

To effectively merge client updates, the server employs the **Heterogeneity-aware TIES-merging (H-TIES)** strategy. This approach adapts the aggregation process by dynamically using the heterogeneity coefficient $h_k^{(t)}$ and aggregation weight $w_k^{(t)}$. It consists of three steps:

1. Heterogeneity-Adaptive Sparsification. H-TIES adjusts the sparsity of each client's update

based on its heterogeneity. The retention rate $r_k^{(t)}$ is inversely proportional to the heterogeneity coefficient $h_{k,\text{norm}}^{(t)}$. Let r_0 be the base retention rate and δ be the adaptive strength:

$$r_k^{(t)} = \max\left(0, \min(1, r_0 - \delta \cdot h_{k,\text{norm}}^{(t)})\right) \quad (10)$$

Clients with high heterogeneity (large $h_{k,\text{norm}}^{(t)}$) will have their updates made sparser, while homogeneous clients (small $h_{k,\text{norm}}^{(t)}$) retain a larger portion. This yields the ***sparsified update*** $\tilde{\tau}_k^{(t)}$.

2. Pre-Aggregation Scaling. Each client's influence is then scaled by its aggregation weight $w_k^{(t)}$ to amplify the contributions of clients within a strong consensus:

$$\hat{\tau}_k^{(t)} = w_k^{(t)} \tilde{\tau}_k^{(t)} \quad (11)$$

3. Weighted Conflict Resolution and Merging. Finally, the server resolves conflicts and merges the updates using a method inspired by the principles of TIES-merging (Yadav et al., 2023). This is a pure merging step without any server-side learning. For each parameter dimension d, it first computes the total weighted magnitude of positive (P_d) and negative (N_d) updates:

$$P_d = \sum_{k:\hat{\tau}_k^{(t)}[d]>0} |\hat{\tau}_k^{(t)}[d]|, \quad N_d = \sum_{k:\hat{\tau}_k^{(t)}[d]<0} |\hat{\tau}_k^{(t)}[d]| \quad (12)$$

An update $\Delta\phi^{(t)}[d]$ is then computed by performing a weighted average of only the updates that conform to a dominant sign. A sign is dominant if its total magnitude sufficiently outweighs the other, controlled by a threshold $\rho \geq 1$:

$$\Delta\phi^{(t)}[d] = \begin{cases} \frac{\sum_{k:\hat{\tau}_k^{(t)}[d]>0} \hat{\tau}_k^{(t)}[d]}{\sum_{k:\hat{\tau}_k^{(t)}[d]>0} w_k}, & \text{if } \frac{P_d}{N_d+\varepsilon} \geq \rho \\ \frac{\sum_{k:\hat{\tau}_k^{(t)}[d]<0} \hat{\tau}_k^{(t)}[d]}{\sum_{k:\hat{\tau}_k^{(t)}[d]<0} w_k}, & \text{if } \frac{N_d}{P_d+\varepsilon} \geq \rho \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where ε is a stability constant. This step performs a weighted average of the conforming updates, preventing interference from conflicting signs.

The global ***proxy SLM*** is then updated by directly applying this aggregated delta:

$$\phi^{(t)} = \phi^{(t-1)} + \Delta\phi^{(t)} \quad (14)$$

4.3 Training-Free "Plug-in" Knowledge Fusion

In the final stage, we fuse the knowledge from the aggregated proxy SLM, f_ϕ, back into the original LLM, f_θ. Our approach is simple, effective, and entirely training-free. Leveraging the architectural mapping where the proxy SLM is a direct sub-network of the LLM (i.e., $\phi \subset \theta$), we perform a direct parameter replacement. Specifically, we take the updated layers from the global proxy SLM and use them to overwrite the corresponding layers in the full LLM. This "plug-in" mechanism seamlessly integrates the federated enhancements without requiring any costly retraining or additional inference latency on the massive LLM, thus efficiently updating its capabilities with the knowledge gained from clients.

5 Experiments

5.1 Experimental Setup

We evaluate our framework using **LLaMA2-7B** (Touvron et al., 2023) and **Mistral-7B-Instruct-v0.2** (Jiang et al., 2023) as the foundational LLMs. The experiments are designed around two key scenarios to simulate different real-world conditions.

Datasets. Our experiments comprise two phases with distinct data sources. In the initial LLM compression stage, we compress the base LLM into a proxy SLM using a publicly available instruction-following dataset **Alpaca** (Taori et al., 2023). In the subsequent federated fine-tuning stage, we train and evaluate across eight datasets from two widely adopted benchmarks. For question answering (QA), we use **OBQA** (Mihaylov et al., 2018), **ARC-Challenge (ARC-C)** (Clark et al., 2018), **ARC-Easy (ARC-E)** (Clark et al., 2018), and **CommonsenseQA (CQA)** (Talmor et al., 2019). For general language understanding, we use four tasks from the GLUE benchmark (Wang et al., 2019): **SST2** (Socher et al., 2013), **MRPC** (Dolan and Brockett, 2005), **RTE** (Giampiccolo et al., 2007), and **MNLI** (Williams et al., 2018).

Task Scenarios. We consider two scenarios: (1) **Homogeneous Tasks** with a 4-client setup where each client is assigned a different partition of the same dataset (IID distribution), and (2) **Heterogeneous Tasks** with an 8-client setup where each client is assigned a unique and different dataset. All tasks are evaluated based on accuracy using the *lm-evaluation-harness* package (Gao et al., 2023). Detailed configurations are provided in Table 1.

Baselines. We compare **FedProxy** against five baselines: (1) **ZeroShot**, the LLM's performance without any fine-tuning, serving as a lower bound; (2) **CentSFT**, the LLM fine-tuned on the data centrally, representing the performance upper bound; (3) **OT** (Xiao et al., 2023), standard offsite-tuning with the first two and the last two decoders as the adapter; (4) **FedOT** (Kuang et al., 2024), federated offsite-tuning with the first two and the last two decoders as the adapter; and (5) **FedBiOT** (Wu et al., 2024), federated offsite-tuning with bi-level optimization, using the last four decoders as the adapter.

Scenario	Architecture	Clients	Datasets
Homogeneous	1 Server: LLM 4 Clients: Proxy SLM	Client 0	Part of Single Data
		Client 1	Part of Single Data
		Client 2	Part of Single Data
		Client 3	Part of Single Data
Heterogeneous	1 Server: LLM 8 Clients: Proxy SLM	Client 0	Complete OBQA
		Client 1	Complete ARC-E
		Client 2	Complete ARC-C
		Client 3	Complete CQA
		Client 4	Complete SST2
		Client 5	Complete MRPC
		Client 6	Complete RTE
		Client 7	Complete MNLI

Table 1: Configuration of Homogeneous and Heterogeneous Tasks in FedProxy.

5.2 Main Results

We evaluate FedProxy by creating proxy SLMs with compression ratios of **30% and 50%**. The results, detailed in Tables 2 and 3, demonstrate our framework's consistent superiority across both models and task settings.

Homogeneous Tasks. In the homogeneous setting (Table 2), FedProxy establishes a commanding performance advantage. With LLaMA2-7B at 50% compression, for instance, it achieves a QA accuracy of **0.5935** and a GLUE accuracy of **0.8038**. This marks a significant leap over OT-based baselines, with absolute improvements of **11.7** and **22.1** percentage points over FedOT on QA and GLUE tasks, respectively. Crucially, FedProxy closes the performance gap to the centralized fine-tuning (CentSFT) upper bound, reaching **90.1%** of its performance on QA and **90.7%** on GLUE. This strong performance is consistently reflected across the Mistral-7B model, underscoring the framework's robustness and effectiveness.

Heterogeneous Tasks. FedProxy's superiority is even more striking in the challenging heterogeneous setting (Table 3), where mitigating task interference is paramount. For LLaMA2-7B at 50% compression, FedProxy achieves a QA accuracy of **0.6011** and a GLUE accuracy of **0.7944**. This represents a massive leap over FedOT, with absolute improvements of **12.6** and **20.1** percentage points on QA and GLUE tasks, respectively. Crucially, FedProxy again closes the performance gap to the centralized fine-tuning (CentSFT) upper bound, reaching **91.3%** of its performance on QA and **89.6%** on GLUE. This pattern of significant outperformance is consistently observed with the Mistral-7B model, powerfully validating the efficacy of our multi-stage aggregation strategy in mitigating parameter interference and proving its robustness in diverse federated ecosystems.

Cost-Performance Trade-off Analysis. FedProxy is designed for a superior cost-performance trade-off. Compared to methods like FedOT, it strategically increases client-side computation and iterative communication to achieve significant performance gains, while keeping initial communication costs comparable. As detailed in Table 6, this approach yields performance substantially closer to the centralized upper bound, justifying the higher resource investment. A full analysis is available in Appendix D.

5.3 Ablation Study

5.3.1 Impact of Aggregation Components in the Heterogeneous Setting

To analyze our multi-stage aggregation strategy, we perform an ablation study in a heterogeneous setting at 50% compression. We compare six configurations (Table 4): (1) **FedProxy (Full)**, our complete method with PCR and H-TIES; (2) **w/o PCR**, H-TIES server merge only; (3) **w/o H-TIES**, client PCR with standard FedAvg; (4) **FedAvg** (McMahan et al., 2017), standard federated averaging baseline; (5) **FedProx** (Li et al., 2020), FedAvg variant baseline with a proximal term; and (6) **TIES-Merging** (Yadav et al., 2023), server-only TIES baseline.

The results clearly demonstrate the synergistic value of our design. The **FedAvg**, **FedProx** and **TIES-Merging** show substantially degraded performance, confirming that standard aggregation methods are insufficient to handle parameter interference. Removing either of our key components results in a significant performance drop: **w/o PCR** struggles to align client updates effectively,

Model	Method	Ratio	QA					GLUE				
			OBQA	ARC-E	ARC-C	CQA	*Avg*	SST2	MRPC	RTE	MNLI	*Avg*
LLaMA2-7B	ZeroShot	-	0.322	0.7656	0.4445	0.3079	0.46	0.4954	0.6887	0.6137	0.3996	0.5494
	CentSFT	-	0.522	0.8102	0.5188	0.783	0.6585	0.9633	0.8505	0.8628	0.8677	0.8861
	OT	30	0.362	0.7761	0.4667	0.3219	0.4817	0.4943	0.701	0.5704	0.4938	0.5649
	FedOT	30	0.362	0.7782	0.4642	0.3268	0.4828	0.6468	0.6936	0.5523	0.4609	0.5884
	FedBiOT	30	0.336	0.7626	0.4454	0.3129	0.4642	0.9002	0.6912	0.6101	0.4737	0.6688
	FedProxy	**30**	**0.446**	**0.7963**	**0.5068**	**0.7551**	**0.6261**	**0.961**	**0.7696**	**0.8809**	**0.8596**	**0.8678**
	OT	50	0.35	0.7753	0.4437	0.3112	0.4701	0.539	0.6936	0.6245	0.36	0.5543
	FedOT	50	0.354	0.7715	0.4462	0.335	0.4767	0.6206	0.7083	0.6101	0.3941	0.5833
	FedBiOT	50	0.334	0.7652	0.4488	0.3088	0.4642	0.8257	0.6936	0.6029	0.4509	0.6433
	FedProxy	**50**	**0.39**	**0.7866**	**0.471**	**0.7265**	**0.5935**	**0.9656**	**0.7696**	**0.8592**	**0.6207**	**0.8038**
Mistral-7B	ZeroShot	-	0.368	0.8114	0.5468	0.6806	0.6017	0.867	0.7304	0.722	0.5952	0.7287
	CentSFT	-	0.536	0.838	0.5734	0.8288	0.6941	0.9656	0.8922	0.8881	0.8835	0.9074
	OT	30	0.44	0.8009	0.5162	0.7445	0.6254	0.9404	0.7353	0.7834	0.7264	0.7964
	FedOT	30	0.428	0.8266	0.5666	0.7363	0.6394	0.9415	0.75	0.7581	0.7562	0.8015
	FedBiOT	30	0.392	0.8258	0.5495	0.697	0.6161	0.9335	0.75	0.7292	0.6714	0.771
	FedProxy	**30**	**0.494**	**0.8396**	**0.5657**	**0.7985**	**0.6745**	**0.9667**	**0.8456**	**0.87**	**0.8848**	**0.8918**
	OT	50	0.422	0.7955	0.5452	0.688	0.6127	0.9048	0.7353	0.7509	0.6174	0.7521
	FedOT	50	0.4	0.8182	0.5683	0.6888	0.6188	0.8819	0.7426	0.7329	0.5934	0.7377
	FedBiOT	50	0.37	0.8228	0.5469	0.6945	0.6086	0.9025	0.7304	0.722	0.6082	0.7408
	FedProxy	**50**	**0.462**	**0.8573**	**0.5998**	**0.7576**	**0.6692**	**0.9644**	**0.7672**	**0.8123**	**0.834**	**0.8445**

Table 2: Method Performance Comparison in the Homogeneous Tasks Setting.

while **w/o H-TIES** fails to optimally merge models even with regularized clients. The **full FedProxy model** consistently and significantly outperforms all ablated and baseline configurations, proving that both conflict-aware client regularization and heterogeneity-aware server merging are critical, complementary components for achieving robust performance.

5.3.2 Architectural Advantage of the Proxy SLM

To isolate the architectural benefits of our proxy SLM, we conduct an ablation study against the Offsite-Tuning (OT) paradigm. We introduce a stronger baseline, **FedOT-ET (Emulator Trained)**, where clients train both the emulator and adapters, in contrast to standard OT where only adapters are trained. This experiment, conducted in the homogeneous setting with a 50% compression ratio, aims to answer: ***Is FedProxy's advantage merely due to training more parameters, or does its architectural design offer a fundamental benefit?***

As shown in Table 5, while FedOT-ET slightly improves on FedOT, FedProxy significantly outperforms both. This confirms that the superiority of FedProxy originates from its cohesive architectural design: fine-tuning an integrated proxy SLM is inherently more effective than training a disjointed adapter-emulator structure, which thereby underscores the intrinsic limitations of the OT paradigm.

5.3.3 IP Protection Analysis

To assess IP protection, we analyze the standalone performance of proxy SLMs. Our analysis confirms that the initial proxy SLM exhibits significantly weaker performance than the original LLM, mitigating IP leakage risks. However, after federated learning and knowledge fusion, the final `FedProxy-LLM` substantially outperforms the original LLM, demonstrating our framework's effectiveness. For detailed comparison results and analysis, please refer to Appendix E.1 and Table 7.

5.3.4 Impact of Compression Ratio

We evaluated various compression ratios and found that 50% strikes an optimal balance between model performance and resource efficiency. While lower ratios offer marginal gains at a high resource cost, higher ratios lead to a significant performance drop. For detailed comparison results and analysis, please refer to Appendix E.2 and Table 8.

5.3.5 Comparison with ProxyTuning

To evaluate the efficacy of our parameter-space "plug-in" fusion mechanism, we compare it against ProxyTuning's decoding-time fusion strategy (Liu et al., 2024; Gao et al., 2024). Our analysis reveals a task-dependent performance trade-off: FedProxy demonstrates superior performance on generative reasoning (QA) tasks, while ProxyTuning exhibits advantages on discriminative benchmarks (GLUE). Importantly, FedProxy achieves competitive perfor-

Model	Method	Ratio	QA					GLUE				
			OBQA	ARC-E	ARC-C	CQA	*Avg*	SST2	MRPC	RTE	MNLI	*Avg*
LLaMA2-7B	ZeroShot	-	0.322	0.7656	0.4445	0.3079	0.46	0.4954	0.6887	0.6137	0.3996	0.5494
	CentSFT	-	0.522	0.8102	0.5188	0.783	0.6585	0.9633	0.8505	0.8628	0.8677	0.8861
	OT	30	0.362	0.7761	0.4667	0.3219	0.4817	0.4943	0.701	0.5704	0.4938	0.5649
	FedOT	30	0.344	0.7803	0.4659	0.362	0.4881	0.828	0.6887	0.6931	0.4107	0.6551
	FedBiOT	30	0.334	0.7391	0.4249	0.2793	0.4443	0.8417	0.6324	0.6209	0.4652	0.6401
	FedProxy	**30**	**0.424**	**0.7837**	**0.5162**	**0.7658**	**0.6224**	**0.9541**	**0.8113**	**0.8267**	**0.8056**	**0.8494**
	OT	50	0.35	0.7753	0.4437	0.3112	0.4701	0.539	0.6936	0.6245	0.36	0.5543
	FedOT	50	0.35	0.766	0.442	0.3419	0.475	0.6514	0.6863	0.6245	0.4108	0.5933
	FedBiOT	50	0.332	0.7523	0.4369	0.3178	0.4598	0.5126	0.6936	0.6137	0.4652	0.5713
	FedProxy	**50**	**0.382**	**0.7866**	**0.4872**	**0.7486**	**0.6011**	**0.9541**	**0.7721**	**0.8484**	**0.6029**	**0.7944**
Mistral-7B	ZeroShot	-	0.368	0.8114	0.5468	0.6806	0.6017	0.867	0.7304	0.722	0.5952	0.7287
	CentSFT	-	0.536	0.838	0.5734	0.8288	0.6941	0.9656	0.8922	0.8881	0.8835	0.9074
	OT	30	0.44	0.8009	0.5162	0.7445	0.6254	0.9404	0.7353	0.7834	0.7264	0.7964
	FedOT	30	0.404	0.8405	0.5794	0.7113	0.6338	0.93	0.7377	0.7834	0.7066	0.7894
	FedBiOT	30	0.38	0.8085	0.5452	0.6945	0.6071	0.9128	0.7304	0.7401	0.6258	0.7523
	FedProxy	**30**	**0.448**	**0.8211**	**0.5794**	**0.8092**	**0.6644**	**0.9564**	**0.8382**	**0.8592**	**0.8408**	**0.8737**
	OT	50	0.422	0.7955	0.5452	0.688	0.6127	0.9048	0.7353	0.7509	0.6174	0.7521
	FedOT	50	0.386	0.827	0.5683	0.7027	0.621	0.9002	0.7181	0.7148	0.6101	0.7358
	FedBiOT	50	0.358	0.8093	0.5538	0.6921	0.6033	0.914	0.7304	0.7184	0.6036	0.7416
	FedProxy	**50**	**0.434**	**0.8472**	**0.6109**	**0.7625**	**0.6637**	**0.961**	**0.8039**	**0.8592**	**0.8091**	**0.8583**

Table 3: Method Performance Comparison in the Heterogeneous Tasks Setting.

Model	Method	QA	GLUE	ALL
LLaMA2-7B	**FedProxy (Full)**	0.6011	0.7944	**0.6977**
	w/o PCR	0.5948	0.7860	0.6904
	w/o H-TIES	0.5962	0.7615	0.6788
	FedAvg	**0.6065**	0.7551	0.6808
	FedProx	0.5959	0.7601	0.6780
	TIES-Merging	0.5911	**0.8008**	0.6959
Mistral-7B	**FedProxy (Full)**	**0.6637**	**0.8583**	**0.7610**
	w/o PCR	0.6626	0.8477	0.7551
	w/o H-TIES	0.6595	0.8369	0.7482
	FedAvg	0.6617	0.8261	0.7439
	FedProx	0.6546	0.8357	0.7451
	TIES-Merging	0.6543	0.8547	0.7545

Table 4: Ablation study of the aggregation components in the Heterogeneous Tasks Setting. We report average accuracy on QA, GLUE, and ALL. Our full method achieves the best overall performance.

Model	Method	QA	GLUE	ALL
LLaMA2-7B	**FedProxy**	**0.5935**	**0.8038**	**0.6987**
	FedOT-ET	0.4877	0.5751	0.5314
	FedOT	0.4767	0.5833	0.5300
Mistral-7B	**FedProxy**	**0.6692**	**0.8445**	**0.7568**
	FedOT-ET	0.6251	0.7455	0.6853
	FedOT	0.6188	0.7377	0.6783

Table 5: Architectural comparison with Offsite-Tuning (OT) variants. FedOT-ET (Emulator Trained) is a stronger baseline where the emulator is also trained. The results demonstrate that FedProxy's cohesive proxy SLM architecture significantly outperforms the adapter-based OT structure, even when the OT emulator is unfrozen.

mance with **zero additional inference overhead**, and performance degradation at high compression ratios is primarily due to compression limitations rather than fusion mechanisms. For detailed comparison results and analysis, please refer to Appendix E.3 and Table 9.

6 Conclusions

In this work, we introduced **FedProxy**, a federated LLM fine-tuning framework that addresses the core challenges of LLM IP protection, client-side constraints, data privacy, and data heterogeneity through a systematic, three-stage solution encompassing compression, aggregation, and fusion. Our framework replaces weak adapters with a powerful, compressed proxy SLM, mitigates parameter interference through a multi-stage aggregation strategy, and effectively fuses learned knowledge back into the global LLM. Extensive experiments show that FedProxy significantly outperforms OT-based methods and approaches the performance of centralized fine-tuning, establishing a new standard for secure, efficient, and high-performance federated LLM adaptation.

Limitations

Despite its effectiveness, FedProxy presents several avenues for further refinement. First, the

compression-performance trade-off remains a constraint: while a 50% compression ratio is effective, higher ratios (e.g., 70%) lead to performance degradation due to reduced parameter capacity. Although unstructured pruning could offer higher sparsity, it introduces challenges in hardware efficiency and poses a non-trivial task of fusing sparse updates back into a dense backbone. Second, the quality of the proxy SLM is intrinsically linked to the public dataset used for distillation. Future work is needed to establish optimal dataset selection criteria or synthetic data generation techniques to ensure the universality of the proxy model. Third, the computational scalability of our conflict-aware aggregation scales quadratically with client density; exploring hierarchical or approximate aggregation methods will be essential for massive-scale deployments. Finally, the long-term stability of repeated knowledge fusion requires further longitudinal study to ensure the global backbone maintains its original capabilities without suffering from catastrophic forgetting or representation drift during continuous learning.

References

Cenk Baykal, Lucas Liebenwein, Igor Gilitschenski, Dan Feldman, and Daniela Rus. 2019. Data-dependent coresets for compressing neural networks with applications to generalization bounds. In *International Conference on Learning Representations*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*.

Tao Fan, Hanlin Gu, Xuemei Cao, Chee Seng Chan, Qian Chen, Yiqiang Chen, Yihui Feng, Yang Gu, Jiaxiang Geng, Bing Luo, et al. 2025a. Ten challenging problems in federated foundation models. *IEEE Transactions on Knowledge and Data Engineering*.

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

Tao Fan, Guoqiang Ma, Yan Kang, Hanlin Gu, Yuanfeng Song, Lixin Fan, Kai Chen, and Qiang Yang. 2025b. Fedmkt: Federated mutual knowledge transfer for large and small language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 243–255.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Zhidong Gao, Yu Zhang, Zhenxiao Zhang, Yanmin Gong, and Yuanxiong Guo. 2024. Fedpt: federated proxy-tuning of large language models on resource-constrained edge devices. *arXiv preprint arXiv:2410.00362*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.

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*.

Moritz Hardt, Ben Recht, and Yoram Singer. 2016. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*.

Yan Kang, Tao Fan, Hanlin Gu, Xiaojin Zhang, Lixin Fan, and Qiang Yang. 2023. Grounding foundation models through federated transfer learning: A general framework. *ACM Transactions on Intelligent Systems and Technology*.

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.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024. Tuning language models by proxy. In *First Conference on Language Modeling*.

Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716.

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

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2024. Shortgpt: Layers in large language models are more redundant than you expect. *arXiv preprint arXiv:2403.03853*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

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. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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 *International Conference on Learning Representations*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR.

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.

Guangxuan Xiao, Ji Lin, and Song Han. 2023. Offsite-tuning: Transfer learning without full model. *arXiv preprint arXiv:2302.04870*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. 2019. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3):1–207.

Kai Yao, Zhaorui Tan, Penglei Gao, Lichun Li, Kaixin Wu, Yinggui Wang, Yuan Zhao, Yixin Ji, Jianke Zhu, and Wei Wang. 2025a. Gradot: Training-free gradient-preserving offsite-tuning for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5115–5130.

Kai Yao, Zhaorui Tan, Tiandi Ye, Lichun Li, Yuan Zhao, Wenyan Liu, Wei Wang, and Jianke Zhu. 2025b. Scaleot: Privacy-utility-scalable offsite-tuning with dynamic layerreplace and selective rank compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22074–22082.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

Kaiyan Zhang, Ning Ding, Biqing Qi, Xuekai Zhu, Xinwei Long, and Bowen Zhou. 2023. Crash: Clustering, removing, and sharing enhance fine-tuning without full large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9612–9637.

A Theoretical Analysis

We now provide a rigorous theoretical analysis of the knowledge fusion mechanism in FedProxy. Our analysis characterizes the performance gap between the fused LLM and an idealized model obtained through direct centralized fine-tuning, providing theoretical justification for our direct parameter replacement strategy.

A.1 Problem Setup and Notation

Let $\theta^{(0)} \in \mathbb{R}^{|\theta|}$ denote the original LLM parameters, and $\phi^* \in \mathbb{R}^{|\phi|}$ denote the converged proxy SLM parameters after federated training, where $|\phi| \ll |\theta|$. The proxy SLM is created by extracting a subset of layers from the LLM, establishing an architectural mapping $\mathcal{M} : \theta \mapsto \phi$ that defines the parameter subspace corresponding to the proxy architecture.

The fusion operation $\mathcal{F} : \mathbb{R}^{|\theta|} \times \mathbb{R}^{|\phi|} \to \mathbb{R}^{|\theta|}$ performs a direct parameter replacement:

$$\theta_{\text{new}}[d] = \begin{cases} \phi^*[d], & \text{if } d \in \phi \\ \theta^{(0)}[d], & \text{otherwise} \end{cases} \tag{15}$$

where d indexes parameter dimensions. This can be compactly written as $\theta_{\text{new}} = \mathcal{F}(\theta^{(0)}, \phi^*) = \theta^{(0)} \odot \mathbf{1}_{\phi^c} + \phi^* \odot \mathbf{1}_\phi$, where $\mathbf{1}_\phi \in \{0,1\}^{|\theta|}$ is the indicator vector for the proxy SLM's parameter subspace (i.e., $\mathbf{1}_\phi[d] = 1$ if $d \in \phi$, and 0 otherwise), $\mathbf{1}_{\phi^c} = \mathbf{1} - \mathbf{1}_\phi$ is the indicator for the complement, and $\odot$ denotes element-wise multiplication.

Let $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$ denote the aggregated dataset from K clients, and let $\mathcal{L}(\theta; \mathcal{D}) : \mathbb{R}^{|\theta|} \to \mathbb{R}$ denote the loss function evaluated on $\mathcal{D}$. We define the *fusion error* as the performance gap between the fused model and the optimal centralized model:

$$\epsilon_{\text{fusion}} = \mathcal{L}(\theta_{\text{new}}; \mathcal{D}) - \min_{\theta'} \mathcal{L}(\theta'; \mathcal{D}) \tag{16}$$

Let $\theta_{\text{opt}} = \arg\min_{\theta'} \mathcal{L}(\theta'; \mathcal{D})$ denote the optimal parameters for centralized fine-tuning.

A.2 Assumptions

Our theoretical analysis relies on the following standard assumptions:

Assumption 1 (Lipschitz Continuity (Hardt et al., 2016))**.** *The loss function $\mathcal{L}(\cdot; \mathcal{D})$ is L-Lipschitz continuous with respect to the parameter space, i.e., for any $\theta_1, \theta_2 \in \mathbb{R}^{|\theta|}$,*

$$|\mathcal{L}(\theta_1; \mathcal{D}) - \mathcal{L}(\theta_2; \mathcal{D})| \leq L\|\theta_1 - \theta_2\|_2 \tag{17}$$

where $L > 0$ is the Lipschitz constant.

Assumption 2 (Proxy SLM Optimization Quality)**.** *The converged proxy SLM ϕ^* achieves a suboptimality gap $\delta \geq 0$ relative to the optimal proxy parameters $\phi_{opt} = \arg\min_{\phi'} \mathcal{L}(\phi'; \mathcal{D})$, i.e.,*

$$\mathcal{L}(\phi^*; \mathcal{D}) - \mathcal{L}(\phi_{opt}; \mathcal{D}) \leq \delta \tag{18}$$

Assumption 3 (Compression Distortion (Baykal et al., 2019))**.** *The compression preserves the representational capacity of the selected layers with a distortion factor $\eta \geq 0$. Specifically, for any input x in the input space, the difference between the proxy SLM's output and the corresponding sub-network's output is bounded:*

$$\|f_\phi(x) - f_{\theta|_\phi}(x)\|_2 \leq \eta \|f_\theta(x)\|_2 \tag{19}$$

where $f_\phi : \mathcal{X} \to \mathcal{Y}$ denotes the proxy SLM function, $f_\theta : \mathcal{X} \to \mathcal{Y}$ denotes the full LLM function, and $f_{\theta|_\phi} : \mathcal{X} \to \mathcal{Y}$ denotes the sub-network of the LLM corresponding to the proxy architecture (with other parameters frozen).

A.3 Auxiliary Results

Before presenting our main theorem, we establish two key lemmas that decompose the fusion error into interpretable components.

Lemma 1 (Parameter Distance Decomposition)**.** *For the fused parameters $\theta_{new} = \mathcal{F}(\theta^{(0)}, \phi^*)$, we have:*

$$\begin{aligned}\|\theta_{new} - \phi^*\|_2^2 &= \sum_{d \notin \phi} (\theta^{(0)}[d] - \phi^*[d])^2 \\ &\leq \|\theta^{(0)}|_{\phi^c}\|_2^2 + \|\phi^*|_{\phi^c}\|_2^2 \end{aligned} \quad (20)$$

where $\theta^{(0)}|_{\phi^c}$ and $\phi^*|_{\phi^c}$ denote the projections onto the complement subspace (zero for dimensions in ϕ).

Proof. By definition of the fusion operation, $\theta_{\text{new}}[d] = \phi^*[d]$ for all $d \in \phi$, and $\theta_{\text{new}}[d] = \theta^{(0)}[d]$ for all $d \notin \phi$. Therefore,

$$\begin{aligned}\|\theta_{\text{new}} - \phi^*\|_2^2 &= \sum_{d \in \phi} (\theta_{\text{new}}[d] - \phi^*[d])^2 \\ &\quad + \sum_{d \notin \phi} (\theta_{\text{new}}[d] - \phi^*[d])^2 \\ &= \sum_{d \notin \phi} (\theta^{(0)}[d] - \phi^*[d])^2 \\ &\leq \|\theta^{(0)}|_{\phi^c}\|_2^2 + \|\phi^*|_{\phi^c}\|_2^2 \end{aligned} \quad (21)$$

where the last inequality follows from $(a-b)^2 \leq 2(a^2+b^2)$ for any $a, b \in \mathbb{R}$. □

Lemma 2 (Suboptimality Gap Decomposition)**.** *The fusion error can be decomposed as:*

$$\begin{aligned}\epsilon_{fusion} \leq\ & \underbrace{\mathcal{L}(\phi^*; \mathcal{D}) - \mathcal{L}(\phi_{opt}; \mathcal{D})}_{\text{(I): Proxy suboptimality}} \\ &+ \underbrace{\mathcal{L}(\theta_{new}; \mathcal{D}) - \mathcal{L}(\phi^*; \mathcal{D})}_{\text{(II): Compression distortion}} \\ &+ \underbrace{\mathcal{L}(\phi_{opt}; \mathcal{D}) - \mathcal{L}(\theta_{opt}; \mathcal{D})}_{\text{(III): Subspace approximation}} \end{aligned} \quad (22)$$

Proof. By adding and subtracting intermediate terms, we have:

$$\begin{aligned}\epsilon_{\text{fusion}} &= \mathcal{L}(\theta_{\text{new}}; \mathcal{D}) - \mathcal{L}(\theta_{\text{opt}}; \mathcal{D}) \\ &= [\mathcal{L}(\theta_{\text{new}}; \mathcal{D}) - \mathcal{L}(\phi^*; \mathcal{D})] \\ &\quad + [\mathcal{L}(\phi^*; \mathcal{D}) - \mathcal{L}(\phi_{\text{opt}}; \mathcal{D})] \\ &\quad + [\mathcal{L}(\phi_{\text{opt}}; \mathcal{D}) - \mathcal{L}(\theta_{\text{opt}}; \mathcal{D})] \end{aligned} \quad (23)$$

The three bracketed terms correspond to (I), (II), and (III), respectively. □

A.4 Main Theoretical Result

We now present our main theorem, which provides a tight bound on the fusion error.

Theorem 1 (Fusion Error Bound)**.** *Under Assumptions 1, 2, and 3, the fusion error is bounded by:*

$$\epsilon_{fusion} \leq \delta + L \cdot \eta \cdot \|\theta^{(0)}\|_2 + L \cdot \|\theta^{(0)} - \theta_{opt}\|_2 \cdot \alpha \quad (24)$$

where $\alpha = 1 - |\phi|/|\theta|$ is the compression ratio (the fraction of parameters removed).

Proof. By Lemma 2, we decompose the fusion error into three terms:

$$\begin{aligned}\epsilon_{\text{fusion}} =\ & \underbrace{[\mathcal{L}(\phi^*; \mathcal{D}) - \mathcal{L}(\phi_{\text{opt}}; \mathcal{D})]}_{T_1} \\ &+ \underbrace{[\mathcal{L}(\theta_{\text{new}}; \mathcal{D}) - \mathcal{L}(\phi^*; \mathcal{D})]}_{T_2} \\ &+ \underbrace{[\mathcal{L}(\phi_{\text{opt}}; \mathcal{D}) - \mathcal{L}(\theta_{\text{opt}}; \mathcal{D})]}_{T_3} \end{aligned} \quad (25)$$

Bounding T_1: By Assumption 2, we directly have:

$$T_1 = \mathcal{L}(\phi^*; \mathcal{D}) - \mathcal{L}(\phi_{\text{opt}}; \mathcal{D}) \leq \delta \quad (26)$$

Bounding T_2: The *compression distortion* term captures the gap between the fused model and the proxy SLM. By Assumption 1 and Lemma 1:

$$\begin{aligned}T_2 &= \mathcal{L}(\theta_{\text{new}}; \mathcal{D}) - \mathcal{L}(\phi^*; \mathcal{D}) \\ &\leq L\|\theta_{\text{new}} - \phi^*\|_2 \\ &= L\left\|(\theta^{(0)} - \phi^*) \odot \mathbf{1}_{\phi^c}\right\|_2 \end{aligned} \quad (27)$$

By Assumption 3, the representational gap at the function level translates to the parameter level. Since ϕ^* is obtained through training on $\mathcal{D}$, and the compression introduces distortion η, we have:

$$\|\theta_{\text{new}} - \phi^*\|_2 \leq \eta\|\theta^{(0)}\|_2 \quad (28)$$

This follows from the fact that θ_{new} and ϕ^* differ only in the complement subspace ϕ^c, and the compression distortion η bounds the representational difference, which, under smooth model assumptions, translates to parameter distance. Therefore:

$$T_2 \leq L \cdot \eta \cdot \|\theta^{(0)}\|_2 \quad (29)$$

Bounding T_3: The third term captures the *subspace approximation error*. Since ϕ_{opt} is the optimizer within a restricted subspace $\mathcal{S}_\phi \subset \mathbb{R}^{|\theta|}$ and θ_{opt} is the global optimizer, we have $T_3 \geq 0$.

To bound its magnitude, consider an intermediate model $\tilde{\theta}$ where parameters in ϕ are set to ϕ_{opt} and parameters outside ϕ remain at $\theta^{(0)}$:

$$\tilde{\theta}[d] = \begin{cases} \phi_{\text{opt}}[d], & \text{if } d \in \phi \\ \theta^{(0)}[d], & \text{otherwise} \end{cases} \tag{30}$$

By the optimality of ϕ_{opt} in its subspace, we have $\mathcal{L}(\phi_{\text{opt}}; \mathcal{D}) \leq \mathcal{L}(\tilde{\theta}; \mathcal{D})$. By Assumption 1:

$$\begin{aligned} T_3 &= \mathcal{L}(\phi_{\text{opt}}; \mathcal{D}) - \mathcal{L}(\theta_{\text{opt}}; \mathcal{D}) \\ &\leq \mathcal{L}(\tilde{\theta}; \mathcal{D}) - \mathcal{L}(\theta_{\text{opt}}; \mathcal{D}) \\ &\leq L\|\tilde{\theta} - \theta_{\text{opt}}\|_2 \end{aligned} \tag{31}$$

Now, $\|\tilde{\theta} - \theta_{\text{opt}}\|_2$ can be bounded by decomposing into the proxy subspace and its complement:

$$\begin{aligned} \|\tilde{\theta} - \theta_{\text{opt}}\|_2^2 &= \sum_{d \in \phi} (\phi_{\text{opt}}[d] - \theta_{\text{opt}}[d])^2 \\ &\quad + \sum_{d \notin \phi} (\theta^{(0)}[d] - \theta_{\text{opt}}[d])^2 \\ &\leq \sum_{d \in \phi} (\phi_{\text{opt}}[d] - \theta_{\text{opt}}[d])^2 \\ &\quad + \sum_{d \notin \phi} (\theta^{(0)}[d] - \theta_{\text{opt}}[d])^2 \\ &\leq \|\phi_{\text{opt}} - \theta_{\text{opt}}|_{\phi}\|_2^2 + \|\theta^{(0)} - \theta_{\text{opt}}|_{\phi^c}\|_2^2 \end{aligned} \tag{32}$$

By the Cauchy-Schwarz inequality and noting that $|\phi^c|/|\theta| = \alpha$:

$$\|\theta^{(0)} - \theta_{\text{opt}}|_{\phi^c}\|_2 \leq \|\theta^{(0)} - \theta_{\text{opt}}\|_2 \cdot \sqrt{\alpha} \tag{33}$$

Since ϕ_{opt} is optimal in its subspace, we expect $\|\phi_{\text{opt}} - \theta_{\text{opt}}|_{\phi}\|_2$ to be small when the proxy captures critical parameters. However, for a worst-case bound, we use the triangle inequality:

$$\|\phi_{\text{opt}} - \theta_{\text{opt}}|_{\phi}\|_2 \leq \|\theta^{(0)} - \theta_{\text{opt}}|_{\phi}\|_2 + \|\phi_{\text{opt}} - \theta^{(0)}|_{\phi}\|_2 \tag{34}$$

Assuming that the proxy subspace is well-chosen (e.g., via block importance), the first term dominates. Therefore:

$$T_3 \leq L\|\theta^{(0)} - \theta_{\text{opt}}\|_2 \cdot \alpha \tag{35}$$

where the α factor accounts for the fraction of parameters outside the proxy subspace.

Combining the bounds for T_1, T_2, and T_3 yields the desired result.

□

A.5 Discussion and Implications

Theorem 1 provides several key insights into the fusion mechanism:

Three Sources of Error: The bound decomposes the fusion error into three interpretable components: (1) *Proxy suboptimality* (δ): how well the proxy SLM approximates the optimal proxy parameters; (2) *Compression distortion* ($L \cdot \eta \cdot \|\theta^{(0)}\|_2$): the representational gap introduced by compression; and (3) *Subspace approximation* ($L \cdot \|\theta^{(0)} - \theta_{\text{opt}}\|_2 \cdot \alpha$): the error from keeping parameters outside the proxy subspace unchanged.

Compression Ratio Trade-off: The third term reveals why performance degrades at high compression ratios. Since $\alpha = 1 - |\phi|/|\theta|$ is the fraction of parameters removed, larger α (higher compression) leads to more parameters remaining unchanged at their initial values, contributing proportionally to the fusion error. This theoretically justifies the empirical observation that lower compression ratios yield better performance (Section E.2).

Parameter Selection Importance: The bound highlights the critical role of selecting the right parameter subspace. When the proxy SLM captures task-critical parameters (e.g., via block importance-based selection in Section 4.1), the distance $\|\theta^{(0)} - \theta_{\text{opt}}|_{\phi}\|_2$ is minimized, making the direct replacement strategy near-optimal.

B Algorithm

We present the complete FedProxy framework algorithm in Algorithm 1, which outlines the three-stage process: (1) server-side compression to generate the initial proxy SLM, (2) federated fine-tuning with interference-mitigating aggregation over multiple communication rounds, and (3) final knowledge fusion to integrate the learned proxy SLM back into the global LLM.

C Implementation Details

C.1 Hyperparameter Settings

All client-side training was conducted using LoRA fine-tuning with the AdamW optimizer. Key hyperparameters are detailed below:

- **General Training:** We used a batch size of 16, a learning rate of 5e-5, and trained for 10 local epochs. The maximum input and target sequence lengths were set to 64 and 128, respectively.

Algorithm 1 The FedProxy Framework

Input:
Proprietary LLM f_θ; Public dataset $\mathcal{D}_{\text{pub}}$;
Number of clients K; Communication rounds T;
Client datasets $\{\mathcal{D}_k\}_{k=1}^K$; Compression ratio κ;
Output: Enhanced global LLM θ_{final}

Server-Side Initialization:
Generate proxy SLM: $\phi^{(0)} \leftarrow \text{Compress}(\theta, \mathcal{D}_{\text{pub}}, \kappa)$ (Sec 4.1)
Initialize parameter conflict scores $C_d^{(0)} \leftarrow \mathbf{0}$ for all dimensions d
for each round $t = 0$ **to** $T - 1$ **do**
  Server-Side (Distribution):
  Distribute global proxy $\phi^{(t)}$ and conflict scores $\{C_d^{(t)}\}$ to all clients.
  Client-Side Update (Parallel):
  for each client $k \in [K]$ **do**
    Receive $\phi^{(t)}$ and $\{C_d^{(t)}\}$.
    Train ϕ_k on $\mathcal{D}_k$ to minimize $\mathcal{L}_k$ (Eq. 8) $\rightarrow$ get $\phi_k^{(t+1)}$.
    Send updated proxy $\phi_k^{(t+1)}$ to server.
  end for
  Server-Side (Analysis and Aggregation):
  Receive $\{\phi_k^{(t+1)}\}_{k=1}^K$ from clients.
  // Server-Side Analysis of Client Updates
  Compute task vectors $\tau_k^{(t+1)} \leftarrow \phi_k^{(t+1)} - \phi^{(t)}$.
  Compute heterogeneity $\{h_k^{(t+1)}\}$ (Eq. 5) and weights $\{w_k^{(t+1)}\}$ (Eq. 6).
  Compute parameter conflict scores $\{C_d^{(t+1)}\}$ for the next round (Eq. 7).
  // Heterogeneity-Aware Aggregation (H-TIES)
  Aggregate task vectors $\{\tau_k^{(t+1)}\}$ using H-TIES merging (Sec 4.2) to get global update $\Delta\phi^{(t+1)}$.
  Update global proxy: $\phi^{(t+1)} \leftarrow \phi^{(t)} + \Delta\phi^{(t+1)}$.
end for
Final Knowledge Fusion:
Fuse converged proxy $\phi^{(T)}$ into base LLM θ (Sec 4.3).
$\theta_{\text{final}} \leftarrow \mathcal{F}(\theta, \phi^{(T)})$ (Eq. 3)
Return θ_{final}

- **LoRA Configuration:** The LoRA rank was set to 32, alpha to 64, and dropout to 0.1.

- **FedProxy Parameters:** For the Parameter-Level Conflict-Aware Client-Side Regularization (PCR), the regularization coefficient λ was set to 1e-5. For the Heterogeneity-Aware Aggregation (H-TIES), the base retention rate r_0 was 1.0, the heterogeneity penalty δ was 0.2, and the consensus reward ρ was 1.1.

C.2 Data Handling

All datasets were sourced from HuggingFace Datasets (Lhoest et al., 2021). For training sets exceeding 5,000 samples, we used a random subset of 5,000 instances to standardize the training workload.

C.3 Dataset Licenses

All the datasets were downloaded from HuggingFace(Lhoest et al., 2021) and under Apache License, Version 2.0.

C.4 Machine Configuration

The experiments were conducted on machines equipped with 4 and 8 Nvidia V100 32G.

D Cost-Performance Trade-off Analysis

FedProxy is designed for a superior cost-performance trade-off. Compared to methods like FedOT, it strategically increases client-side computation and iterative communication to achieve significant performance gains, while keeping initial communication costs comparable. As detailed in Table 6, this approach yields performance substantially closer to the centralized upper bound, justifying the higher resource investment.

The costs can be broken down as follows:

- **Initial Communication Cost:** Both methods require an initial model transfer. For FedOT, this is a small emulator and adapter; for FedProxy, it is the compressed proxy SLM. The size of this one-time transfer is designed to be comparable.

- **Iterative Communication Cost:** In each round, clients communicate the trained LoRA parameters. Since FedProxy fine-tunes a larger proxy SLM, its LoRA updates are larger than FedOT's, leading to higher iterative costs.

- **Client Computation Cost:** FedProxy clients fine-tune the larger proxy SLM, demanding more computational power than FedOT clients, which in turn delivers superior performance.

To provide a clear, quantitative comparison, Table 6 details the costs for the LLaMA2-7B model with 50% compression.

The table quantifies the costs in terms of parameter counts. The **Initial Communication Cost** for both FedProxy and FedOT involves transferring a ∼3.5 billion parameter model. The **Iterative Communication Cost** (LoRA parameters) is approximately 76.24M for FedProxy, about four times higher than FedOT's ∼19.06M. Similarly, the **Client Computation Cost** (trainable parameters) is ∼38.12M for FedProxy, also four times that of FedOT's ∼9.53M. These figures underscore that

Method	Initial Comm Cost	Iterative Comm Cost	Client Compute Cost	Performance (ALL Avg.)
CentSFT	∼7B	N/A	∼76.24M	0.7723
FedOT	∼3.5B	∼19.06M	∼9.53M	0.5300
FedProxy	**∼3.5B**	**∼76.24M**	**∼38.12M**	0.6987

Table 6: Cost-performance analysis in terms of parameter counts. FedProxy's performance gain is achieved by investing more in iterative communication and client-side computation, while maintaining a comparable initial model download size.

FedProxy's performance gain is achieved by investing more in iterative communication and client-side computation.

E Additional Ablation Study

E.1 IP Protection Analysis

To assess IP protection, we analyze the standalone performance of proxy SLMs. We conducted an ablation study under a homogeneous setting with a 50% compression ratio. The results are presented in Table 7. The results in Table 7 lead to two key conclusions. First, the initial zero-shot proxy SLM (`FedProxy-SLM-ZS`) performs significantly worse than the original zero-shot LLM (`LLM-ZS`) across all benchmarks (e.g., for LLaMA2-7B, an overall score of 0.3727 vs. 0.5047). This confirms that the compressed SLM is not a powerful standalone model, mitigating IP risks. Second, after federated fine-tuning, the final **`FedProxy-LLM`** substantially outperforms the original `LLM-ZS` (e.g., for LLaMA2-7B, 0.6987 vs. 0.5047). This demonstrates the effectiveness of our framework: the proxy SLM serves as an effective vehicle for federated learning, and the knowledge fusion mechanism successfully integrates the gains back into the full model.

Model	Method	QA	GLUE	ALL
LLaMA2-7B	LLM-ZS	0.4600	0.5494	0.5047
	`FedProxy-SLM-ZS`	**0.2242**	**0.5213**	**0.3727**
	`FedProxy-SLM-SFT`	0.4260	0.8603	0.6432
	`FedProxy-LLM`	0.5935	0.8038	0.6987
Mistral-7B	LLM-ZS	0.6017	0.7287	0.6652
	`FedProxy-SLM-ZS`	0.2510	0.4419	0.3464
	`FedProxy-SLM-SFT`	0.4713	0.8564	0.6638
	`FedProxy-LLM`	0.6692	0.8445	0.7568

Table 7: Analysis of standalone SLM performance. The initial compressed SLM (`FedProxy-SLM-ZS`) is significantly weaker than the original LLM.

E.2 Impact of Compression Ratio

To investigate the impact of the compression ratio, a critical hyperparameter, we conducted an ablation study on FedProxy in the homogeneous task setting. We compare the performance of 30%, 50%, and 70% compression ratios. The results are summarized in Table 8. As the table illustrates, a lower compression ratio (30%) yields the best performance, but the gains over the 50% ratio are marginal, while the client-side resource requirements are significantly higher. Conversely, a higher compression ratio (70%) leads to a more noticeable drop in performance, as the proxy SLM loses too much capacity to serve as a high-fidelity surrogate for the LLM. Therefore, we find that the 50% ratio strikes a strong balance between performance and resource efficiency.

Model	Ratio	QA	GLUE	ALL
LLaMA2-7B	30%	0.6261	0.8678	0.7469
	50%	**0.5935**	**0.8038**	**0.6987**
	70%	0.4657	0.6682	0.5670
Mistral-7B	30%	0.6745	0.8918	0.7831
	50%	**0.6692**	**0.8445**	**0.7568**
	70%	0.6042	0.7528	0.7096

Table 8: Impact of different compression ratios on FedProxy performance.

E.3 Comparison with ProxyTuning

To evaluate the efficacy of our parameter-space "plug-in" mechanism, we benchmark FedProxy against ProxyTuning (Liu et al., 2024; Gao et al., 2024), a representative decoding-time logit-level fusion strategy. While ProxyTuning (PT) modulates the model's output distribution during inference, FedProxy achieves knowledge integration directly within the parameter space. We evaluate two fusion approaches: (1) **FedProxy**, our default parameter-space fusion that directly replaces proxy SLM parameters in the LLM; (2) **FedProxy+PT**, replacing FedProxy's default parameter-space fusion with ProxyTuning's logit-level adjustment during inference, where the logit difference between tuned and untuned proxy SLMs is added to the LLM's logits. This experiment, conducted in the

homogeneous setting with compression ratios of 50% and 70%, aims to answer: ***Does parameter-space fusion match or exceed the performance decoding-time logit-level fusion for knowledge transfer?***

Table 9 illustrates a task-dependent performance trade-off. FedProxy consistently outperforms FedProxy+PT on generative QA tasks(e.g., for LLaMA2-7B at 50% compression, 0.5935 vs. 0.5665), suggesting that direct parameter fusion better preserves complex reasoning capabilities. Conversely, FedProxy+PT exhibits advantages on GLUE tasks (e.g., 0.8733 vs. 0.8038 for LLaMA2-7B at 50% compression), indicating that logit-shifting may be sufficient for discriminative tasks. Importantly, at 70% compression, both methods experience significant performance degradation (e.g., for LLaMA2-7B, FedProxy drops from 0.6987 to 0.5670, and FedProxy+PT drops from 0.7199 to 0.6053 on ALL tasks). This indicates that the performance decline is primarily due to compression-induced limitations such as missing critical layers or insufficient parameter coverage, rather than the fusion mechanism itself.

Crucially, FedProxy achieves these competitive results with **zero additional inference overhead**, as it eliminates the need to dual-run proxy models during decoding, which constitutes a significant advantage for practical deployment in resource-constrained scenarios.

Model	Ratio	Method	QA	GLUE	ALL
LLaMA2-7B	50%	FedProxy+PT	0.5665	**0.8733**	**0.7199**
		FedProxy	**0.5935**	0.8038	0.6987
	70%	FedProxy+PT	0.4565	**0.7540**	**0.6053**
		FedProxy	**0.4657**	0.6682	0.5670
Mistral-7B	50%	FedProxy+PT	0.6332	**0.8506**	0.7419
		FedProxy	**0.6692**	0.8445	**0.7568**
	70%	FedProxy+PT	0.6014	**0.8029**	**0.7021**
		FedProxy	**0.6042**	0.7528	0.6785

Table 9: Comparative analysis between parameter-space fusion (FedProxy) and decoding-time fusion (FedProxy+PT). FedProxy excels in generative QA tasks by internalizing knowledge within model parameters, whereas FedProxy+PT (ProxyTuning) exhibits advantages in discriminative GLUE tasks. Critically, FedProxy achieves competitive performance while maintaining **zero additional inference overhead**, whereas logit-level fusion requires dual-proxy execution.