Optimizing Multilingual LLMs via Federated Learning: A Study of Client Language Composition

Aleix Sant[1], **Jordi Luque**[1], **Carlos Escolano**[2]

Scientific Research, Telefónica Innovación Digital
Universitat Politècnica de Catalunya
Barcelona, Spain

{aleix.santsavall, jordi.luqueserrano}@telefonica.com
carlos.escolano@upc.edu

Abstract

Federated Learning (FL) of Large Language Models (LLMs) in multilingual environments presents significant challenges stemming from heterogeneous language distributions across clients and disparities in language resource availability. To address these challenges, we extended the *FederatedScope*-LLM framework to support multilingual instruction-tuning experiments with LLMs. We also introduced a novel client-specific early stopping mechanism, Local Dynamic Early Stopping (LDES-FL), which allows clients to pause and resume local training based on client-side validation performance, enhancing training efficiency and sustainability. Through a series of experiments, we studied how client language composition — from fully monolingual to increasingly multilingual clients — affects multilingual quality, fairness and training cost. Monolingual local fine-tuning remains the most effective for single-language specialization, whereas federated training is better suited to learning a single balanced multilingual model. In FL, increasing within-client multilinguality leads to stronger and fairer global models, narrows the gap to centralized multilingual fine-tuning, and yields the largest gains for lower-resource languages, albeit at the cost of more optimization steps. Overall, our results identify client language composition as a key design variable in multilingual FL, shaping performance, fairness and efficiency.

Keywords: Federated Learning, Large Language Models, Multilingual NLP, Early Stopping

1. Introduction

Federated Learning (FL) is a distributed machine learning paradigm that enables collaborative model training across multiple clients while preserving data privacy. Unlike traditional centralized approaches, FL keeps data localized: clients train models on their private data and share only model updates (e.g., weights or gradients) with a central server (the aggregator). This design reduces privacy risks and supports compliance with regulations such as GDPR. The standard FL workflow involves a server distributing a global model, clients performing local training, and the server aggregating their updates.

Despite its potential, applying FL to Large Language Models (LLMs) (Cheng et al., 2025; Hu et al., 2025) introduces challenges, particularly in communication and computation for resource-constrained clients. Parameter-Efficient Fine-Tuning (PEFT) methods, such as Low-Rank Adaptation (LoRA) (Hu et al., 2022), mitigate these issues by reducing trainable parameters and communication overhead (Ye et al., 2024b). Another major challenge is data heterogeneity. In real-world FL deployments, client data is often non-IID (i.e., not Independent and Identically Distributed), which impairs convergence and degrades generalization performance for classical algorithms like FedAvg (McMahan et al., 2017). This phenomenon, commonly referred to as *client drift* (Lu et al., 2024), also arises in multilingual FL scenarios (Weller et al., 2022; Wang et al., 2022), where each client may hold data in a different language, representing a specific type of non-IID distribution. Such linguistic diversity induces highly skewed distributions (Manoel et al., 2023), complicating the optimization process and slowing convergence, as local updates tend to diverge from the global objective.

In this work, we investigate how the distribution of multilingual and monolingual data in clients affects performance, fairness and convergence behavior in the federated fine-tuning of multilingual LLMs. To this end, we design a series of experimental scenarios in which the language composition of each client's local dataset is systematically varied. These scenarios range from fully monolingual settings – where each client contains data in a single language – to fully multilingual ones, where each client holds data from several languages, thereby approximating increasingly IID data distributions within the FL framework. Our experiments show a common FL pattern: stronger cross-client heterogeneity (here, a larger share of disjoint monolingual data across clients) increases client drift and harms the global solution. Making clients more multilingual

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 8985–8996
11–16 May 2026.

improves average multilingual performance and reduces cross-lingual disparities (i.e., increases multilingual fairness) – especially for lower-resource languages – at the cost of additional optimization steps.

To carry out this research, we built upon the *FederatedScope*-LLM framework (Kuang et al., 2024), extending its capabilities to obtain a flexible repository[1], designed for multilingual FL experiments with LLMs. The main contributions of this paper are listed below, with the corresponding implementation provided in the repository:

1. **Multilingual FL support:** We add explicit multilingual support for federated fine-tuning of LLMs with flexible prompt integration, language-aware sample processing and multilingual FL data pipelines.
2. **Local Dynamic Early Stopping (LDES-FL):** A client-level early stopping mechanism based on local validation loss, allowing clients to pause and resume training dynamically, enabling a more nuanced analysis of multilingual FL dynamics and reducing unnecessary computation.
3. **Federated FT experiments on different language client compositions:** We fine-tuned `salamandra-2b-instruct` across multiple FL settings with varying degrees of within-client multilinguality, ranging from fully monolingual to highly multilingual clients.

2. Related Work

Federated Learning is an active area of privacy-aware research that has been extensively studied across domains such as computer vision, speech processing and natural language understanding. However, while research at the intersection of FL and LLMs is growing, it is still limited (Hilmkil et al., 2021; Zheng et al., 2024; Ye et al., 2024b; Liu et al., 2025), and studies specifically addressing multilingual FL with LLMs are even scarcer (Weller et al., 2022; Guo et al., 2024; Lee et al., 2025; Zhao et al., 2025), as the setting introduces additional difficulties due to language heterogeneity and highly imbalanced resource distributions across languages.

The authors in Guo et al. (2024) propose a federated multilingual framework for language modeling and text classification tasks that integrates LoRA-based parameter-efficient tuning with language-family clustering, thereby reducing communication overhead while alleviating cross-lingual interference across heterogeneous language distributions. The work in Lee et al. (2025) introduces an FL approach that enables clients to collaboratively learn personalized PEFT configurations, such as LoRA ranks. The method automatically discovers language-agnostic rank structures and facilitates cross-lingual transfer. According to Weller et al. (2022), fine-tuning pretrained multilingual models in FL mitigates the performance degradation typically seen in FL, yielding performance close to (and sometimes better than) centralized training, even when clients are partitioned by language. This suggests pretrained multilingual models are a strong practical choice for federated settings with diverse private language data.

Beyond multilinguality, extensive research has addressed the effects of data heterogeneity (i.e., non-IID data distributions) in FL (Kairouz et al., 2019; Mendieta et al., 2022; Ye et al., 2024a), a key factor that contributes to the well-known problem of client drift, which hinders convergence and degrades global model performance. The FL community has proposed a range of approaches to mitigate this issue, typically categorized into data-centric and model-centric strategies (Tan et al., 2023).

Data-centric approaches seek to reduce distributional discrepancies through data augmentation or shared proxy datasets (Zhao et al., 2018). However, these methods often assume access to public or shared data, which weakens FL's privacy guarantees. Other techniques, such as FAVOR (Wang et al., 2020), address heterogeneity by selectively sampling clients in each communication round to balance non-IID effects. Model-centric strategies, in contrast, focus on modifying the training process or optimization objectives. For instance, FedProx (Li et al., 2020) introduces a proximal term to constrain local updates and stabilize convergence under heterogeneous data conditions, while SCAFFOLD (Karimireddy et al., 2019) reduces client drift using control variates on both the client and server sides to correct biased updates.

Despite this progress, the intersection of multilinguality, LLM fine-tuning and federated optimization dynamics still remains largely underexplored. Our work contributes to filling this gap by systematically analyzing how alternative multilingual client-level data distributions impact both convergence behavior and final task performance in federated fine-tuning of multilingual LLMs. Concretely, we test in controlled experiments whether increasing within-client multilinguality reduces client drift.

3. Multilingual FL Adaptation

In the standard FL scenario, let $\mathcal{D}_T$ denote the overall training dataset partitioned across the $|\mathcal{C}|$ clients into local subsets $\mathcal{D}_{T_i}$. The server then aggregates

[1] https://github.com/Telefonica-Scientific-Research/FedEloquence

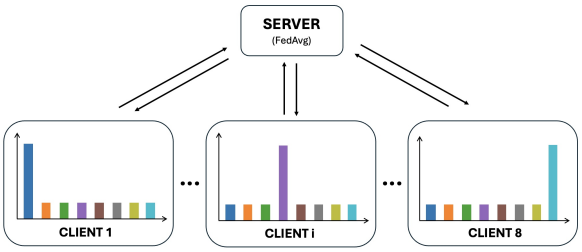

Figure 1: Illustration of a multilingual FL setup. Each client primarily contains data from a single dominant language (represented by the tallest bar) along with smaller, equal portions of data from the remaining languages. Each color corresponds to a different language. As in our experiments, there are eight clients in total.

the weights $\mathbf{w}_i$ sent by the clients as:

$$\mathbf{w} = \sum_{i=1}^{|\mathcal{C}|} \alpha_i \mathbf{w}_i, \tag{1}$$

where $\mathbf{w}_i$ is the parameter vector trained by client i, $\mathbf{w}$ is the parameter vector after aggregation on the server, $\mathcal{C}$ is the set of clients, and $\alpha_i = \frac{|\mathcal{D}_{T_i}|}{|\mathcal{D}_T|} \geq 0$ denotes the aggregation weight of client i, with $\sum_{i=1}^{|\mathcal{C}|} \alpha_i = 1$. Here, $|\mathcal{D}_{T_i}|$ is the size of the local training set of client i, and $|\mathcal{D}_T| = \sum_{i=1}^{|\mathcal{C}|} |\mathcal{D}_{T_i}|$ is the total number of training samples across all clients. Formally, the optimization problem can be expressed as:

$$\min_{\mathbf{w} \in \mathbb{R}^d} F(\mathbf{w}) := \frac{1}{|\mathcal{C}|} \sum_{\mathcal{C}_i} F_i(\mathbf{w}) \tag{2}$$

where

$$F_i(\mathbf{w}) := \mathbb{E}_{(x,y) \sim \mathcal{D}_{T\,i}} \left[F_i(\mathbf{w}; x, y) \right] \tag{3}$$

denotes the expected loss of the i-th client over its local dataset $\mathcal{D}_{T_i}$ (i.e., the expectation is taken over all data pairs (x, y) sampled from the client's data distribution.) In the context of instruction-tuning, x represents the input (consisting of the instruction and associated prompt) and y is the corresponding ground-truth response. For more details, see Section 5.2.

Aggregation strategies in FL range from the widely used FedAvg (McMahan et al., 2016) to optimization-aware methods like FedProx (Sahu et al., 2018), SCAFFOLD (Karimireddy et al., 2019) or FedOpt (Reddi et al., 2020). In our federated settings, the size of each client's local training dataset $|\mathcal{D}_{T\,i}|$ is the same for all clients. Thus, all α_i in equation 1 have the same value, resulting in a FedAvg algorithm where all clients contribute the same to the global model.

3.1. Parameter-Efficient Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) is highly effective in fine-tuning LLMs. In PEFT, most model parameters remain frozen, and only a small targeted subset is optimized for downstream tasks (Houlsby et al., 2019). Although this significantly reduces computational and memory overhead, by updating only a small subset of parameters – typically around 1% or 2% of the whole pre-trained model –, empirical findings in Zhang et al. (2023) indicate that incorporating PEFT techniques can sometimes reduce language model performance.

PEFT techniques, classified by Ding et al. (2022), fall into three distinct categories. Addition-based approaches, exemplified by adapters (Houlsby et al., 2019), augment the original model with new trainable elements. Specification-based approaches, such as diff pruning (Guo et al., 2021), refine the fine-tuning process by selectively activating existing parameters. Lastly, reparameterization-based methods, including LoRA (Hu et al., 2022), optimize fine-tuning efficiency through reparameterization strategies.

While the formulations in 1, 2 and 3 are written in a generic way, in the experiments conducted in this work, w_i denotes the set of trainable LoRA parameters (A_i, B_i) of client i (i.e., the collection of LoRA matrices across all adapted layers), whereas w denotes the aggregated global LoRA parameters. Concretely, given a frozen pre-trained weight matrix W_0, LoRA learns two low-rank matrices A_i and B_i for each client i, which induce an update $\Delta W_i = B_i A_i$. Hence, the adapted weight matrix becomes $W_0 + \Delta W_i$, and the federated communication and aggregation operate only on these trainable LoRA parameters, rather than on the full backbone parameters.

3.2. FederatedScope Extension for multilingual FL with LLMs

To support multilingual FL with LLMs, we extended *FederatedScope-LLM* with the necessary modifications to handle multilingual scenarios, allowing each client to operate on language-specific data. We developed tools for multilingual splitting that include a shared multilingual validation and test set for the global model $\mathbf{w}$ in the server, alongside language-specific training, validation and test sets for each client. This partitioning reflects a realistic FL setting in which the server and clients are distinct entities, and each client i holds a private local dataset $\mathcal{D}_i$ consisting exclusively of examples in a single language. We refer to this configuration in our experiments as 100% mono, which means fully monolingual clients.

Building on the initial scenario, we also prepared configurations with increasing degrees of client mul-

tilinguality (85% mono, 70% mono, 50% mono, etc.). These settings were created by redistributing part of each client's monolingual data across other clients and replacing it with samples from different languages. As a result, the share of same-language data per client decreases as within-client multilinguality increases. Figure 1 illustrates this multilingual client composition. Importantly, the total number of training examples per client is kept constant in all settings. Although real-world multilingual FL deployments often involve strongly imbalanced per-language and per-client data volumes, we enforce equal per-client dataset sizes to maintain a controlled experimental setting and isolate the impact of multilinguality from confounding effects due to diverse client dataset sizes.

4. Local Dynamic Early Stopping

We introduce Local Dynamic Early Stopping for Federated Learning (LDES-FL), a method that enables each client to autonomously decide when to stop local training based on validation performance on its private validation dataset. Each client monitors its validation loss, $F_i(\mathbf{w}^{(t)})$, and stops training once no further improvement is observed. The client then periodically sends its best-performing local model to the server for aggregation. The server continues to perform model averaging across all clients, but for those halted clients, it employs their best local models in subsequent communication rounds.

Crucially, our algorithm supports adaptive client rejoining. Even after stopping local training, stopped clients continue to evaluate the validation loss of the downloaded global model on their local validation dataset. A client that previously stopped may resume local training if the downloaded model yields an improvement in its local validation loss, $F_i(\mathbf{w}^{(t)})$. Federated training terminates once all clients are stopped. This process is illustrated in Figure 2, where colored regions indicate active training periods and blank regions represent idle periods after local early stopping. Note that clients `ES` and `DE` both stop and later resume training. For a better understanding of the algorithm, see Algorithm 1. In the algorithm, a_i indicates whether client $\mathcal{C}_i$ is active (1) or stopped (0), while $\mathbf{w}$ is used as a generic notation for the trainable client parameters. In our LoRA-based setting, these correspond to the LoRA matrices A_i and B_i for each client.

This dynamic participation mechanism contrasts with standard federated early stopping, in which global training termination is determined by a global validation loss (e.g., the mean of all clients' losses; black dashed curve in Figure 3), as implemented in *FederatedScope*. In contrast, LDES-FL enables personalized convergence, allowing clients to stop and resume training independently.

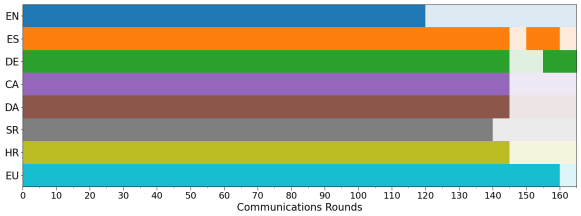

Figure 2: Training evolution of clients using LDES-FL with FedAvg, where each client holds data in a different language (100% mono).

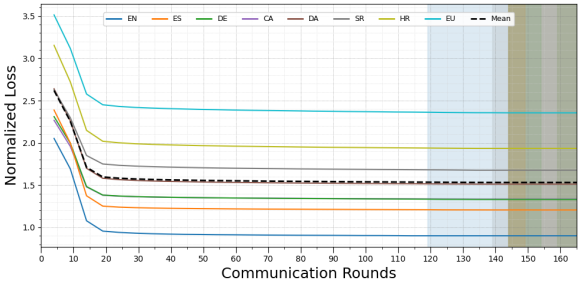

Figure 3: Validation loss across clients under standard FedAvg (100% mono). Low-resource languages show higher validation loss, whereas high-resource languages achieve lower loss. All clients improve at a similar rate during federated training. Shaded regions mark rounds where a client is stopped.

Figure 3 shows both the best validation loss for each monolingual client and the average across clients for the 100% mono setting. The order of the languages in the figure reflects their resource availability in the model used (`salamandra-2b-instruct`), based on the amount of data used during pretraining. Languages with more pretraining data (high-resource languages) tend to perform better, yielding lower losses, and are shown at the bottom. In contrast, languages with less pretraining data (low-resource languages) tend to perform worse, yielding higher losses, and are shown at the top.

5. Multilingual Experiments

5.1. Data

We employ a multilingual version of the Alpaca Cleaned dataset covering eight languages with their corresponding ISO 639-1 codes: English (`en`), Spanish (`es`), German (`de`), Catalan (`ca`), Danish (`da`), Serbian (`sr`), Croatian (`hr`) and Basque (`eu`), each containing 52,002 samples. The original Alpaca dataset, introduced by Stanford for instruction-tuning (Taori et al., 2023), was later refined using GPT-4 to improve consistency and formatting (Gururise, 2023). This cleaned version was subsequently expanded through large-scale English-to-

Algorithm 1: Local Dynamic Early Stopping

Data: $\mathcal{D}_i$, $\mathbf{w}^{(0)}$, $a_i \in \{0,1\}$, $\mathcal{C}_i$, $P_{\max}$

Initialize $t \leftarrow 1$, $p_i \leftarrow 0$, $\mathbf{w}_i^{(0)} \leftarrow \mathbf{w}^{(0)}$, $\mathbf{w}_i^{best} \leftarrow \mathbf{w}^{(0)}$ and $a_i \leftarrow 1$, $\forall i \in \mathcal{C}$

while $\sum_i a_i > 0$ **do**

  Local training process:

  for $\mathcal{C}_i \in \mathcal{C}$ **do**

    if $a_i = 1$ **then**

      Update the local parameters $\mathbf{w}_i^{(t)}$

      using local TRAIN data $\mathcal{D}_{T_i} \in \mathcal{D}_i$

      $\mathbf{w}_i^{(t)} \leftarrow \arg\min_{\mathbf{w}_i} F_i(\mathbf{w}_i)$

  Model aggregation process:

  The server computes weights

  $$\mathbf{w}^{(t)} = \sum_{i=1}^{N} \alpha_i \mathbf{w}_i^{(t)} \text{ with } \alpha_i = \frac{|\mathcal{D}_{Ti}|}{|\mathcal{D}_T|}$$

  The server broadcasts global parameters

  Local validation process:

  for $\mathcal{C}_i \in \mathcal{C}$ **do**

    Validate the global parameters $\mathbf{w}^{(t)}$

    using local VAL data $\mathcal{D}_{V_i} \in \mathcal{D}_i$

    if $F_i(\mathbf{w}^{(t)}) > F_i(\mathbf{w}_i^{best})$ **then**

      $p_i \leftarrow p_i + 1$

      if $p_i \geq P_{\max}$ **then**

        $a_i \leftarrow 0$ stopped

        $\mathbf{w}_i^{(t)} \leftarrow \mathbf{w}_i^{best}$

    else

      $a_i \leftarrow 1$ resumed

      $p_i \leftarrow 0$

      $\mathbf{w}_i^{best} \leftarrow \mathbf{w}^{(t)}$

  Clients upload local VAL loss $F_i(\mathbf{w}^{(t)})$

  $t \leftarrow t + 1$

Result: $\mathbf{w}^{(t)}$

XX translation (Upadhayay and Behzadan, 2023). From this resource, we constructed data partitions specifically designed for our FL experiments, ensuring comparable data volumes and distributions across clients in the multilingual instruction-tuning setup. The datasets used in our experiments can be downloaded from Hugging Face via the instructions provided in the repository[1].

Each federated client receives 48,960 training samples and 1,020 validation samples. As introduced earlier, in our experiments the language composition of each client's data varies across settings, ranging from fully monolingual (i.e., clients contain data from a single language) to fully multilingual (i.e., all clients share data from all languages). In the mixed settings, clients contain different ratios of monolingual and multilingual data, which determines the level of cross-client language heterogeneity in the federation. The specific data distributions evaluated in our experiments are summarized in Table 1.

Client Composition	Heterogeneity Level
100% mono - 0% multi	● ● ● ● ● ●
85% mono - 15% multi	● ● ● ● ● ❍
70% mono - 30% multi	● ● ● ● ❍ ❍
50% mono - 50% multi	● ● ● ❍ ❍ ❍
30% mono - 70% multi	● ● ❍ ❍ ❍ ❍
15% mono - 85% multi	● ❍ ❍ ❍ ❍ ❍

Table 1: Client language compositions used to control cross-client language heterogeneity in our FL experiments. More filled circles indicate higher heterogeneity and stronger cross-client non-IIDness (i.e., larger monolingual share), whereas fewer filled circles indicate a more IID setting in which clients have more similar language distributions.

In all the federated settings, the central server holds no training data. Instead, it maintains a fixed multilingual test set containing 501 samples per language (4,008 samples in total) used exclusively for global evaluation.

Client data was randomly partitioned without enforcing cross-lingual alignment, preserving only the desired language proportions per client. Consequently, translated instances can end up in different splits across clients (e.g., an example used for training in one language may appear in the validation or test split in another). Throughout all FL experiments, training and validation data remain client-side, while the server uses only the held-out multilingual test set for evaluation.

Although the multilingual dataset contains 52,002 instances per language, the experiments reported here rely only on the fixed partitions described above. Beyond the client-side training and validation splits and the held-out server-side test set, we also defined additional disjoint subsets that are not used in the current study. These reserved subsets are intended for auxiliary analyses and future extensions beyond the current FL experiments, including client-side testing and server-side validation.

5.2. Model and Prompt

As the backbone model for all experiments, we employ the `salamandra-2b-instruct` model (Gonzalez-Agirre et al., 2025), a multilingual instruction-tuned LLM whose pre-training corpus covers 35 European languages and code, including the languages considered in our experiments. To reduce training costs while maintaining competitive performance, we fine-tune the model using LoRA adapters with rank $r = 16$ and scaling factor $\alpha = 32$ (see Section 3.1). For prompting the model, we use the popular ChatML-style template (as recommended by the model creators) plus the original prompt template format from the ALPACA

dataset for processing samples. We translated the English Alpaca-style template into the seven additional languages used in our experiments, ensuring that each sample is processed with the prompt in its corresponding language. The German version, for example, is shown below.

```
Nachfolgend finden Sie eine Anweisung,
die eine Aufgabe beschreibt, gepaart
mit einer Eingabe, die weiteren
Kontext liefert.
Schreiben Sie eine Antwort, die die
Anfrage angemessen ergänzt.

### Anweisung:  {Instruction in German}

### Eingabe:  {Input in German}

### Antwort:  {LLM response}
```

5.3. Methodology

First, we performed preliminary experiments comparing LDES-FL with standard federated early stopping under the same federated fine-tuning setup (FedAvg, patience $= 1$) in two client language compositions: 100% mono and 15% mono. The results, presented in Section 6, motivate our use of LDES-FL in the subsequent federated experiments.

Then, we fine-tuned separate models for each of the eight languages in a centralized, monolingual setting. For each language, we initialized the LoRA adapters using the default PEFT configuration, with matrix A randomly initialized and matrix B initialized to zero, and trained the model on the corresponding training split $\mathcal{D}_{T_i}$. Standard early stopping was applied based on validation loss, with a patience of 5 epochs and a minimum improvement threshold of 0.001. Validation loss was computed after every epoch. This setup corresponds to a purely local learning baseline, in which each client optimizes its own model without any communication or parameter sharing with other languages. These models are reported as *Local FT (lang)* in Table 3.

We also fine-tuned a single model on the union of all client data, $\mathcal{D} = \bigcup_i \mathcal{D}_i$ (i.e., the whole multilingual dataset), shuffling samples across languages and using the same LoRA initialization and early-stopping configuration as in the monolingual local fine-tunings. Since this dataset is eight times larger, validation loss was computed after processing a portion of training data equivalent to a single client dataset, ensuring a comparable evaluation frequency for early stopping. Results for this setting are reported as *Local FT (multilingual)* in Table 3.

Following this, we fine-tuned the model in a federated setting using the FedAvg algorithm across the multilingual FL scenarios described in Section 5.1, again starting from the default LoRA configuration. All federated experiments used the LDES-FL mechanism as the stopping criterion, with a local patience of 1. This parameter specifies the number of consecutive validation rounds without improvement that each client can tolerate before halting its local training. It is denoted by P_{max} in Algorithm 1, while p_i represents the corresponding counter for client i. In all FL setups, each active client performed 160 local mini-batch steps per round with micro-batch size 2 and gradient accumulation over 16 steps, corresponding to 10 optimizer updates per round. This yields an effective local batch size of 32 samples per GPU per optimizer update and 320 processed samples per GPU per round. Optimization was performed using OneBitAdam (learning rate 0.001, gradient clipping norm 1.0) on cross-entropy loss. Validation was conducted every 5 training rounds.

		Mean ↑		σ ↓		Optim. steps
		R	FB	R	FB	
100% mono	Standard	0.202	0.877	6.75e-2	1.29e-2	1.44e+04
	LDES	0.203	0.877	6.47e-2	1.26e-2	1.12e+04
15% mono	Standard	0.224	0.880	6.09e-2	1.22e-2	1.11e+05
	LDES	0.221	0.880	6.09e-2	1.23e-2	7.57e+04

Table 2: Performance and optimization steps obtained with two early stopping methods (standard federated early stopping and LDES-FL) under two client language compositions. Federated fine-tuned models are evaluated on the multilingual test set described in Section 5.1. R and FB denote ROUGE-L and FBERT, respectively.

6. Analysis and Discussion

For the preliminary early-stopping experiments shown in Table 2, LDES-FL reduces unnecessary computation relative to standard federated early stopping while maintaining comparable performance. In the 100% mono setting, LDES-FL preserves performance, slightly reduces cross-lingual dispersion, and lowers the number of optimization steps by approximately 22%. In the 15% mono setting, performance again remains essentially unchanged, while the number of optimization steps decreases by approximately 32%. For these reasons, we adopted LDES-FL for the rest of the FL experiments.

Having established the efficiency of LDES-FL, we now turn to the main results. Table 3 reports the performance of all models on the multilingual test set of the server. Inference was performed with greedy decoding to ensure reproducibility. *Base Model* refers to the original pretrained instruction-tuned model (`salamandra-2b-instruct`) before any additional adaptation.

A first observation is that *Local FT (multilingual)* provides the strongest overall results in the table. It achieves the best aggregate performance, with a mean ROUGE-L of 0.237 and a mean FBERT of 0.884, and it also obtains the best score in almost

	EN		ES		DE		CA		DA		SR		HR		EU		Mean ↑		σ ↓	
	R	FB	R	FB	R	FB	R	FB	R	FB	R	FB	R	FB	R	FB	R	FB	R	FB
Base Model	0.307	0.893	0.215	0.877	0.170	0.873	0.207	0.871	0.160	0.869	0.074	0.861	0.126	0.856	0.080	0.840	0.167	0.867	7.70e-2	1.57e-2
Local FT (EN)	0.351	0.903	0.232	0.882	0.187	0.876	0.233	0.878	0.186	0.875	0.131	0.869	0.147	0.863	0.105	0.852	0.197	0.875	7.73e-2	1.49e-2
Local FT (ES)	0.295	0.897	0.248	0.883	0.188	0.877	0.196	0.876	0.187	0.875	0.168	0.868	0.159	0.866	0.102	0.850	0.193	0.874	5.80e-2	1.47e-2
Local FT (DE)	0.322	0.898	0.237	0.881	0.212	0.883	0.233	0.880	0.189	0.877	0.181	0.871	0.165	0.870	0.105	0.851	0.206	0.876	6.32e-2	1.34e-2
Local FT (CA)	0.296	0.893	0.131	0.866	0.168	0.871	0.253	0.884	0.146	0.865	0.145	0.860	0.104	0.849	0.065	0.834	0.163	0.865	7.61e-2	1.86e-2
Local FT (DA)	0.315	0.895	0.241	0.882	0.185	0.875	0.244	0.881	0.228	0.886	0.168	0.868	0.151	0.863	0.109	0.849	0.205	0.875	6.47e-2	1.45e-2
Local FT (SR)	0.259	0.883	0.184	0.862	0.154	0.862	0.174	0.860	0.107	0.848	**0.207**	0.876	0.024	0.869	0.046	0.819	0.144	0.860	8.03e-2	1.97e-2
Local FT (HR)	0.263	0.887	0.229	0.879	0.190	0.877	0.207	0.874	0.181	0.874	0.028	0.866	0.211	**0.881**	0.099	0.844	0.176	0.873	7.62e-2	1.31e-2
Local FT (EU)	0.273	0.888	0.224	0.877	0.182	0.871	0.223	0.876	0.178	0.872	0.163	0.862	0.145	0.859	0.143	0.864	0.191	0.871	4.52e-2	**9.40e-3**
Local FT (multilingual)	**0.353**	**0.905**	**0.258**	**0.886**	**0.223**	**0.885**	**0.260**	**0.886**	**0.231**	**0.887**	0.203	**0.878**	**0.217**	0.880	**0.147**	**0.865**	**0.237**	**0.884**	**5.90e-2**	1.12e-2
FedAvg (100% mono)	0.330*	0.900*	0.229*	0.881*	0.194*	0.878*	0.237*	0.880*	0.189*	0.878*	0.162*	0.871*	0.174*	0.871*	0.110*	0.854*	0.203	0.877	6.47e-2	1.29e-2
FedAvg (85% mono)	0.334*	0.900*	0.229*	0.881*	0.197*	0.879*	0.242*	0.881*	0.193*	0.878*	0.157*	0.872*	0.173*	0.872*	0.112*	0.854*	0.205	0.877	6.64e-2	1.28e-2
FedAvg (70% mono)	0.335*	0.900*	0.232*	0.881*	0.206*	0.880*	0.240*	0.881*	0.203*	0.880*	0.154*	0.872*	0.181*	0.872*	0.114*	0.855*	0.208	0.878	6.57e-2	1.28e-2
FedAvg (50% mono)	0.339*	0.901*	0.238*	0.882*	0.210*	0.881*	0.243*	0.881*	0.211*	0.883*	0.179*	0.873*	0.195*	0.874*	0.121*	0.857*	0.217	0.879	6.23e-2	1.23e-2
FedAvg (30% mono)	0.337*	0.901*	0.244*	0.882*	0.206*	0.881*	0.251*	0.883*	0.208*	0.882*	0.194*	0.874*	0.197*	0.875*	0.125*	0.858*	0.220	0.880	6.06e-2	1.21e-2
FedAvg (15% mono)	0.341*	0.902*	0.242*	0.882*	0.206*	0.881*	0.247*	0.883*	0.211*	0.883*	0.194*	0.873*	0.197*	0.875*	0.127*	0.858*	0.221	0.880	6.09e-2	1.22e-2

Table 3: Results for the multilingual test set of the server. R and FB stand for ROUGE-L (Lin, 2004) and FBERT (Zhang et al., 2020) respectively. The language in columns corresponds to the monolingual part of the test set. Mean reports the average score across all languages, whereas σ denotes the corresponding standard deviation, used here as an indicator of multilingual fairness. A bootstrap test with 100 sets (p-value < 0.05) confirms that all federated models achieve statistically significant improvements over the *Base Model* on all evaluated metrics and languages, as indicated by an asterisk (*).

Training Setting	Norm. Optim. Steps
Local FT (EN)	0.106
Local FT (ES)	0.312
Local FT (DE)	0.345
Local FT (CA)	0.362
Local FT (DA)	0.351
Local FT (SR)	0.429
Local FT (HR)	0.340
Local FT (EU)	0.385
Local FT (multilingual)	1.000
FedAvg (100% mono)	0.061
FedAvg (85% mono)	0.103
FedAvg (70% mono)	0.146
FedAvg (50% mono)	0.274
FedAvg (30% mono)	0.370
FedAvg (15% mono)	0.414

Table 4: Training cost normalized by the highest-step configuration, *Local FT (multilingual)*, which requires 1.83×10^5 total optimization steps. Each value is reported as a fraction of this reference.

all per-language metrics, with only minor exceptions (e.g., SR in ROUGE-L and HR in FBERT). This indicates that, when all multilingual data can be pooled and jointly optimized, centralized multilingual fine-tuning yields the strongest multilingual model in terms of average quality. Since this setting assumes centralized access to all training data, it should be interpreted as an upper bound for the federated setting, rather than as a directly comparable privacy-preserving alternative. Regarding multilingual fairness, *Local FT (multilingual)* also shows low standard deviations across languages ($\sigma_R = 5.90 \times 10^{-2}$ and $\sigma_{F\text{BERT}} = 1.12 \times 10^{-2}$), consistent with its strong multilingual behavior. Although the lowest standard deviation is obtained by *Local FT (EU)*, this is somewhat misleading, since that model does not achieve the best multilingual performance overall and instead produces more uniformly modest scores across languages. Thus, *Local FT (multilingual)* provides a more meaningful reference point for balanced multilingual performance, since it combines high average multilingual quality with relatively low cross-lingual dispersion.

Interestingly, in most cases, the multilingual Local FT model outperforms the set of monolingual Local FT models across all languages and metrics – even outperforming each monolingual model on the very language it was trained for. This suggests that multilingual joint optimization provides benefits beyond language-specific specialization. Such improvements may reflect positive cross-lingual transfer, whereby representations learned from multiple languages also help improve performance on individual target languages.

By contrast, monolingual Local FT models exhibit a different pattern. Each model achieves its highest performance on the language it was specifically trained on, compared to its performance on other languages. This confirms that local adaptation is highly effective for specializing the model to a single client language. However, these gains are not uniformly transferred to the rest of the multilingual test set. As a result, although monolingual Local FT improves target language performance, it generally leads to more uneven cross-lingual behavior than *Local FT (multilingual)*, as reflected in its higher standard deviation and lower average multilingual performance.

Compared with the *Base Model*, monolingual Local FT models often show small gains on languages that were not seen during fine-tuning. While this could suggest some degree of cross-lingual transfer, such improvements should be interpreted with caution. Since the multilingual splits are not strictly parallel, examples from different languages may still share content, structure or task-specific answer patterns. Therefore, part of the apparent trans-

	ROUGE-L						FBERT					
	Base Model	Avg. Local FT (mono)	Local FT (multi)	FedAvg 100% mono	FedAvg 15% mono	Δ	Base Model	Avg. Local FT (mono)	Local FT (multi)	FedAvg 100% mono	FedAvg 15% mono	Δ
H	0.231	0.232	0.278	0.251	0.263	+0.012	0.881	0.881	0.892	0.886	0.888	+0.002
M	0.184	0.198	0.246	0.213	0.229	+0.016	0.870	0.874	0.882	0.879	0.883	+0.004
L	0.093	0.128	0.189	0.149	0.173	+0.024	0.852	0.859	0.868	0.865	0.869	+0.004

Table 5: Average performance by resource group. H, M and L stand for high-resource, mid-resource and low-resource respectively. Avg. Local FT (mono) is the average of the eight monolingual Local FT models within each resource group. Δ denotes the absolute improvement from 100% mono to 15% mono.

fer may arise from indirect overlap in instance format or solution templates rather than from robust language-agnostic generalization. This is an important limitation of the dataset and should be kept in mind when interpreting zero-shot transfer across languages.

On the other hand, all federated models consistently improve over the *Base Model* on every reported aggregate metric. The strongest federated setting reaches a mean ROUGE-L of 0.221 and a mean FBERT of 0.880, compared with 0.167 and 0.867 for the *Base Model*. This corresponds to a substantial gain in average multilingual performance, especially in ROUGE-L. Moreover, all federated settings improve multilingual fairness, indicating that the gains are not concentrated in only a few languages, but are instead distributed more evenly across all languages. Under this criterion, FL not only improves quality, but also yields a fairer multilingual model.

Restricting the comparison to monolingual Local FT and federated adaptation, we observe an important trade-off. Monolingual Local FT yields the strongest language-specific specialization, but at the cost of weaker multilingual balance. Federated training, by contrast, achieves less extreme specialization on any single language while providing better overall multilingual coverage. Notably, from the 50% mono setting onward, federated models outperform all monolingual Local FT variants in mean performance, showing that aggregating updates from linguistically diverse clients produces a stronger and more balanced single model across languages. However, federated training remains below centralized multilingual fine-tuning, reflecting the cost of decentralization under privacy constraints.

The language composition of clients also has a clear effect on the behavior of FedAvg. As the proportion of multilingual data within each client increases (i.e., moving from 100% mono to 15% mono), average multilingual performance generally improves and the standard deviation across languages generally decreases. The best average performance is obtained in the 15% mono setting (mean ROUGE-L 0.221, mean FBERT 0.880), while the best fairness is observed in the 30% mono setting ($\sigma_R = 6.06 \times 10^{-2}$ and $\sigma_{F\text{BERT}} = 1.21 \times 10^{-2}$). This suggests that making clients more multilingual mitigates the tension between language-specific optimization and global multilingual generalization. A plausible explanation is that increasing client multilinguality reduces client-drift during federated optimization, since the local objective functions become closer approximations of the global objective, leading to more aligned gradient updates and less conflict across clients.

These improvements, however, come with a non-negligible computational cost. Table 4 shows that the total number of optimization steps increases sharply as clients become more multilingual. The most expensive configuration is *Local FT (multilingual)*, which serves as the highest-cost upper bound. By comparison, FedAvg (100% mono) requires only 1.12×10^4 total steps, while FedAvg (15% mono) requires 7.57×10^4 steps, approximately 6.8 times more than the monolingual federated setting but still substantially less than centralized multilingual fine-tuning. In other words, strongly monolingual clients exhibit faster apparent convergence in terms of optimization steps, but much of this speed reflects early saturation at a biased local optimum induced by non-IID data, whereas more multilingual clients support longer, more productive optimization toward a better global solution. This behavior is consistent with federated optimization theory: as clients become more multilingual, their local objectives align better with the global objective, which reduces client drift and leads to higher final performance, but also prolongs training because the model continues to improve for more communication rounds instead of plateauing early at a suboptimal solution. From a practical perspective, the results suggest a trade-off frontier: highly multilingual clients yield the strongest and fairest models at the cost of substantially more optimization steps, whereas more monolingual clients converge quickly but to a less optimal and less balanced global solution. Thus, the choice depends on whether deployment prioritizes training efficiency (fewer rounds to an early plateau) or balanced multilingual quality (more rounds to a better optimum).

As shown in Table 5, the effect of increasing client multilinguality is not uniform across resource language levels. When moving from FedAvg (100% mono) to FedAvg (15% mono), the

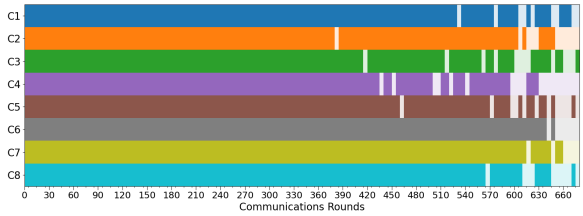

Figure 4: Training evolution of clients using LDES-FL with FedAvg in the 50% mono setting. Note that clients are not labeled by language as in Figure 2, as here each client contains a mix of languages.

average ROUGE-L score increases from 0.251 to 0.263 (+4.78%) for the high-resource group (EN, ES, DE), from 0.213 to 0.229 (+7.51%) for the mid-resource group (CA, DA), and from 0.149 to 0.173 (+16.11%) for the low-resource group (SR, HR, EU). Thus, the absolute gain (Δ) grows as resource availability decreases (0.012 → 0.016 → 0.024). A similar but smaller pattern is observed for FBERT. Overall, these results suggest that making clients more multilingual is especially beneficial for lower-resource languages, and helps reduce performance disparities across language groups.

A practical advantage of federated training is that it avoids the need for each participant to fine-tune and maintain its own separate language-specific model. Under our Local FT setup, which uses early stopping with patience 5 and $\delta = 0.001$, training all eight monolingual Local FT models would require a total of 4.814×10^5 optimization steps, whereas the federated configurations require only between 1.12×10^4 and 7.57×10^4 steps. While this is not a perfectly controlled comparison, since Local FT and federated training do not share exactly the same stopping conditions (standard early stop vs LDES-FL), it nevertheless provides a useful estimate of relative training effort. Even the most expensive federated setting is still about 6.4 times cheaper in total training steps than the full set of Local FTs, while the cheapest one is about 43 times cheaper. Furthermore, Table 5 shows that, in every resource group, the global federated model (either FedAvg composition) outperforms the *average* of the monolingual Local FT models. For ROUGE-L, *FedAvg (100% mono)* exceeds *Avg. Local FT (mono)* by +0.019 (H), +0.015 (M), and +0.021 (L), while *FedAvg (15% mono)* increases this margin to +0.031 (H), +0.031 (M), and +0.045 (L). The same trend holds for FBERT. In practical terms, this means that clients can join a collaborative training workflow and obtain a single multilingual model with strong overall performance at a much lower total computational cost than training separate models in isolation. The trade-off is reduced language-specific specialization, but the resulting model is considerably more attractive when multilingual coverage is the primary goal.

Regarding the FL training scheme, the evolution of client participation confirms the proper functioning of our LDES-FL method. A comparison between Figures 2 and 4 shows that client rejoining occurs more frequently in settings where clients hold multilingual data, while it is almost absent in fully monolingual configurations. This behavior aligns with our expectations. In the monolingual client scenario, where both the training and validation data within each client correspond to a single language, local models learn mostly from their own data, yielding limited benefits from cross-client updates. Conversely, as the proportion of multilingual data increases, rejoining events become more frequent, since clients benefit not only from their own multilingual datasets but also from the aggregated updates contributed by other multilingual clients.

7. Conclusions

In this work, we extended *FederatedScope*-LLM to support multilingual federated instruction-tuning of LLMs and introduced Local Dynamic Early Stopping (LDES-FL), a novel stopping criterion that allows clients to pause and resume local training based on their validation loss. This mechanism preserves performance while reducing unnecessary computation, thereby improving training efficiency and sustainability in terms of total optimization steps.

Across all studied client language compositions, ranging from fully monolingual to increasingly multilingual clients, LoRA-based federated fine-tuning consistently improves over the base model, increasing average multilingual performance and fairness. Compared to monolingual local fine-tuning, federated training yields a stronger single multilingual model, while monolingual local fine-tuning remains most effective when the goal is to maximize performance in one target language. At the same time, multilingual local fine-tuning achieves the strongest overall results, and can therefore be interpreted as an upper bound for the federated setting, since it assumes direct access to all multilingual training data without privacy restrictions.

Finally, our results indicate that client language composition is a key design variable in multilingual FL, with direct consequences for performance, fairness and efficiency. More multilingual clients generally improve average multilingual performance and fairness, and particularly benefit lower-resource languages, although at a higher training cost. This finding is consistent with the FL literature on client drift and non-IID data, and is here observed in multilingual federated instruction-tuning of LLMs. Future work should use more strictly controlled multilingual splits to better isolate true cross-lingual transfer.

8. Acknowledgements

Funded by the European Union's Horizon 2020. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or European Commission-EU. Neither the European Union nor the granting authority can be held responsible for them.

9. Ethical Statement

This study uses a multilingual dataset derived from the Alpaca Cleaned corpus, distributed under the CC BY-NC license for academic and research use. No personally identifiable information is present in the dataset and it was anonymized before release. We acknowledge that the multilingual dataset may carry linguistic and cultural biases originating from the source data and automatic translation process.

10. Bibliographical References

Yujun Cheng, Weiting Zhang, Zhewei Zhang, Chuan Zhang, Shengjin Wang, and Shiwen Mao. 2025. Toward Federated Large Language Models: Motivations, Methods, and Future Directions. *IEEE Communications Surveys Tutorials*, 27(4):2733–2764.

Ning Ding, Bing Qin, and Ting Liu. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. In *Findings of the Association for Computational Linguistics (ACL Findings)*.

Aitor Gonzalez-Agirre, Marc Pàmies, Joan Llop, Irene Baucells, Severino Da Dalt, Daniel Tamayo, José Javier Saiz, Ferran Espuña, Jaume Prats, Javier Aula-Blasco, Mario Mina, Iñigo Pikabea, Adrián Rubio, Alexander Shvets, Anna Sallés, Iñaki Lacunza, Jorge Palomar, Júlia Falcão, Lucía Tormo, et al. 2025. Salamandra Technical Report. *CoRR*, abs/2502.08489.

Demi Guo, Alexander M Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhihan Guo, Yifei Zhang, Zhuo Zhang, Zenglin Xu, and Irwin King. 2024. FedLFC: Towards Efficient Federated Multilingual Modeling with LoRA-based Language Family Clustering. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1519–1528. Association for Computational Linguistics.

Gururise. 2023. Cleaned Alpaca Dataset. Accessed: 2025-09-06.

Agrin Hilmkil, Sebastian Callh, Matteo Barbieri, Leon René Sütfeld, Edvin Listo Zec, and Olof Mogren. 2021. Scaling Federated Learning for Fine-Tuning of Large Language Models. In *Natural Language Processing and Information Systems - 26th International Conference on Applications of Natural Language to Information Systems, NLDB 2021, Saarbrücken, Germany, June 23-25, 2021, Proceedings*, volume 12801 of *Lecture Notes in Computer Science*, pages 15–23. Springer.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Brianna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jiahui Hu, Dan Wang, Zhibo Wang, Xiaoyi Pang, Huiyu Xu, Ju Ren, and Kui Ren. 2025. Federated Large Language Model: Solutions, Challenges and Future Directions. *IEEE Wireless Communications*, 32(4):82–89.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2019. Advances and Open Problems in Federated Learning. *CoRR*, abs/1912.04977.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. 2019. SCAFFOLD:

stochastic controlled averaging for on-device federated learning. *CoRR*, abs/1910.06378.

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, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 5260–5271. ACM.

Royson Lee, Minyoung Kim, Fady Rezk, Rui Li, Stylianos I. Venieris, and Timothy M. Hospedales. 2025. FedP^{2}EFT: Federated Learning to Personalize Parameter Efficient Fine-Tuning for Multilingual LLMs. *CoRR*, abs/2502.04387.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. In *Proceedings of the Third Conference on Machine Learning and Systems, MLSys 2020, Austin, TX, USA, March 2-4, 2020*. mlsys.org.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xiao-Yang Liu, Rongyi Zhu, Daochen Zha, Jiechao Gao, Shan Zhong, Matt White, and Meikang Qiu. 2025. Differentially Private Low-Rank Adaptation of Large Language Model Using Federated Learning. *ACM Trans. Manag. Inf. Syst.*, 16(2):1–24.

Zili Lu, Heng Pan, Yueyue Dai, Xueming Si, and Yan Zhang. 2024. Federated Learning With Non-IID Data: A Survey. *IEEE Internet of Things Journal*, 11(11):19188–19209.

Andre Manoel, Mirian del Carmen Hipolito Garcia, Tal Baumel, Shize Su, Jialei Chen, Robert Sim, Dan Miller, Danny Karmon, and Dimitrios Dimitriadis. 2023. Federated Multilingual Models for Medical Transcript Analysis. In *Conference on Health, Inference, and Learning, CHIL 2023, Broad Institute of MIT and Harvard (Merkin Building), 415 Main Street, Cambridge, MA, USA*, volume 209 of *Proceedings of Machine Learning Research*, pages 147–162. PMLR.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. 2016. Federated Learning of Deep Networks using Model Averaging. *CoRR*, abs/1602.05629.

Matías Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. 2022. Local Learning Matters: Rethinking Data Heterogeneity in Federated Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 8387–8396. IEEE.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. 2020. Adaptive Federated Optimization. *CoRR*, abs/2003.00295.

Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. 2018. On the Convergence of Federated Optimization in Heterogeneous Networks. *CoRR*, abs/1812.06127.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. 2023. Towards Personalized Federated Learning. *IEEE Trans. Neural Networks Learn. Syst.*, 34(12):9587–9603.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpaca: A Strong, Replicable Instruction-Following Model. Accessed: 2025-09-06.

Bibek Upadhayay and Vahid Behzadan. 2023. TaCo: Enhancing Cross-Lingual Transfer for Low-Resource Languages in LLMs through Translation-Assisted Chain-of-Thought Processes. *CoRR*, abs/2311.10797.

Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. 2020. Optimizing Federated Learning on Non-IID Data with Reinforcement Learning. In *39th IEEE Conference on Computer Communications, INFOCOM 2020, Toronto, ON, Canada, July 6-9, 2020*, pages 1698–1707. IEEE.

Haoyu Wang, Handong Zhao, Yaqing Wang, Tong Yu, Jiuxiang Gu, and Jing Gao. 2022. FedKC: Federated Knowledge Composition for Multilingual Natural Language Understanding. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1839–1850, New York, NY, USA. Association for Computing Machinery.

Orion Weller, Marc Marone, Vladimir Braverman, Dawn J. Lawrie, and Benjamin Van Durme. 2022. Pretrained Models for Multilingual Federated Learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1413–1421. Association for Computational Linguistics.

Mang Ye, Xiuwen Fang, Bo Du, Pong C. Yuen, and Dacheng Tao. 2024a. Heterogeneous Federated Learning: State-of-the-art and Research Challenges. *ACM Comput. Surv.*, 56(3):79:1–79:44.

Rui Ye, Rui Ge, Xinyu Zhu, Jingyi Chai, Yaxin Du, Yang Liu, Yanfeng Wang, and Siheng Chen. 2024b. FedLLM-Bench: Realistic Benchmarks for Federated Learning of Large Language Models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Han Zhang et al. 2023. Multilingual Parameter-Efficient Fine-tuning of Large Language Models. *arXiv preprint arXiv:2305.12345*.

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, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wanru Zhao, Yihong Chen, Royson Lee, Xinchi Qiu, Yan Gao, Hongxiang Fan, and Nicholas D. Lane. 2025. Breaking Physical and Linguistic Borders: Multilingual Federated Prompt Tuning for Low-Resource Languages. *CoRR*, abs/2507.03003.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated Learning with Non-IID Data. *CoRR*, abs/1806.00582.

Jiaying Zheng, Hainan Zhang, Lingxiang Wang, Wangjie Qiu, Hong-Wei Zheng, and Zhi Ming Zheng. 2024. Safely Learning with Private Data: A Federated Learning Framework for Large Language Model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 5293–5306. Association for Computational Linguistics.