Less is *Fed* More: Sparsity Reduces Feature Distortion in Federated Learning

Aashiq Muhamed[*1], **Harshita Diddee**[*1], **Abhinav Rao**[*1]

[1]Language Technologies Institute, Carnegie Mellon University
[*]Equal contribution
{amuhamed, hdiddee, abhinavr}@andrew.cmu.edu

Abstract

Our work studies Multilingual Federated Learning (FL), a decentralized paradigm that, although promising, grapples with issues such as client drift and suboptimal generalization in diverse, multilingual settings. We highlight limitations in existing approaches to generalize across both actively participating and inactive client language pairs. To mitigate these challenges, we introduce FedSparseNet, which incorporates sparse-network training, and LoRA, based on Low-Rank Adaptation. These approaches maintain the model's fidelity to its pretraining distribution, thereby ensuring robust performance on both seen and unseen language pairs, while simultaneously enhancing communication efficiency by selectively transmitting trainable parameters. Our empirical evaluations demonstrate that FedSparseNet outperforms conventional FL models on both seen and unseen clients, while LoRA shows remarkable improvements in unseen client performance. Additionally, we propose the Continuous Relative Robustness Metric, a novel metric to uniformly assess a model's performance across diverse language pairs. We open-source our code for reproducibility on GitHub.[1]

1 Introduction

The development of NLP applications capable of leveraging multilingual, multi-source, heterogeneous data while safeguarding user privacy is essential (Deng et al., 2022). FL (McMahan et al., 2016) addresses this by facilitating the utilization of personally identifiable information within a decentralized framework, thereby obviating the need for direct data sharing among clients. However, FL faces challenges such as client drift and suboptimal generalization in heterogeneous environments (Karimireddy et al., 2020). Furthermore, multilingual FL not only contends with these FL-specific optimization difficulties but also grapples with the complexities of extending to low-resource languages. This can hinder the accessibility of language technologies for various communities and intensify systemic biases (Santy et al., 2023).

While there is extensive research on FL for NLP, studies specifically addressing multilingual FL translation remain limited, with minimal exploration of how FL impacts the training process. Multilingual FL is an inherently heterogeneous data setting, offering a unique area of interest within the FL community. The closest work is Weller et al. (2022b), where the authors investigate Federated Multilingual Translation. The study involves fine-tuning and communicating the entire parameter set of a 418M M2M encoder-decoder model. Their findings suggest that fine-tuning a pre-trained model using FL can achieve comparable results to centralized learning, even in Non-IID settings with clients segmented by language.

In our research, we challenge the prevailing narrative that communicating all parameters in a multilingual translation model is viable for practical translation tasks. We argue that this approach is largely impractical. Moreover, translation applications require the server model to not only generalize to client language pairs actively involved in FL but also to maintain pretraining performance on unseen language pairs or inactive clients. Our findings reveal that baseline performance for unseen language pairs declines when fine-tuning with active client data. This issue stems from the distortion of pretrained features (Kumar et al., 2022), a problem not adequately addressed by current FL approaches, especially in the context of NLP tasks like translation. To address the challenges identified, our approach builds on the current literature on Parameter Efficient Finetuning (PEFT) to: a) ensure the model remains close to its pretraining distribution, facilitating balanced generalization across both seen and unseen language pairs, and b) enhance federated fine-tuning and communication

[1]https://github.com/AetherPrior/less-is-fed-more

Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U), pages 37–46

efficiency by transmitting only a sparse subset of trainable parameters. Our contributions include:

• We propose **FedSparseNet**, leveraging sparse-network training, and employing Low-Rank Adaptation (**LoRA**) to mitigate pretrained feature distortion and enhance communication efficiency. FedSparseNet dominates the corresponding fully finetuned FL baseline on client-seen and client-unseen performance (by 1.4 BLEU), while LoRA significantly improves the client-unseen performance but falls short on seen-client performance.

• We propose the **Continuous Relative Robustness Metric**, a metric that measures how well a given model uniformly dominates the pretrained model on **both**, seen and unseen language pairs.

2 Methodology

2.1 FedSparseNet: Composable Sparse Fine-tuning for FL

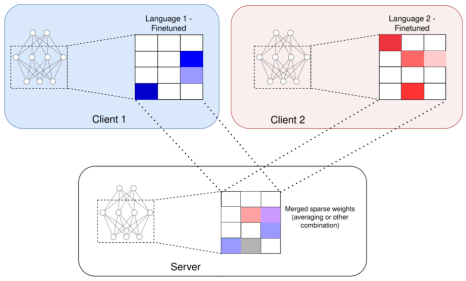

Figure 1: The FedSparseNet framework

We propose a variant of the Lottery Ticket Algorithm for federated training called FedSparseNet. Our work is inspired by Lottery Ticket Sparse Fine-Tuning (LT-SFT) for cross-lingual transfer (Frankle and Carbin, 2018). The Lottery Ticket Hypothesis (LTH) (Frankle and Carbin, 2018) states that each neural model contains a sub-network (a "winning ticket") that, if trained again in isolation, can match or even exceed the performance of the original model. To recover this ticket, the sparse ticket is selected using a pruning stage where some parameters are zero-masked and frozen according to some criterion (e.g., weight magnitude), and the remaining parameters are restored to their original values and then re-tuned. This process of pruning and re-training can be iterated multiple times.

FedSparseNet (Fig. 1) consists of two stages on the client. Let i denote the i-th round of training and $\theta^{(i)}$, the server model parameters at round (i).

(Stage 1) This phase is only applicable at i=1. Let $\theta_0^{(1)}$ represent the pretrained (client) model parameters, and $\theta_1^{(1)}$, the parameters after fine-tuning on the target language or task data D. The parameters are ranked according to the greatest absolute difference $|\theta_0^{(1)} - \theta_1^{(1)}|$, and the top K are selected for subsequent tuning. A binary mask μ is set to have 1 in positions corresponding to these parameters, and 0 elsewhere. This mask state is frozen and preserved for each client across rounds.

(Stage 2) If we are at round 1, the parameters are reset to their original values $\theta_0^{(1)}$, and at any other round, we use the server checkpoint $\theta_s^{(i)}$. The model is again fine-tuned, but this time, only the K-selected parameters using the mask μ are trainable, whereas the others are kept frozen. This is implemented by using the masked gradient $\mu \odot \nabla_\theta L(F(\cdot;\theta), D)$ (where $\odot$ denotes element-wise multiplication and L a loss function) in the optimizer at each step. If we denote the sparse finetuned checkpoint as $\theta_2^{(i)}$, only the sparse vector of differences $\theta_2^{(i)} - \theta_s^{(i)}$ is communicated at every round. The sparse vectors from every client are then aggregated at the server using an aggregation strategy like FedAvg before being broadcasted to clients in the next round.

FedSparseNet enhances communication efficiency by minimizing data transmission which is often about 1% of the client parameters. The modular design allows for effective composability, reducing knowledge overlap and interference among the client languages. Sparsity also serves as a natural form of regularization, making these networks less prone to overfitting, and helping the model retain generalization properties of the pretrained model on unseen data. Sparse networks also have other advantages: it does not introduce additional parameters like the adapter (Houlsby et al., 2019), thereby not reducing inference speed; and the model architecture remains identical to the pretrained model, simplifying code development and ensuring the method is model-agnostic.

2.2 LoRA

We also propose to use Low-Rank Approximation (Hu et al., 2021), as a parameter-efficient client optimization technique that maintains compositionality and proximity to the pretrained weights.

Low Rank Approximation or LoRA encodes the parameter updates of a model undergoing finetuning in a much smaller subspace. Specifically, for a model $P_\Phi(y|x)$ parameterized by Φ, the typical model finetuning would involve updating the entire

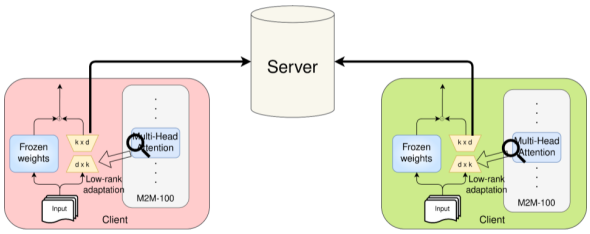

Figure 2: The LoRA framework

parameter space according to:

$$\max_{\Phi} \sum_{(x,y)\in\mathcal{Z}} \sum_{t=1}^{|y|} \log\left(P_{\Phi}(y_t|x, y_{<t})\right) \quad (1)$$

LoRA hypothesizes the existence of a low-rank approximation of the parameter updates, and posits that the full rank update, denoted by $\Delta\Phi$ can be approximated by a much lower rank matrix $\Delta\Phi(\theta)$. In other words, Φ can be expressed as $\Phi_0 + \Delta\Phi(\theta)$.

Several works have studied combining LoRA with Federated learning. Qi et al. (2024) study the use of LoRA for LLM personalization; however, they do not freeze the model's layers during training, thereby compromising on efficiency. We instead maintain efficiency to be our core-focus similar to the works of Zhang et al. (2024); Ye et al. (2024); Kuang et al. (2023). During training, we instantiate each client with LoRA modules of the same rank. In the first iteration, this implies injecting LoRA modules into the pretrained model. During finetuning, we freeze all other parameters but the LoRA modules and subsequently communicate LoRA modules to the server for aggregation. The reduction in parameter update space brought by LoRA, brings significant memory reduction while training with large models, which is advantageous in the FL setting.

2.3 Continuous Relative Robustness Metric for Federated Learning Models

In this work, we employ a fixed model selection strategy on the clients to optimize for client-seen performance. We propose modeling enhancements to improve performance on client-unseen data while retaining performance on client-seen data. To select among the models that perform better than the baseline on both client-seen and unseen data, we propose a new robustness metric to balance performance (in BLEU) on client-seen and client-unseen data. Given a model M and a pre-trained model M_{pre}, we consider a continuous range of trade-off coefficients, $k \in [0, 1]$, to evaluate the balance between client-seen (CS) and client-unseen (CU) performance metrics. The performance metric $P(M, k)$ for a model M is defined over the continuous domain as:

$$P(M, k) = k \cdot \text{perf}_{\text{CS}}(M) + (1 - k) \cdot \text{perf}_{\text{CU}}(M)$$

Relative Robustness Score The relative robustness of model M against the pre-trained model M_{pre} is quantified by integrating the performance advantage of M over M_{pre} across the continuous range of k:

$$RRS(M) = \int_0^1 \mathbf{1}\{P(M, k) > P(M_{\text{pre}}, k)\}\, dk$$

Here, $\mathbf{1}\{\}$ is the indicator function, which is 1 when M outperforms M_{pre} at a given k and 0 otherwise. The integral effectively counts the proportion of the trade-off range where M surpasses M_{pre}. This metric compares FL models in balancing client-seen and client-unseen performance over a continuum.

Language Pair	ISO 639-2 codes	Dataset Source
Client-Seen Languages		
English - French	En-Fr	UNMT corpus
Arabic-Spanish	En-Fr	UNMT corpus
Russian-Chinese	Ru-Zh	UNMT corpus
Client-Unseen Languages - High Resource		
Portuguese-English	Pt-En	FLORES-200
Hindi-English	Hi-En	FLORES-200
Korean-English	Ko-En	FLORES-200
Client-Unseen Languages - Mid Resource		
Tamil-English	Ta-En	FLORES-200
Ukrainian-English	Uk-En	FLORES-200
Finnish-English	Fi-En	FLORES-200
Client-Unseen Languages - Low Resource		
Swahili-English	Sw-En	FLORES-200
Sinhala-English	Si-En	FLORES-200
Malayalam-English	Ml-En	FLORES-200

Table 1: All Language Pairs used in our experiments. We mimic the setup from Weller et al. (2022b) for client-seen language pairs, and pick 9 language pairs from FLORES-200 for our client-unseen languages, based on M2M-100's pretraining distribution.

2.4 Experimental Details

We choose machine translation for all our base tasks and define 'seen' and 'unseen' language-pairs as those pairs that are visible or invisible to the client model during finetuning. We use

Language Pair	Pretrained	Centralized	IID FL	Non-IID FL	FedSparseNet (Non-IID)	FedSparseNet (IID)	LoRA (Non-IID)	LoRA (IID)
Client-Seen Languages								
En-Fr	31.8 ± 0.6	38.0 ± 0.7	37.7 ± 0.7	36.9 ± 0.7	38.6 ± 0.7	38.8 ± 0.7	36.0 ± 0.6	36.2 ± 0.6
Ar-Es	28.0 ± 0.5	35.5 ± 0.7	35.9 ± 0.7	32.9 ± 0.6	36.4 ± 0.7	36.5 ± 0.6	33.4 ± 0.6	33.2 ± 0.6
Ru-Zh	30.3 ± 0.5	37.5 ± 0.6	37.7 ± 0.4	38.7 ± 0.6	37.7 ± 0.7	38.0 ± 0.7	34.3 ± 0.6	34.6 ± 0.6
Avg	30.0 ± 0.5	37.0 ± 0.7	37.1 ± 0.6	36.2 ± 0.6	37.6 ± 0.7	37.8 ± 0.7	34.5 ± 0.6	34.6 ± 0.6
Client-Unseen Languages - High Resource								
Pt-En	40.0 ± 1.1	31.8 ± 1.1	32.0 ± 1.0	26.7 ± 1.2	32.2 ± 1.3	34.9 ± 1.2	39.5 ± 1.1	39.5 ± 1.2
Hi-En	29.6 ± 1.0	22.0 ± 1.1	22.8 ± 0.9	19.3 ± 1.0	21.8 ± 1.3	25.1 ± 1.1	28.3 ± 1.0	28.9 ± 1.0
Ko-En	20.5 ± 0.9	15.0 ± 0.9	14.4 ± 0.9	13.0 ± 0.8	14.5 ± 1.0	16.7 ± 0.9	19.6 ± 0.9	20.0 ± 1.0
Avg	30.0 ± 1.0	22.9 ± 1.0	23.1 ± 0.9	19.7 ± 1.0	22.8 ± 1.2	25.6 ± 1.0	29.1 ± 1.0	29.4 ± 1.1
Client-Unseen Languages - Mid Resource								
Ta-En	8.0 ± 0.6	3.9 ± 0.4	5.0 ± 0.5	3.7 ± 0.4	1.6 ± 0.2	4.7 ± 0.5	9.2 ± 0.7	8.6 ± 0.7
Uk-En	27.9 ± 1.0	18.2 ± 1.0	21.8 ± 0.9	20.7 ± 1.0	21.2 ± 1.2	23.8 ± 0.9	28.2 ± 1.0	27.8 ± 1.0
Fi-En	25.7 ± 1.0	18.2 ± 1.0	18.8 ± 0.8	14.4 ± 1.0	18.8 ± 1.1	21.0 ± 0.9	25.0 ± 1.0	24.9 ± 0.9
Avg	20.5 ± 0.9	13.4 ± 0.8	15.2 ± 0.7	12.9 ± 0.8	13.9 ± 0.8	16.5 ± 0.8	20.8 ± 0.9	20.4 ± 0.8
Client-Unseen Languages - Low Resource								
Sw-En	26.0 ± 0.9	17.2 ± 1.0	18.4 ± 1.0	13.6 ± 1.0	15.0 ± 1.1	21.0 ± 1.0	24.4 ± 1.0	24.8 ± 1.0
Si-En	15.9 ± 0.8	8.8 ± 0.7	9.6 ± 0.8	7.3 ± 0.7	6.1 ± 0.7	10.9 ± 0.8	15.1 ± 0.8	14.9 ± 0.9
Ml-En	15.3 ± 0.9	8.0 ± 0.8	8.6 ± 0.8	6.3 ± 0.6	5.5 ± 0.6	10.3 ± 0.8	14.3 ± 0.8	15.0 ± 0.9
Avg	19.1 ± 0.9	11.3 ± 0.8	12.2 ± 0.9	9.1 ± 0.8	8.9 ± 0.8	14.0 ± 0.9	17.9 ± 0.9	18.2 ± 0.9
Weighted Metric Calculation								
RRS	0.000	0.488	0.525	0.397	0.485	0.632	0.897	0.882

Table 2: UN-MT Bleu for Client-Seen and Client-Unseen Language Pairs. FedSparseNet uses sparsity ratio 0.01 on embedding matrix. LoRA trained with rank 8, on embedding matrices. All models are trained for 1 epoch/round.

the M2M100-418M model (Fan et al., 2020) as our base, UN parallel corpus (which we term as UNMT) (Ziemski et al., 2016) for finetuning, FLORES-200 (Costa-jussà et al., 2022) for evaluation and report performance using BLEU (Papineni et al., 2002). All client models are trained for 100 rounds, and the best model is selected based on the local validation loss. We choose our seen language-pairs similar to that of Weller et al. (2022b), and pick 9 unseen language pairs (3 from High, Middle and Low resource languages respectively) from FLORES-200, based on M2M-100's pretraining distribution. We choose English to be our target language for simplicity in evaluation and comparison. Table 1 presents all of our language pairs and their respective ISO-693-2 codes, which we shall use from here on. Additional details on training dataset and metrics can be found in Appendix A.1. We conduct all experiments over three settings: standard finetuning of the base model without any federation (the *Centralized* setting), FL on IID data (*IID FL*), where all three language pairs are uniformly mixed and distributed across clients, and FL on non-IID or heterogeneous data (*Non-IID FL*), where each client receives a separate language pair for training. We use `FedAvg` (McMahan et al., 2016) as our aggregation algorithm.

3 Results

Table 2 compares our approach with the baseline (Weller et al., 2022b): the performance of the federated fully FT models on unseen-client data shows a significant drop in performance relative to the Pretrained model on all client-unseen language pairs.

FedSparseNet FedSparseNet dominates the corresponding baselines across seen and unseen client datasets (Table 2), demonstrating their overall effectiveness. Interestingly, no significant trend is observed across High-, Mid-, and Low-Resource languages. We also note that while FedSparseNet (IID) and FedSparseNet (Non-IID) achieve similar performance on client-seen data, the latter exhibits significantly lower performance on unseen data, especially for Low-Resource languages. This suggests that Non-IID FL potentially distorts pretrained features more than IID-FL, impacting performance in ways not captured by client-seen accuracy alone. Consistent with these observations, the RRS metric reveals a higher value for FedSparseNet in the IID setting compared to the Non-IID setting. This highlights the effectiveness of FedSparseNet in scenarios with balanced and representative data distributions (IID).

LoRA In Table 2, we compare LoRA with the baseline. LoRA demonstrates its highest efficacy on unseen languages, effectively minimizing the distortion introduced by optimization on seen-client data during federated finetuning. This capability to recover unseen client performance can be attributed to the inherent regularizing effect of LoRA on the distribution of the federated model.

The strengths of LoRA are further illuminated by its superior performance in the RRS metric — that endorses LoRA as a more viable alternative than full FT and FedSparseNet, for achieving balanced improvements across seen and unseen language pairs. However, it is imperative to approach these results with caution. LoRA's performance on seen clients, in both IID and Non-IID settings, falls short of the centralized model and FedSparseNet. This observed degradation suggests possible shortcomings in LoRA's ability to effectively compose client knowledge across diverse heterogeneous datasets. While FedSparseNet also appears to benefit from its approach of localizing seen-language-specific information through strategic subnet selection—a method documented to personalize and compose well across clients (Ansell et al., 2021), LoRA may encounter challenges in achieving a similar level of integration, particularly due to interference between client-specific modules during federated optimization.

Comparing Communication Efficiency We compare both methods with the baselines for communication efficiency up to the point of convergence in Appendix A.2. We observe a *54x* and *5.9x* increase in communication efficiency for FedSparseNet and LoRA respectively.

4 Conclusion

Motivated by the need to improve generalization in FL for unseen client data, we introduce FedSparseNet and LoraFed. These methods focus on sparsifying the client parameter space, addressing the challenge of pretrained feature distortion due to seen-client optimization, and enhancing communication efficiency.

References

Alan Ansell, Edoardo Maria Ponti, Anna Korhonen, and Ivan Vulić. 2021. Composable sparse fine-tuning for cross-lingual transfer. *arXiv preprint arXiv:2110.07560*.

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. 2023. When is multilinguality a curse? language modeling for 250 high- and low-resource languages. *ArXiv*, abs/2311.09205.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Jieren Deng, Chenghong Wang, Xianrui Meng, Yijue Wang, Ji Li, Sheng Lin, Shuo Han, Fei Miao, Sanguthevar Rajasekaran, and Caiwen Ding. 2022. A secure and efficient federated learning framework for nlp.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. 2023. Sparse low-rank adaptation of pre-trained language models. In *Conference on Empirical Methods in Natural Language Processing*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation.

Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.

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

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

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR.

Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. 2023. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*.

H. B. McMahan, Eider Moore, Daniel Ramage, S. Hampson, and B. A. Y. Arcas. 2016. Communication-efficient learning of deep networks from decentralized data. *International Conference on Artificial Intelligence and Statistics*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jiaxing Qi, Zhongzhi Luan, Shaohan Huang, Carol Fung, Hailong Yang, and Depei Qian. 2024. Fdlora: Personalized federated learning of large language model via dual lora tuning.

Sebastin Santy, Jenny Liang, Ronan Le Bras, Katharina Reinecke, and Maarten Sap. 2023. NLPositionality: Characterizing design biases of datasets and models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9080–9102, Toronto, Canada. Association for Computational Linguistics.

Orion Weller, Marc Marone, Vladimir Braverman, Dawn Lawrie, and Benjamin Van Durme. 2022a. Pre-trained 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*, pages 1413–1421, Seattle, United States. Association for Computational Linguistics.

Orion Weller, Marc Marone, Vladimir Braverman, Dawn J Lawrie, and Benjamin Van Durme. 2022b. Pretrained models for multilingual federated learning. *ArXiv*, abs/2206.02291.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. 2024. Openfedllm: Training large language models on decentralized private data via federated learning.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Yufan Zhou, Guoyin Wang, and Yiran Chen. 2024. Towards building the federated gpt: Federated instruction tuning.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

A Appendix

A.1 Task Experimental Details

The UN corpus contains written records of the UN proceedings from 1990-2014. For seen-languages, we consider training, validation, and tests sets for the same source and target language pairs as described in Weller et al. (2022a), namely `(En-Fr)`, `(Ar-Es)`, and `(Ru-Zh)`, sampling 10k training examples and 5k testing examples for each. For client-unseen languages, we consider the FLORES-200 (Costa-jussà et al., 2022) dataset. FLORES-200 consists of 3001 parallel sentences manually translated across 200 different languages. We choose its `devtest` subset, with 1013 sentences for each language. We consider 9 different source languages, choosing 3 across high-resource (Portuguese `(Pt)`, Hindi `(Hi)`, Korean `(Ko)`), mid-resource (Tamil `(Ta)`, Ukranian `(Uk)`, Finnish `(Fi)`), and low-resource (Swahili `(Sw)`, Sinhalese `(Si)`, Malayalam `(Ml)`) settings each. For ease of evaluation and comparison, we fix the target language to English, leading to 9 `(X-En)` language pairs, where `X` represents our source language.

Metrics and Model Selection We evaluate and report client-seen and client-unseen performance using BLEU (Papineni et al., 2002). We use the standard sacreBLEU settings (`nrefs:1, mixed case, eff:no, tok:13a, smooth:exp,` and `version 2.0.0`). For Ja and Zh we use their respective tokenizers. All client models are trained for 100 rounds, and the best model is selected based on the local validation loss. To select among models that perform better than the corresponding fully finetuned baselines we use the RRS defined in Section 2.3.

Compute We train each model on a configuration of 3 A6000 GPUs. The baselines reach convergence in under 12 hours. FedSparseNet and LoRA exhibit slightly faster training times.

A.2 Communication Efficiency

To assess the communication efficiency of a model, we consider the total volume of data (in bytes) transmitted across clients until the model reaches its optimal state, as indicated by its best checkpoint. This efficiency over n rounds until convergence can be formulated as: trainable_params $\times$ num_clients $\times$ $n \times 2$. The factor of 2 accounts for the bidirectional communication between the server and all clients at both the beginning and the end of each round. Figure 3 and 4 show the communication efficiency

curves for the methods.

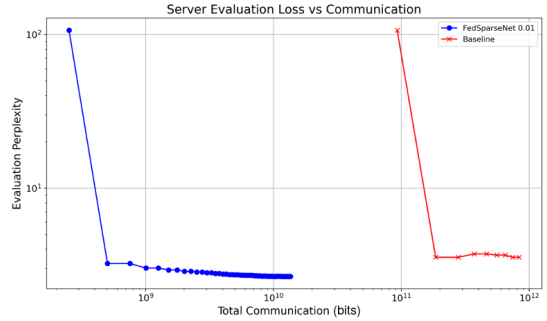

Figure 3: FedSparse 0.01 vs Full FT communication efficiency.

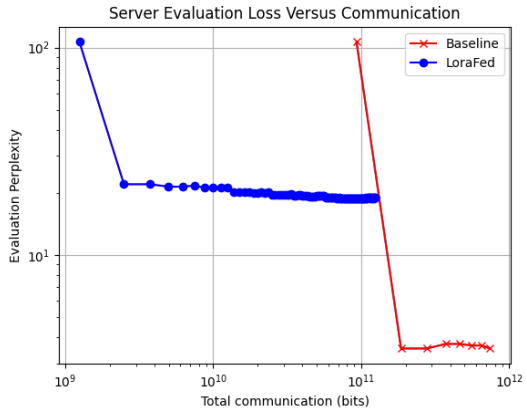

Figure 4: Communincation Overhead reduction in LoRA

A.3 FedSparse Ablations

Where and how to apply FedSparseNet?

In Table 3, we conduct a series of ablation studies to evaluate the impact of varying the target module for sparsity application as well as the sparsity ratio within the FedSparseNet framework. Specifically, FedSparseNet (0.01) denotes the application of a sparse mask with a 0.01 sparsity ratio to the tied embeddings (encoder and decoder) of the M2M model. Our comparative analysis between FedSparseNet (1.0) and FedSparseNet (1.0) + Body (0.01) reveals that applying a sparse mask to the tied embeddings layer yields superior performance on both client-seen and client-unseen data compared to applying the mask to the Body of the M2M model. This could be attributed to the reduced feature distortion achieved through sparsity in the embedding layers (Kumar et al., 2022). Furthermore, our findings indicate that FedSparseNet (0.0) + Body (0.01) outperforms FedSparseNet (1.0) + Body (0.01) in the RRS metric. This suggests that a higher sparsity ratio applied to the body of the model might further constrain feature distortion, enhancing the model's performance.

When examining the optimal degree of sparsity to apply, we observed that FedSparseNet configurations with varying sparsity ratios (0.01, 0.1, and 1.0) delivered comparable performances on client-unseen data. FedSparseNet (0.01) emerged as the most efficient model overall in terms of RRS and communication efficiency. Introducing a regularization penalty to FedSparseNet (0.01) with a λ 0.1 did not result in statistically significant differences in performance on both client-seen and client-unseen data.

What is the Impact of Increasing Local Work for FedSparseNet? In Table 4 in A, we compare FedSparseNet and the baselines when each model is trained for 5 epochs/round. We observe that increasing local work generally amplifies pretrained feature distortion for both baselines and FedSparseNet. Consequently, the performance of IID FL and FedSparseNet (Non-IID FL and IID) deteriorates compared to Table 2. While FedSparseNet (IID) outperforms IID FL on both seen and unseen client performance, a surprising trend emerges for the Non-IID FL baseline. The model trained with local work exhibits performance comparable to the 1-epoch/round baseline on seen data, but surpasses it on unseen data, with increasing gains observed in HRL, followed by MRL and LRL. While FedSparseNet still achieves better client-seen data generalization than Non-IID FL, it lags behind on client-unseen data and the RRS metric. This suggests that the sparsity mechanism in FedSparseNet might hinder its ability to fully exploit the benefits of increased local work for unseen data. This is particularly relevant for low-resource languages characterized by limited training data and potentially weaker local data signals.

Takeaways

1. When examining the optimal degree of sparsity to apply, we observed that FedSparseNet configurations with varying sparsity ratios (0.01, 0.1, and 1.0) delivered comparable performances on client-unseen data. FedSparseNet (0.01) emerged as the most efficient model overall in terms of RRS and communication efficiency.

2. Introducing a regularization penalty to

Language Pair	FedSparseNet (0.01)	FedSparseNet (0.1)	FedSparseNet (1.0)	FedSparseNet+Reg (0.01)	FedSparseNet (1.0)+Body(0.01)	FedSparseNet (0.0)+Body(0.01)
Client-Seen Languages						
En-Fr	38.6 ± 0.7	38.7 ± 0.7	38.6 ± 0.7	38.6 ± 0.7	35.1 ± 0.7	35.9 ± 0.7
Ar-Es	36.4 ± 0.7	36.4 ± 0.6	36.5 ± 0.7	36.4 ± 0.6	29.6 ± 0.6	33.0 ± 0.6
Ru-Zh	37.7 ± 0.7	37.7 ± 0.6	37.6 ± 0.6	37.7 ± 0.6	38.7 ± 0.6	38.0 ± 0.6
Avg	37.6 ± 0.7	37.6 ± 0.6	37.6 ± 0.6	37.6 ± 0.6	34.5 ± 0.6	35.6 ± 0.6
Client-Unseen Languages - High Resource						
Pt-En	32.2 ± 1.3	31.6 ± 1.5	32.0 ± 1.3	31.8 ± 1.4	27.7 ± 1.0	29.2 ± 1.0
Hi-En	21.8 ± 1.3	21.2 ± 1.3	21.8 ± 1.3	21.3 ± 1.4	19.1 ± 0.9	20.2 ± 0.9
Ko-En	14.5 ± 1.0	14.1 ± 1.0	14.1 ± 1.0	14.5 ± 1.1	12.4 ± 0.7	13.8 ± 0.8
Avg	22.8 ± 1.2	22.3 ± 1.3	22.6 ± 1.2	22.5 ± 1.3	19.7 ± 0.9	21.1 ± 0.9
Client-Unseen Languages - Mid Resource						
Ta-En	1.6 ± 0.2	1.6 ± 0.2	1.7 ± 0.2	1.6 ± 0.2	4.1 ± 0.5	4.4 ± 0.5
Uk-En	21.2 ± 1.2	20.7 ± 1.1	21.5 ± 1.1	21.2 ± 1.2	18.4 ± 0.9	19.1 ± 1.0
Fi-En	18.8 ± 1.1	18.1 ± 1.1	18.4 ± 1.1	18.7 ± 1.1	14.7 ± 0.7	16.6 ± 1.0
Avg	13.9 ± 0.8	13.5 ± 0.8	13.9 ± 0.8	13.8 ± 0.8	12.4 ± 0.7	13.4 ± 0.8
Client-Unseen Languages - Low Resource						
Sw-En	15.0 ± 1.1	14.6 ± 1.1	14.4 ± 1.1	15.2 ± 1.1	14.8 ± 0.8	15.0 ± 1.0
Si-En	6.1 ± 0.7	6.2 ± 0.7	6.0 ± 0.7	6.3 ± 0.7	7.7 ± 0.7	8.1 ± 0.8
Ml-En	5.5 ± 0.6	5.6 ± 0.6	5.2 ± 0.6	5.5 ± 0.6	7.1 ± 0.7	7.6 ± 0.7
Avg	8.9 ± 0.8	8.8 ± 0.8	8.5 ± 0.8	9.0 ± 0.8	9.9 ± 0.7	10.2 ± 0.8
Weighted Metric Calculation						
RRS	0.485	0.477	0.481	0.484	0.328	0.403

Table 3: Different FedSparseNet configurations on non-IID FL are compared. We report BLEU for Client-Seen and Client-Unseen Language Pairs.

FedSparseNet (0.01) with a λ 0.1 did not result in statistically significant differences in performance on both client-seen and client-unseen data.

3. The impact of varying local work needs deeper investigation: Sparsification induced by FedSparseNet might be limiting the efficacy of local work for FedSparse.

B LoRA Ablations

Where and how to apply LoRA ? We explore the candidates for two critical LoRA hyperparameters: rank and its target modules to understand the ideal composition of target location and capacity for the sparsification we induce.

LoRA Rank The approximation rank in LoRA is a critical hyperparameter that governs the reduction in the projection we carry with the gradient updates. We experimented with 2 LoRA ranks: 8 and 32. Table 5 summarizes LoRA's performance with these: 8 and 32. In our experiments, the increase in rank shows a marginal improvement with the numbers though we include even greater ranges for sweeping over ranks in our future work. We posit that the lack of any significant improvement in the capacity of the model could be attributed to the need for differential language-specific capacity i.e., it is possible that languages belonging to different categories (seen or unseen, high-resource or low-resource) may require different rank attributed capacities as has been explored in multilingual literature like Chang et al. (2023) and since we train with a uniform rank, we may be under-allocating or over-allocating capacity specifically to the seen clients. Recent work like Ding et al. (2023) also highlights an important caveat of LoRA is training with a fixed rank (for the entirety of the model's training) which could also be impeding LoRA's efficacy.

LoRA Target Modules We explore applying LoRA to (a) all layers (Key and Query projections) and (b) Input Embedding of the models. We notice a significant improvement in the performance with the use of embedding projections (in alignment with our observation in FedSparse). We posit that the perturbation induced by applying LoRA to all the layers is either too extreme (we see a drop in performance even on the seen clients) or not

Language Pair	Pretrained	Centralized	IID FL	Non-IID FL	FedSparseNet (Non-IID FL)	FedSparseNet (IID)
Client-Seen Languages						
En-Fr	31.8 ± 0.6	38.0 ± 0.7	36.3 ± 0.7	33.1 ± 0.6	38.6 ± 0.7	38.5 ± 0.7
Ar-Es	28.0 ± 0.5	35.5 ± 0.7	35.6 ± 0.7	36.7 ± 0.6	36.3 ± 0.6	36.4 ± 0.6
Ru-Zh	30.3 ± 0.5	37.5 ± 0.6	37.4 ± 0.6	39.2 ± 0.6	37.3 ± 0.6	37.9 ± 0.6
Avg	30.0 ± 0.5	37.0 ± 0.7	36.4 ± 0.7	36.3 ± 0.6	37.4 ± 0.6	37.6 ± 0.6
Client-Unseen Languages - High Resource						
Pt-En	40.0 ± 1.1	31.8 ± 1.1	20.7 ± 1.0	34.6 ± 1.1	32.3 ± 1.2	32.7 ± 1.4
Hi-En	29.6 ± 1.0	22.0 ± 1.1	14.1 ± 0.9	25.5 ± 0.9	21.8 ± 1.3	24.0 ± 1.1
Ko-En	20.5 ± 0.9	15.0 ± 0.9	9.3 ± 0.7	17.5 ± 0.9	14.6 ± 1.1	15.6 ± 0.9
Avg	30.0 ± 1.0	22.9 ± 1.0	14.7 ± 0.9	25.9 ± 1.0	22.9 ± 1.2	24.1 ± 1.1
Client-Unseen Languages - Mid Resource						
Ta-En	8.0 ± 0.6	3.9 ± 0.4	2.5 ± 0.3	7.0 ± 0.7	2.2 ± 0.3	4.2 ± 0.5
Uk-En	27.9 ± 1.0	18.2 ± 1.0	13.1 ± 0.9	24.9 ± 1.0	21.4 ± 1.1	22.1 ± 1.1
Fi-En	25.7 ± 1.0	18.2 ± 1.0	10.0 ± 0.8	21.8 ± 0.9	18.9 ± 1.0	19.5 ± 1.0
Avg	20.5 ± 0.9	13.4 ± 0.8	8.5 ± 0.7	18.2 ± 0.5	14.2 ± 0.8	15.3 ± 0.9
Client-Unseen Languages - Low Resource						
Sw-En	26.0 ± 0.9	17.2 ± 1.0	9.4 ± 0.8	20.1 ± 1.0	16.1 ± 1.1	19.6 ± 0.9
Si-En	15.9 ± 0.8	8.8 ± 0.7	4.5 ± 0.5	12.5 ± 0.9	7.3 ± 0.7	10.3 ± 0.8
Ml-En	15.3 ± 0.9	8.0 ± 0.8	4.3 ± 0.4	12.0 ± 0.8	6.6 ± 0.7	9.6 ± 0.8
Avg	19.1 ± 0.9	11.3 ± 0.8	6.1 ± 0.6	15.0 ± 0.9	10.0 ± 0.8	13.2 ± 0.8
Weighted Metric Calculation						
RRS	0.000	0.496	0.323	0.643	0.497	0.573

Table 4: UN-MT Bleu for Client-Seen and Client-Unseen Language Pairs. FedSparseNet uses sparsity ratio 0.01. All models are trained for 5 epochs/round.

coupled with the right rank (may require a lower rank) to achieve optimal results. Our best model eventually used the model where embeddings were perturbed by LoRA.

Takeaways

1. Applying LoRA to the embedding layer gives significant gains over perturbing the Key and Query projections.

2. Increasing Rank over a limited range [8-32] does not induce a statistically significant improvement in performance.

Language Pair	Pretrained	Centralized	IID FL	Non-IID FL	LoRA (embedding, rank=8)	LoRA (embedding, rank=32)	LoRA (k,q), rank=8
Client-Seen Languages							
En-Fr	31.8 ± 0.6	38.0 ± 0.7	37.7 ± 0.7	36.9 ± 0.7	36.0 ± 0.6	36.4 ± 0.6	35.8 ± 0.6
Ar-Es	28.0 ± 0.5	35.5 ± 0.7	35.9 ± 0.7	32.9 ± 0.6	33.4 ± 0.6	33.2 ± 0.6	32.4 ± 0.6
Ru-Zh	30.3 ± 0.5	37.5 ± 0.6	37.7 ± 0.4	38.7 ± 0.6	34.3 ± 0.6	34.7 ± 0.6	33.2 ± 0.6
Avg	30.0 ± 0.5	37.0 ± 0.7	37.1 ± 0.6	36.2 ± 0.6	34.6 ± 0.6	34.8 ± 0.6	33.8 ± 0.8
Client-Unseen Languages - High Resource							
Pt-En	40.0 ± 1.1	31.8 ± 1.1	32.0 ± 1.0	26.7 ± 1.2	39.5 ± 1.1	39.5 ± 1.1	38.5 ± 1.1
Hi-En	29.6 ± 1.0	22.0 ± 1.1	22.8 ± 0.9	19.3 ± 1.0	28.3 ± 1.0	28.7 ± 1.0	28.3 ± 1.0
Ko-En	20.5 ± 0.9	15.0 ± 0.9	14.4 ± 0.9	13.0 ± 0.8	19.6 ± 0.9	19.5 ± 0.9	19.5 ± 0.9
Avg	30.0 ± 1.0	22.9 ± 1.0	23.1 ± 0.9	19.7 ± 1.0	29.1 ± 1.0	29.2 ± 1.0	28.8 ± 1.0
Client-Unseen Languages - Mid Resource							
Ta-En	8.0 ± 0.6	3.9 ± 0.4	5.0 ± 0.5	3.7 ± 0.4	9.2 ± 0.7	9.5 ± 0.7	8.2 ± 0.7
Uk-En	27.9 ± 1.0	18.2 ± 1.0	21.8 ± 0.9	20.7 ± 1.0	28.2 ± 1.0	28.0 ± 1.0	27.5 ± 1.0
Fi-En	25.7 ± 1.0	18.2 ± 1.0	18.8 ± 0.8	14.4 ± 1.0	25.0 ± 1.0	24.8 ± 1.0	24.4 ± 0.9
Avg	20.5 ± 0.9	13.4 ± 0.8	15.2 ± 0.7	12.9 ± 0.8	20.8 ± 0.9	20.8 ± 0.9	20.3 ± 0.9
Client-Unseen Languages - Low Resource							
Sw-En	26.0 ± 0.9	17.2 ± 1.0	18.4 ± 1.0	13.6 ± 1.0	24.4 ± 1.0	24.6 ± 1.0	23.5 ± 1.1
Si-En	15.9 ± 0.8	8.8 ± 0.7	9.6 ± 0.8	7.3 ± 0.7	15.1 ± 0.8	14.9 ± 0.8	14.2 ± 0.9
Ml-En	15.3 ± 0.9	8.0 ± 0.8	8.6 ± 0.8	6.3 ± 0.6	14.3 ± 0.8	14.9 ± 0.8	13.7 ± 0.9
Avg	19.1 ± 0.9	11.3 ± 0.8	12.2 ± 0.9	9.1 ± 0.8	17.9 ± 0.9	18.1 ± 0.9	17.1 ± 1.0
Weighted Metric Calculation							
RRS	0.000	0.496	0.323	0.643	0.896	0.882	0.882

Table 5: Different LoRA configurations varying the target modules and ranks. All models are trained for 1 epoch/round and for 100 rounds.