FedP^{2}EFT: Federated Learning to Personalize PEFT for Multilingual LLMs

Royson Lee[1], **Minyoung Kim**[1], **Fady Rezk**[2], **Rui Li**[1], **Stylianos I. Venieris**[1], **Timothy Hospedales**[1,2]

[1]Samsung AI Center, Cambridge, UK
[2]University of Edinburgh, UK
royson.lee@samsung.com

Abstract

Federated learning (FL) has enabled training of multilingual large language models (LLMs) on diverse and decentralized multilingual data, especially on low-resource languages. To improve client-specific performance, personalization via the use of parameter-efficient fine-tuning (PEFT) modules such as LoRA is common. This involves a *personalization strategy* (PS), such as the design of the PEFT adapter structures (*e.g.*, in which layers to add LoRAs and what ranks) and choice of hyperparameters (*e.g.*, learning rates) for fine-tuning. Instead of manual PS configuration, we propose FedP^{2}EFT, a federated *learning-to-personalize* method for multilingual LLMs in cross-device FL settings. Unlike most existing PEFT structure selection methods, which are prone to overfitting low-data regimes, FedP^{2}EFT collaboratively learns the optimal personalized PEFT structure for each client via Bayesian sparse rank selection. Evaluations on both simulated and real-world multilingual FL benchmarks demonstrate that FedP^{2}EFT largely outperforms existing personalized fine-tuning methods, while complementing other existing FL methods.

Code — https://github.com/SamsungLabs/fedp2eft

1 Introduction

Federated learning (FL) makes it possible to train multilingual large language models (LLMs) across different geographical regions, protecting linguistic diversity for low-resource languages (Zhao et al. 2024) while being compliant with privacy regulations (Lim et al. 2020), *e.g.*, General Data Protection Regulation (GDPR). Despite the impressive capabilities demonstrated by these models across various languages, their performance varies significantly depending on the language (Rust et al. 2021) and the data volume (Adelani et al. 2021) per client.

Moreover, the majority of existing FL-based multilingual LLM approaches have thus far focused on training a single global model (Iacob et al. 2025; Ye et al. 2024), limiting their performance on specific languages. Concretely, scaling a single model to different languages is challenged by issues such as the *Curse of Multilinguality* (Conneau et al. 2020), where adding more languages often leads to diminishing returns, and *Negative Interference* (Wang, Lipton, and Tsvetkov 2020), where diverse languages compete for limited model capacity. From a personalized FL (pFL) perspective, learning a global model often increases the initial performance at the expense of personalized performance, *e.g.*, fine-tuning from the global model (Jiang et al. 2019).

Naturally, pFL approaches can help bridge the gap and improve language personalization. However, existing techniques are either too costly to be applied to LLMs, *e.g.*, the use of meta-learning (Chen et al. 2018) and hypernetworks (Shamsian et al. 2021), or rely on suboptimal hand-crafted personalization strategies, *e.g.*, manual choice of personalized and language-specific layers (Iacob et al. 2025; Zhao et al. 2024; Wu et al. 2024), parameter-efficient fine-tuning (PEFT) adapter structures (Yang et al. 2024a,b), or clustering based on class labels and/or language commonality (Mansour et al. 2020; Ye et al. 2024).

Intuitively, optimizing personalization in pFL often necessitates dataset- and task-specific methods. The optimal level of personalization varies significantly depending on the characteristics of the data and the specific FL scenario (Chen et al. 2022; Ye et al. 2024). For instance, an English-pretrained LLM may require stronger personalization, *e.g.*, higher learning rates, when fine-tuning on German than on English. The optimal personalization strategy (PS) is thus contingent upon the specific task, the client, and the given *base model* (Lee et al. 2023).

In this paper, we address the issues above by proposing a novel federated hyperoptimization strategy that learns personalized PEFT configurations for each client. To this end, we propose FedP^{2}EFT, a method that enables clients to collaboratively learn language personalization strategies using FL. Specifically, we federatedly train a PS generator (PSG), as depicted in Fig. 1(a), which allows all participating clients to collaboratively learn the optimal mapping between local meta-data to optimal LoRA (Hu et al. 2022) ranks. Fig. 1(b) illustrates the personalization process of FedP^{2}EFT on a single client during inference, where per-layer LoRA ranks are generated depending on the initial *base model*, the client's dataset, and their resource budget. These personalized LoRA modules are then used to apply PEFT on the *base model* and yield the personalized model.

As FedP^{2}EFT focuses on improving the PEFT process per-dataset/task/client, it is directly pluggable to any starting *base model*, which may or may not be federatedly trained. This

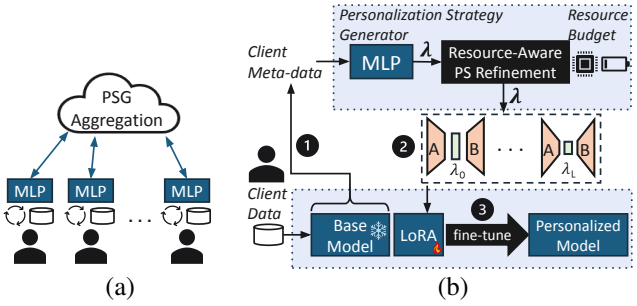

Figure 1: (a) We train our personalization strategy generator (PSG) using standard FL approaches. (b) FedP^{2}EFT's inference stage on a single client. ❶ Given the *base model* and the client's train dataset, features are extracted and passed into our PSG to generate a PS, $\boldsymbol{\lambda}$, for the client's budget. ❷ $\boldsymbol{\lambda}$ is then used to initialize all LoRA modules before ❸ the *base model* is personalized. The resulting personalized model is then used to evaluate on the client's test samples.

includes off-the-shelf pretrained models, FL approaches that learn a single global model (McMahan et al. 2017; Oh, Kim, and Yun 2022), and even pFL approaches that deploy personalized layers, *e.g.*, monolingual tokenizer and embeddings (Iacob et al. 2025), personalized LoRA adapters (Wu et al. 2024; Yang et al. 2024b), and language embeddings (Silva, Tambwekar, and Gombolay 2023). Through our experiments (Section 4), we show that our method *1)* largely outperforms both existing non-FL LoRA rank selection and FL-based learning-to-personalize techniques, and *2)* complements well with a range of existing FL approaches.

2 Related Work

Multilingual LLMs (MLLMs). Existing efforts in multilingual LLMs often underperform on low-resource languages due to *1)* data scarcity (Xu et al. 2024), *2)* the model's limited capacity to learn the intricacies of multiple languages (Conneau et al. 2020), and *3)* negative transfer learning among languages (Wang, Lipton, and Tsvetkov 2020). Common ways to counteract these challenges include the use of separate vocabulary and embeddings (Artetxe, Ruder, and Yogatama 2020), hand-crafted adapters (Pfeiffer et al. 2020), automatic data annotation (Dubey et al. 2024), clustering and merging languages with similar representations (Chung et al. 2020) , among other contributions (Wang et al. 2020; Conneau et al. 2019). Our work is orthogonal to these approaches and builds upon recent FL-based MLLMs (Iacob et al. 2025; Ye et al. 2024), which utilize FL to tap into previously inaccessible low-resource data sources.

Personalized Federated Learning (pFL). To obtain personalized client-specific models, various approaches have been proposed, including the use of personalized layers (Arivazhagan et al. 2019), meta-learning (Chen et al. 2018), model mixtures (Marfoq et al. 2021), hypernetworks (Shamsian et al. 2021), and transfer learning between global and local models (Shen et al. 2020). Some of these techniques have also been adopted for LLMs, *e.g.*, personalized LoRAs (Yang et al. 2024b), hypernetworks for client embeddings (Silva, Tambwekar, and Gombolay 2023), and mixtures of LoRA experts (Zhang et al. 2024b). Our work complements these approaches as personalized models can benefit from further fine-tuning as shown in Section 4.4.

Federated Hyperparameter Optimization (HPO). Most federated approaches to HPO do not utilize the client dataset for personalized hyperparameters. Instead, they employ a single set of hyperparameters across all clients based on the local validation loss evaluated before FL (Zhou et al. 2023) or sample from federatedly learnt hyperparameter categorical distributions for each client (Khodak et al. 2021). An exception to this is FedL2P (Lee et al. 2023) which utilizes a PSG for personalized per-layer learning rates and batch normalization hyperparameters. While FedL2P has been shown to work well in standard small-scale image and speech benchmarks, their applicability to LLMs is unclear: i) LLMs don't use batch normalization (BN), ii) LLMs often use adaptive optimizers, making learning rate a less sensitive hyparam for downstream performance. Learning the learning rates also require FedL2P to adopt expensive 2nd-order optimization methods as the learning rate is not a direct gradient of the loss, a limitation that is further aggravated with LLMs. We compare with FedL2P in our experiments.

PEFT Structure Learning. Contrary to the conventional approach of distributing uniform adapter modules across all layers, a recent line of work allows different LoRA ranks to be used across a model's weight matrices. Using fine-grained per-layer rank selection, existing methods include SVD-based LoRA reformulation followed by importance-based rank assignment (Zhang et al. 2023), trainable rank-gating units (Ding et al. 2023), selectively employing parallel weight modules (Song et al. 2024), meta-learning-based (Zhang et al. 2024a) and black-box optimization techniques (Tribes et al. 2024), specialized training recipes for multi-rank LoRA modules that allow flexible extraction of a range of ranks (Valipour et al. 2023), and coarse-grained dropping of LoRA-enhanced layers (Yao et al. 2024). While these methods can be effective in centralized setups, they typically require an excessive number of optimization steps, which is prone to overfitting in FL settings, where clients have limited amount of data.

3 Our Approach

3.1 Preliminaries & Motivation

In pFL, the goal is to minimize each client's local objective $\mathbb{E}_{(x,y)\sim P^i}\mathcal{L}^i(\Phi^i; x, y)$ where P^i represents the data distribution of the i-th client, x and y are the input data and labels, respectively, and $\mathcal{L}^i(\Phi^i; x, y)$ is the loss function for client i given model parameters Φ^i. This is typically achieved via fine-tuning (Chen et al. 2022) a *base model*, with parameters Φ^i_{BM} and a set of hyperparameters, *e.g.* learning rate. Note that Φ^i_{BM} may differ across clients if it is already personalized, *e.g.* if Φ^i_{BM} is obtained using a pFL algorithm.

Fine-tuning LLMs, however, is unprecedentedly compute and memory intensive, and prone to overfitting. As such, the majority of existing federated LLM works (Zhao et al. 2024; Sun et al. 2024) rely on PEFT methods, with LoRA (Hu

et al. 2022) being a prevalent choice due to its efficiency and performance. Specifically, for a frozen weight matrix $W \in \mathbb{R}^{d\times e}$, LoRA introduces low-rank matrices $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times e}$ where $r \ll \min(d, e)$. The adapted weights are then expressed as: $W' = W + \frac{\alpha_{\text{lora}}}{r} BA$ where α_{lora} is a hyperparameter and only B and A are trained during fine-tuning. Although effective, these FL works rely on a fixed hand-crafted PS, *e.g.*, a manually defined LoRA rank on hand-picked layers, for all clients, leading to suboptimal personalized models.

3.2 Personalized PEFT

We, instead, propose using a different PS for each client. Common hyperparameter choices from previous federated HPO approaches (Section 2) include learning rates and BN hyperparameters. While these hyperparameters have been shown to be effective for handling data heterogeneity in popular vision and speech benchmarks (Li et al. 2021, 2016; Arivazhagan et al. 2019), they are less consequential or not applicable when fine-tuning LLMs. This stems from the fact that LLMs are often fine-tuned using adaptive optimizers, *e.g.* Adam, which are more robust to the learning rate (Zhao et al. 2025), and BN layers are not typically used. A more critical hyperparameter choice shown to be effective, especially for cross-lingual transfer learning (Pfeiffer et al. 2020), is the PEFT adapter structure; specifically which layers to introduce LoRAs in and what ranks to utilize (Zhang et al. 2023, 2024a).

Adapting BayesTune for LoRA Rank Selection. We build upon BayesTune (Kim and Hospedales 2023), a Bayesian sparse model selection approach. Directly using BayesTune in our approach, however, will not only incur an intractable cost but also be difficult to optimize due to the high dimensionality of predicting per-parameter scaling coefficients (Appendix C). Hence, we adapt BayesTune by formulating PEFT personalization as a sparse LoRA rank selection problem and propose BayesTune-LoRA (BT-LoRA). Concretely, we introduce rank-wise latent variables $\lambda \in \mathbb{R}^r,\ \lambda_i > 0,\ \forall i = 1, 2, \cdots, r$ for each LoRA matrix: $B\lambda A$. Let $\boldsymbol{\lambda} = \{\lambda_{l,\cdot}\}_{l=1}^{L}$ be the set of all λ where $\lambda_{l,\cdot}$ represents the rank-wise scales for layer l in a model with L LoRA modules (similarly for $\boldsymbol{A}$ and $\boldsymbol{B}$). Using BayesTune, the values for $\theta = (\boldsymbol{\lambda}, \boldsymbol{A}, \boldsymbol{B})$ are optimized as:

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{\text{CE}}(\theta; D) + \frac{\alpha_s}{N}\mathcal{L}_s(\boldsymbol{\lambda}, \boldsymbol{B}) + \frac{\alpha_p}{N}\mathcal{L}_p(\boldsymbol{\lambda}) \quad (1)$$

where $D{=}\{(x_i, y_i)\}_{i=1}^{N}$ is the train dataset, N the size of D, $\mathcal{L}_{\text{CE}}(\theta; D)$ the cross-entropy loss, α_p and α_s hyperparameters, $\mathcal{L}_s$ the logarithm of the Laplace distribution (prior imposed on $p(B|\lambda)$[1]), $f(\|B_{l,i}\|_1; \mu, b) = \frac{1}{2b}\exp\left(-\frac{|\|B_{l,i}\|_1 - \mu|}{b}\right)$ with $\mu = 0$ (B is initialized to 0 in LoRA) and $b = \lambda_{l,i}$:

$$\mathcal{L}_s(\boldsymbol{\lambda}, \boldsymbol{B}) = \sum_{l}^{L}\sum_{i}^{r}\left(\log\lambda_{l,i} + \frac{\|B_{l,i}\|_1}{\lambda_{l,i}} + \log 2\right) \quad (2)$$

and $\mathcal{L}_p$ is the logarithm of the Gamma distribution (hyper-prior imposed on λ), $\mathcal{G}(\lambda_{l,i}; \alpha_g, \beta_g) = \frac{\beta_g^{\alpha_g}}{\Gamma(\alpha_g)}\lambda_{l,i}^{\alpha_g - 1}e^{-\beta_g\lambda_{l,i}}$ where $\alpha_g = 0.01, \beta_g = 100$ following the hyperparameters set by the original authors[2]:

$$\mathcal{L}_p(\boldsymbol{\lambda}) = \sum_{l}^{L}\sum_{i}^{r}(0.99\cdot\log\lambda_{l,i} + 100\cdot\lambda_{l,i} - 0.01\log(100) + \log\Gamma(0.01)) \quad (3)$$

In practice, we can save computations by removing all constants and the duplicate term $\log\lambda$, resulting in the following approximated penalty losses:

$$\mathcal{L}_s(\boldsymbol{\lambda}, \boldsymbol{B}) = \sum_{l}^{L}\sum_{i}^{r}\frac{\|B_{l,i}\|_1}{\lambda_{l,i}} \quad (4)$$

$$\mathcal{L}_p(\boldsymbol{\lambda}) = \sum_{l}^{L}\sum_{i}^{r}(\log\lambda_{l,i} + 100\cdot\lambda_{l,i}) \quad (5)$$

Intuition of $\mathcal{L}_p$ and $\mathcal{L}_s$. $\mathcal{L}_p$ encourages small λ while $\mathcal{L}_s$ encourages larger λ for larger LoRA B (per column) updates. Hence, minimizing the losses in Eq. (1) encourages larger λ in more significant ranks.

Personalizing PEFT with BT-LoRA. For each client, we attach BT-LoRA modules, θ, to all linear layers of its *base model* with rank $r_{\text{init}} = \alpha_{r\_mul}\cdot r_{\text{max target}}$ where $r_{\text{max target}}$ is the maximum inference resource budget and r_{init} is the initial rank before pruning. θ is then optimized using Adam (Kingma and Ba 2015) as per Eq. (1).[3]

After training, we freeze the resulting $\boldsymbol{\lambda}$ and use it for personalization. Specifically, given a resource budget (total rank budget) of $r\cdot L$, we prune $\boldsymbol{\lambda}$ by taking the top-$(r\cdot L)$ largest ranks, along with the corresponding rows of $\boldsymbol{A}$ and columns of $\boldsymbol{B}$.[4] We then reinitialize the pruned $\boldsymbol{A}$ and $\boldsymbol{B}$ and perform standard fine-tuning on $\mathcal{L}_{\text{CE}}$ with the frozen pruned $\boldsymbol{\lambda}$ to obtain the personalized model. Note that we only have to train $\boldsymbol{\lambda}$ once for all ranks $\leq r_{\text{max target}}$.

3.3 FedP²EFT: FL to Personalize PEFT

Overfitting often occurs when training an effective per-client PS in isolation due to limited client data. Following FedL2P (Lee et al. 2023), we mitigate this by federatedly learning a common PSG that generates client-wise PS. Concretely, we use a one hidden layer multilayer perceptron (MLP) with parameters ϕ that takes as input the client meta-data and outputs an estimated PS:

$$\hat{\boldsymbol{\lambda}} = \text{MLP}(\phi;\ E(h_0), SD(h_0), E(h_1), SD(h_1), \cdots, E(h_{L-1}), SD(h_{L-1})) \quad (6)$$

[1]Unlike BayesTune, where every parameter is associated with its own prior scale, we use an "independent" Laplace prior where each $\lambda_{l,i}$ applies to all entries of $B_{l,i}$

[2]α_g and β_g do not need to be tuned. Details in Appendix C.

[3]BayesTune proposed using SGLD (Welling and Teh 2011), adding Gaussian noise to the gradient updates and sampling from the posterior distribution. Due to the challenges of estimating the full posterior distribution in FL settings, particularly with limited client data, we opt to find a point estimate.

[4]The LoRA module is discarded for layers where $\|\lambda_l\|_1 = 0$

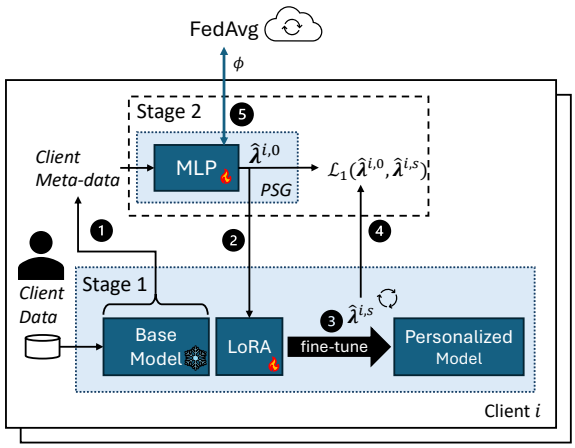

Figure 2: FedP^{2}EFT's federated training of PSG for each federated round. Details in Section. 3.3.

where h_{l-1} is the input feature to the l-th layer in the *base model*, and $E(\cdot)$ and $SD(\cdot)$ are the mean and standard deviation (SD), respectively.

In contrast to FedL2P, which adopts a computationally demanding meta-learning approach to train MLP, we take a two-stage strategy for each client: *1)* first, learn $\boldsymbol{\lambda}$, followed by *2)* regression learning of MLP to target the learned $\boldsymbol{\lambda}$.

Federated Training of FedP^{2}EFT. Fig. 2 shows the entire FedP^{2}EFT algorithm during federated training. For each federated round, each sampled participating client i receives ϕ from the server and loads them into its MLP. They then ❶ perform a forward pass of the local train dataset on their *base model* and a forward pass of the MLP with the resulting features as per Eq. (6). ❷ The estimated $\hat{\boldsymbol{\lambda}}^i$ is plugged into our proposed BT-LoRA (Section 3.2) and ❸ fine-tuning is performed as per Eq. (1) for s steps (Stage 1). ❹ The resulting $\hat{\boldsymbol{\lambda}}^{i,s}$ is used as an approximated ground-truth for regression learning of MLP to target the learned $\hat{\boldsymbol{\lambda}}^{i,s}$, where $\mathcal{L}_1$ is the L1 loss (Stage 2). Finally, ❺ ϕ is sent back to the server for aggregation. As there is no single aggregation method that outperforms all others in every situation (Matsuda et al. 2022; Chen et al. 2022; Ye et al. 2024), we utilize FedAvg (McMahan et al. 2017). The aggregated ϕ is then sent to clients for the next round. To complement Figure 2, we provide Algorithm 1 in Appendix D.

The learned ϕ is deployable to any client (*seen* or *unseen*) after federated training. Note that unlike FedL2P, which requires federated training for every target rank, FedP^{2}EFT inherits the property of BT-LoRA; federated training is a one-time cost for all ranks $\leq r_{\text{max target}}$.

Inference with FedP^{2}EFT. Fig. 1(b) shows how FedP^{2}EFT personalizes PEFT for each client upon deployment. Given the learned MLP and the client's *base model*, ❶ the client meta-data are retrieved (Eq. 6) and used to generate the client's PS, $\boldsymbol{\lambda}$. ❷ Given the client's resource budget of total rank $r \cdot L$, we take the top-$(r \cdot L)$ largest ranks in $\boldsymbol{\lambda}$, freeze them, and initialize our proposed BT-LoRA modules for all layers where $\|\lambda_l\|_1 > 0$. ❸ The personalized LoRA ranks are used for fine-tuning before merging back to the *base model* to obtain the final personalized model. To complement Figure 1(b), we provide Algorithm 2 in Appendix D.

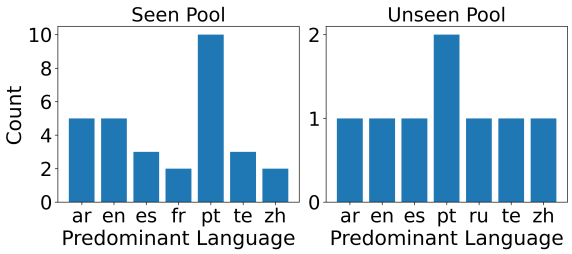

Figure 3: No. of clients by predominant language in our FedAya setup

4 Evaluation

We conduct experiments on multilingual scenarios, where clients with diverse high- and low-resource languages can collaboratively learn how to personalize a given base model to better cater to their language preferences. In all experiments, we divide clients in two pools, *seen* and *unseen*, where only the clients in the *seen* pool actively participate in federated training. We set the maximum number of communication rounds for training the PSG to 150, randomly sampling 10% of participating clients every round. We use Adam as the default optimizer for all our experiments. Unless mentioned otherwise, we show results for resource budget $r = 2$ where the total rank budget is $r \cdot L$ and leave results for other resource budgets $r = 4, 8, 16$ in the Appendix. In the following subsections, we summarize the FL scenarios that we consider in our experiments, leaving comprehensive details in Appendix A.

4.1 Setup: Tasks, Models, and Datasets

Text Classification. We adopt the pretrained multilingual BERT (Devlin et al. 2018) (mBERT) for all text classification experiments. For datasets, we introduce additional data heterogeneity to the simulated FL setups, XNLI (Conneau et al. 2018) and MasakhaNEWS (Adelani et al. 2023), proposed in PE_FL (Zhao et al. 2024).

For our XNLI setup, we sample 2k instances for train and 500 for test in each pool. In contrast to PE_FL, which had 15 clients (1 client per language), we divide the data equally among 20 clients per language. We then adopt the latent Dirichlet allocation (LDA) partition method (Hsu, Qi, and Brown 2019), $y \sim Dir(\alpha)$, to simulate non-IID label shifts among these clients, with $\alpha = 0.5$. Hence, there is a total of 600 clients (15 languages $\cdot$ 20 clients $\cdot$ 2 pools), consisting of both label and feature heterogeneity.

For MasakhaNEWS, we first split the data in each of the 16 languages by half for each pool. Similar to our XNLI setup, we divide each language's data equally among 10 clients and adopt LDA with $\alpha = 0.5$, resulting in 320 clients in total. Differing from our XNLI setup, each language varies in the amount of samples, adding another layer of data heterogeneity to the setup: quantity skew.

Lan	LoRA	AdaLoRA	BT-LoRA	FedL2P	FedP^{2}EFT
eng	90.4±0.1	89.9±0.0	89.9±0.0	90.7±0.6	**92.0±0.0**
som	60.1±0.3	59.4±0.3	59.6±0.3	61.0±1.2	**65.5±1.2**
run	81.4±0.5	81.2±0.3	81.4±0.0	82.0±0.9	**87.8±0.3**
fra	88.6±0.0	88.6±0.0	88.6±0.0	89.1±0.7	**93.5±0.2**
lin	83.5±0.5	82.8±0.0	83.1±0.5	83.9±0.0	**88.5±0.9**
ibo	79.8±0.2	79.0±0.0	79.0±0.0	79.7±0.2	**82.6±0.0**
amh	45.7±0.0	45.2±0.0	45.2±0.0	45.7±0.0	**52.0±0.2**
hau	75.8±0.0	75.0±0.1	75.2±0.0	76.7±1.1	**79.7±0.3**
pcm	96.0±0.0	96.0±0.0	96.0±0.0	96.0±0.0	**97.6±0.3**
swa	79.0±0.0	78.6±0.0	78.6±0.0	79.7±0.4	**84.7±0.5**
orm	64.2±0.0	64.0±0.3	64.0±0.3	64.4±0.3	**72.2±1.0**
xho	69.1±0.3	69.6±0.0	69.6±0.0	69.1±0.3	**76.3±0.6**
yor	79.2±0.2	78.9±0.2	79.0±0.0	79.7±0.6	**82.1±0.2**
sna	78.8±0.0	78.8±0.0	78.8±0.0	78.8±0.0	**81.3±0.7**
lug	67.6±0.0	67.6±0.0	67.6±0.0	67.9±0.4	**69.1±0.4**
tir	44.9±0.0	44.9±0.0	44.9±0.0	45.3±0.7	**63.5±0.3**

Table 1: Mean±SD Accuracy of each language across 3 different seeds for *seen* clients of our MasakhaNEWS setup ($r = 2$, see Appendix Table 7 for other r values). The pretrained model is trained using **Standard FL with full fine-tuning** and the resulting *base model* is personalized to each client given a baseline approach.

Lan	LoRA	AdaLoRA	BT-LoRA	FedL2P	FedP^{2}EFT
eng	**90.7±0.0**	90.3±0.0	90.3±0.0	90.6±0.1	**90.7±0.1**
som	68.5±0.3	67.3±0.0	67.3±0.0	68.5±0.3	**72.1±0.6**
run	82.0±0.5	80.8±0.0	80.8±0.0	82.6±0.0	**88.4±0.3**
fra	84.4±0.0	84.4±0.0	84.4±0.0	84.4±0.0	**88.2±0.4**
lin	79.5±0.0	79.5±0.0	79.5±0.0	79.5±0.0	**86.4±0.0**
ibo	76.8±0.2	76.9±0.0	76.9±0.0	76.4±0.4	**81.5±0.4**
amh	46.3±0.0	46.3±0.0	45.9±0.2	46.8±0.8	**51.1±0.0**
hau	75.5±0.1	74.8±0.1	74.6±0.0	76.0±0.6	**79.5±0.3**
pcm	90.2±0.0	90.2±0.0	90.2±0.0	90.2±0.0	**93.7±0.3**
swa	75.6±0.3	75.9±0.2	75.5±0.2	75.9±0.4	**78.0±0.4**
orm	62.0±0.0	61.5±0.3	61.5±0.3	62.2±0.3	**73.0±0.5**
xho	64.2±0.6	64.2±0.3	63.8±0.0	64.4±0.9	**78.5±1.1**
yor	80.1±0.0	79.6±0.0	79.5±0.2	80.3±0.2	**83.7±0.2**
sna	74.6±0.0	74.6±0.0	74.4±0.2	74.8±0.3	**80.0±0.4**
lug	65.2±0.0	65.2±0.0	65.2±0.0	66.1±0.0	**69.6±0.0**
tir	41.9±0.0	41.9±0.0	42.6±0.0	41.9±0.0	**58.3±0.3**

Table 2: Mean±SD Accuracy of each language for *unseen* clients of our MasakhaNEWS setup ($r = 2$, see Appendix Table 8 for other r values). The pretrained model is trained using **Standard FL with full fine-tuning** and the *base model* is personalized to each client given a baseline approach.

Instruction-Tuning Generation. We use pretrained MobileLLaMA-1.4B (Chu et al. 2023) and Llama-3.2-3B (Dubey et al. 2024), which are representative of commonly supported model sizes on high-end edge devices (Tan et al. 2024)[5], and run experiments with them on the Fed-Aya dataset (Ye et al. 2024). Fed-Aya, a real-world FL dataset naturally partitioned by annotator ID, comprises 38 clients, 8 of whom we designate as our *unseen* pool. We split each client's data $80\%/20\%$ for training and testing. Each client has up to 4 languages and Fig. 3 shows the predominant language (client's most frequent language) distribution in our setup.

4.2 Setup: Complementary Approaches

FedP^{2}EFT is compatible with off-the-shelf models and models trained with existing FL approaches. Concretely, given a pretrained model, we obtain a *base model* using one of the following approaches:

Standard FL. We further train the pretrained model federatedly on the *seen* pool, either using existing PEFT methods or full fine-tuning (Ye et al. 2024; Sun et al. 2024),

Personalized FL (pFL). We adopt two recent pFL works: *i)* FedDPA-T (Yang et al. 2024b), which learns per-client personalized LoRA modules in addition to global LoRA modules, and *ii)* DEPT (SPEC) (Iacob et al. 2025), which learns per-client personalized token and positional embeddings while keeping the rest of the model shared. The *base model* hence differs for each client.

Off-the-shelf. We use the pretrained model as the *base model* without additional training.

[5]Refer to Appendix Sec E for experiments with Llama-3.1-8B.

4.3 Setup: Baselines

Given a *base model*, we compare FedP^{2}EFT with existing fine-tuning and *learning to personalize* approaches. For each baseline, we either follow best practices recommended by the corresponding authors or employ a simple grid search and pick the best performing hyperparameters (Appendix A).

LoRA PEFT. We deploy LoRA (Hu et al. 2022) on all linear layers of the model with a fixed rank r.

Non-FL Rank Selection. We compare with AdaLoRA (Zhang et al. 2023) and our proposed LoRA-variant of BayesTune (Kim and Hospedales 2023), BT-LoRA (Section 3.2), which optimizes $\boldsymbol{\lambda}$ separately for each client.

FL to Personalize. We compare with FedL2P (Lee et al. 2023) which trains a MLP federatedly to output per-client learning rates for each LoRA module.

4.4 Results on Text Classification

We evaluate our approach in a typical FL setup, where the pretrained model is first trained using Standard FL with full fine-tuning and the resulting *base model* is then personalized to each client. Tables 1 & 2 show the mean and standard deviation (SD) of the accuracy for each language in our MasakhaNEWS setup for *seen* and *unseen* pool respectively (similarly for XNLI in Appendix Tables 15 & 16).

The results in all four tables show that federated *learning to personalize* methods (FedL2P and FedP^{2}EFT) outperform the other baselines in most cases. Non-FL rank selection approaches (AdaLoRA and BT-LoRA), on the other hand, tend to overfit and/or struggle to learn an optimal rank structure given the limited number of samples in each client. Comparing FedL2P and FedP^{2}EFT, FedP^{2}EFT largely surpass FedL2P with a few exceptions, indicating that learning to personalize LoRA rank structure is the better hyperparameter choice than personalizing learning rates; this finding is also

Lan	LoRA	AdaLoRA	BT-LoRA	FedL2P	FedP²EFT
bg	45.8±0.3	44.1±0.1	43.8±0.0	47.5±2.8	**64.4±0.2**
hi	42.8±0.2	41.2±0.0	40.6±0.0	44.5±3.2	**57.8±1.3**
es	48.7±0.2	47.5±0.1	47.2±0.0	50.3±2.9	**58.5±1.3**
el	50.9±0.1	50.0±0.0	50.0±0.0	51.5±1.4	**59.7±2.6**
vi	53.0±0.0	52.4±0.0	52.4±0.0	53.6±0.7	**60.8±0.7**
tr	48.0±0.3	46.5±0.1	46.2±0.0	50.1±2.8	**58.9±2.9**
de	49.9±0.1	48.0±0.0	47.4±0.2	50.9±2.6	**55.0±0.2**
ur	41.9±0.1	38.8±0.0	38.8±0.0	44.8±4.5	**63.7±0.2**
en	46.5±0.3	44.7±0.1	44.3±0.1	49.1±4.3	**55.9±0.2**
zh	44.4±0.2	42.2±0.0	41.6±0.0	46.5±4.0	**56.3±0.3**
th	42.5±0.2	40.7±0.1	40.5±0.1	44.4±3.7	**58.3±0.2**
sw	51.8±0.2	50.4±0.0	49.8±0.0	52.3±1.3	**59.5±0.5**
ar	46.9±0.2	45.0±0.0	44.7±0.1	48.4±3.0	**55.2±0.2**
fr	48.1±0.2	46.0±0.0	46.0±0.0	49.5±3.0	**57.5±0.2**
ru	50.5±0.2	48.8±0.0	48.1±0.2	51.6±2.3	**55.2±0.4**

Table 3: Mean±SD Accuracy of each language across 3 seeds for clients in the *seen* pool of our XNLI setup ($r = 2$, see Appendix Table 9 for other r values). The pretrained model is trained using **FedDPA-T** and the resulting *base model* is personalized to each client given a baseline approach. See Appendix Table 14 for results with **DEPT (SPEC)**.

aligned with recent LLM-based optimizer findings (Zhao et al. 2025), which shows that Adam's performance is robust with respect to its learning rate.

FedP²EFT's pFL Compatibility. Apart from Standard FL, we show that FedP²EFT can be plugged into existing pFL works that trains both a subset of the pretrained model and personalized layers for each client. Table 3 and Appendix Table 14 show that FedP²EFT outperforms baselines in almost all cases in our XNLI setup given a *base model* trained using FedDPA-T (Yang et al. 2024b) and DEPT(SPEC) (Iacob et al. 2025) respectively. In short, FedP²EFT can be integrated into a larger family of existing pFL approaches, listed in Section 2, to further improve personalization performance.

4.5 Results on Instruction-Tuning Generation

We evaluate our approach on the more challenging real-world multilingual benchmark, Fed-Aya. Tables 4 and 5 show the average METEOR (Banerjee and Lavie 2005)/ROUGE-L (Lin 2004)[6] of each language given the off-the-shelf instruction finetuned Llama-3.2-3B (Llama-3.2-3B-Instruct) for *seen* and *unseen* clients respectively. Similarly, in Appendix Tables 17 and 18, we show the same tables given a pretrained MobileLLaMA-1.4B model trained using Standard FL with LoRA following the training recipe from FedLLM-Bench (Ye et al. 2024). These two models represent scenarios where the *base model* may or may not be trained using FL.

In all four tables, FedP²EFT outperforms baselines in most scenarios. We also observe that FedL2P mostly underperforms standard baselines, a phenomenon also observed for our XNLI setup when the *base model* is trained with FedDPA-T (Appendix Table 9). We hypothesize that the inner-loop

[6]Due to limited space, we include ROUGE-1 in the Appendix.

Lan	LoRA	AdaLoRA	BT-LoRA	FedL2P	FedP²EFT
te	**0.24**/0.13	**0.24/0.14**	**0.24/0.14**	0.23/0.13	**0.24/0.14**
ar	0.34/0.07	0.34/0.07	0.32/0.06	0.33/0.07	**0.37/0.08**
es	**0.39**/0.39	**0.39/0.40**	0.38/0.38	0.38/0.38	**0.39**/0.39
en	0.33/0.31	0.35/0.33	0.29/0.23	0.33/0.30	**0.37/0.36**
fr	0.29/0.30	0.29/0.29	0.29/0.29	0.29/0.30	**0.34/0.33**
zh	**0.11/0.12**	**0.11/0.12**	0.09/0.11	0.09/**0.12**	**0.11/0.12**
pt	0.38/0.41	0.38/0.41	0.37/0.39	0.38/0.40	**0.40/0.42**

Table 4: Avg. METEOR/ROUGE-L for *seen* clients in our Fed-Aya setup. ($r = 2$, see Appendix Table 10 for other r values) *Base model* is off-the-shelf Llama-3.2-3B-Instruct.

Lan	LoRA	AdaLoRA	BT-LoRA	FedL2P	FedP²EFT
te	0.16/0.08	0.16/**0.11**	**0.17**/0.09	0.16/**0.11**	**0.17**/0.07
ar	0.24/0.04	0.23/**0.07**	0.22/0.05	0.24/0.05	**0.26**/0.05
es	0.43/0.44	0.43/0.44	0.38/0.41	0.41/0.43	**0.44/0.48**
en	**0.32**/0.25	**0.32**/0.25	**0.32**/0.24	0.31/0.24	0.31/**0.27**
fr	**0.55/0.67**	**0.55/0.67**	0.40/0.33	0.40/0.33	0.22/0.22
zh	0.25/0.00	0.25/0.00	0.23/0.00	0.23/0.00	**0.28/0.01**
pt	**0.36**/0.40	0.33/0.40	0.31/0.36	0.34/0.39	0.35/**0.41**
ru	0.22/0.17	0.23/0.17	0.25/0.18	0.25/0.20	**0.34/0.27**

Table 5: Avg. METEOR/ROUGE-L for *unseen* clients in our Fed-Aya setup ($r = 2$, see Appendix Table 11 for other r values). *Base model* is off-the-shelf Llama-3.2-3B-Instruct.

optimization in FedL2P fail to reach a stationary point[7] due to the inherent task difficulty (Fed-Aya) or a less-performant *base model*, resulting in a sub-optimal hypergradient and downstream performance.

FedP²EFT Limitations. In some cases, FedP²EFT falls short, especially in the recall performance (ROUGE), such as Russian (*ru*) and French (*fr*) for *unseen* clients for both *base models* in most scenarios. These cases highlight two limitations of our approach:

i) Challenge with zero-shot transfer. *ru* is not seen by PSG during federated training; there are no *ru* samples, nor other languages similar to *ru*, in the *seen* pool, resulting in worse *ru* performance. Hence, we do not expect a similar outcome in datasets with a more diverse pool of clients.

ii) Predominant language dependence. None of the clients in the *unseen* pool have *fr* as a predominant language (Fig. 3). Because the number of predominant language samples heavily influences $\boldsymbol{\lambda}$, the performance for *fr* is worse. This predominant language bias is quantified and future potential directions are discussed in Appendix Section E.

4.6 Cost of FedP²EFT

Table 6 shows the mean latency and the peak memory usage across 100 runs on the first client in the *seen* pool for $r = 16$ using an Nvidia A100 GPU. Non-FL baselines do not incur a federated training cost while FL approaches requires training the PSG. Comparing FedL2P and FedP²EFT, FedP²EFT does

[7]FedL2P relies on the implicit function theorem for hypergradient computation.

Federated Training			
Dataset (Model)	**Approach**	**Mean Latency (s)**	**Peak Memory (GB)**
XNLI (mBERT)	FedL2P	28.2	3.5
	FedP²EFT	3.4	3.1
Fed-Aya (Llama-3.2-3B)	FedL2P	226.5	32.2
	FedP²EFT	80.3	25.2

Inference			
Dataset (Model)	**Approach**	**Mean Latency (s)**	**Peak Memory (GB)**
XNLI (mBERT)	LoRA	3.1	2.6
	AdaLoRA	3.4	2.7
	BT-LoRA	2.4[1]+ 3.1	3.1[1]+ 2.7
	FedL2P	4.5	3.0
	FedP²EFT	4.4	2.7
Fed-Aya (Llama-3.2-3B)	LoRA	347.3	17.9
	AdaLoRA	423.9	18.8
	BT-LoRA	72.3[1]+ 357.6	25.2[1]+ 18.3
	FedL2P	380.0	19.8
	FedP²EFT	400.6	15.7

[1] One time cost per client for all targeted ranks

Table 6: Mean latency and memory costs across 100 runs of the first client in the *seen* pool using an NVIDIA A100 GPU.

not require second-order optimization, resulting in better efficiency. Note that FedL2P needs to be run for every r while FedP²EFT runs once for all targeted ranks.

For communication costs, not shown in the table, FedP²EFT is more costly as it predicts per LoRA rank while FedL2P predicts per layer. Nonetheless, these costs are negligible compared to running FL on the *base model*; FedL2P uses 0.02% and 0.002% and FedP²EFT uses 0.2% and 0.16% of the parameters of mBERT and Llama-3.2-3B respectively.

During inference, FL-based approaches incur an additional forward pass of *base model* and the PSG compared to non-FL approaches. Memory-wise, FedP²EFT results in the smallest memory footprint for autoregressive generation as the PSG learns not to attach LoRA modules $\lambda_l = 0$ on some layers, skipping *matmul* operations entirely. More details discussions on cost can be found in Appendix B.

4.7 Further Analysis

We further analyze $\boldsymbol{\lambda}$ and how they differ across languages. Surprisingly, we find that FedP²EFT learns language-agnostic rank structures. In other words, depending on the task and the *base model*, the rank structure of $\boldsymbol{\lambda}$ is fixed across languages. For instance, in the case where $r = 2$, FedP²EFT allocate ranks to dense layers instead of attention blocks. With more budget, *e.g.*, $r = 16$, FedP²EFT allocates more rank to either the query attention layer or the value attention layer depending on the setup. We show these rank structures across all setups for $r = 2$ and $r = 16$ in Appendix Fig. 5-14.

While the rank structure is the same across languages, the rank-wise scales (absolute values of $\boldsymbol{\lambda}$) differ. Following FedL2P, we visualize the difference in $\boldsymbol{\lambda}$ for different languages using the normalized mean distance, $d(j, k)$, between all clients pairs holding data for languages j and k. Fig. 4 and Appendix Fig. 15 show these distances for XNLI and MasakhaNEWS setup respectively. Specifically, the value of each block in each figure is computed as follows: $\log(\frac{d(j,k)}{\sqrt{d(j,j)}\sqrt{d(k,k)}})$. Hence, the smaller the distance, the more similar $\boldsymbol{\lambda}$ is between languages. The results are aligned with our intuition that similar languages have similar $\boldsymbol{\lambda}$. For instance, the closest language to Urdu (*ur*) is Arabic (*ar*), both of which have the closest $\boldsymbol{\lambda}$ similarity (Fig. 4); likewise, for Tigrinya (*tir*) and Amharic (*amh*) in Appendix Fig. 15. We also observe that unrelated languages have similar $\boldsymbol{\lambda}$, *e.g.*, Mandarin (*zh*) and Vietnamese (*vi*) share similar $\boldsymbol{\lambda}$ with the Indo-European languages (Fig. 4). This finding adds to existing evidence that leveraging dissimilar languages can sometimes benefit particular languages (Ye et al. 2024).

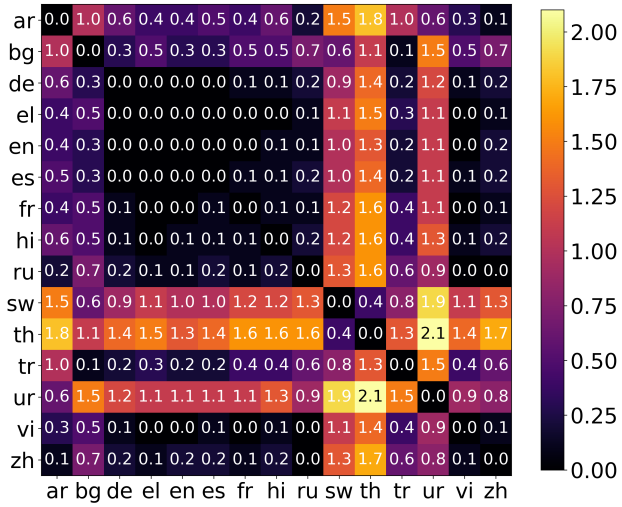

Figure 4: Cross-lingual $\boldsymbol{\lambda}$ distance in our XNLI setup. Each block shows the log-scale normalized average Euclidean distances between all pairs of clients' $\boldsymbol{\lambda}$ in their respective languages. The smaller the distance, the more similar $\boldsymbol{\lambda}$ is.

5 Conclusion

In this work, we tackle language personalization through FedP²EFT, a federated learning method that learns how to perform PEFT on heterogeneous data. We show that our proposed federated *learning-to-personalize* approach is easily pluggable to off-the-shelf LLMs and standard and personalized FL methods alike, surpassing other personalized fine-tuning baselines in most cases. Our results show that FedP²EFT automatically learns model- and task-specific language-agnostic LoRA rank structures as well as effective cross-lingual transfers, where both diverse low- and high-resource languages can share similar LoRA rank magnitudes. Despite clear advantages, our approach falls short in personalizing for each client's minority languages, as the personalized solution is skewed towards their predominant language. Nonetheless, our work is a significant step towards successfully merging the benefits of multilingual learning and personalized FL.

References

Adelani, D. I.; Abbott, J.; Neubig, G.; D'souza, D.; Kreutzer, J.; Lignos, C.; Palen-Michel, C.; Buzaaba, H.; Rijhwani, S.; Ruder, S.; et al. 2021. MasakhaNER: Named entity recognition for African languages. *Transactions of the Association for Computational Linguistics*.

Adelani, D. I.; Masiak, M.; Azime, I. A.; Alabi, J. O.; Tonja, A. L.; Mwase, C.; Ogundepo, O.; Dossou, B. F.; Oladipo, A.; Nixdorf, D.; et al. 2023. MasakhaNEWS: News Topic Classification for African languages. In *4th Workshop on African Natural Language Processing*.

Arivazhagan, M. G.; Aggarwal, V.; Singh, A. K.; and Choudhary, S. 2019. Federated Learning with Personalization Layers. *arXiv preprint arXiv:1912.00818*.

Artetxe, M.; Ruder, S.; and Yogatama, D. 2020. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Banerjee, S.; and Lavie, A. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Chen, D.; Gao, D.; Kuang, W.; Li, Y.; and Ding, B. 2022. pFL-Bench: A Comprehensive Benchmark for Personalized Federated Learning. *Advances in Neural Information Processing Systems*.

Chen, F.; Luo, M.; Dong, Z.; Li, Z.; and He, X. 2018. Federated Meta-Learning with Fast Convergence and Efficient Communication. *arXiv preprint arXiv:1802.07876*.

Chu, X.; Qiao, L.; Lin, X.; Xu, S.; Yang, Y.; Hu, Y.; Wei, F.; Zhang, X.; Zhang, B.; Wei, X.; et al. 2023. MobileVLM : A Fast, Strong and Open Vision Language Assistant for Mobile Devices. *arXiv preprint arXiv:2312.16886*.

Chung, H. W.; Garrette, D.; Tan, K. C.; and Riesa, J. 2020. Improving Multilingual Models with Language-Clustered Vocabularies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Unsupervised Cross-lingual Representation Learning at Scale. In *Annual Meeting of the Association for Computational Linguistics*.

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Conneau, A.; Rinott, R.; Lample, G.; Williams, A.; Bowman, S. R.; Schwenk, H.; and Stoyanov, V. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Ding, N.; Lv, X.; Wang, Q.; Chen, Y.; Zhou, B.; Liu, Z.; and Sun, M. 2023. Sparse Low-rank Adaptation of Pre-trained Language Models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Hsu, T.-M. H.; Qi, H.; and Brown, M. 2019. Measuring the Effects of Non-identical Data Distribution for Federated Visual Classification. *arXiv preprint arXiv:1909.06335*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. *International Conference on Learning Representations*.

Iacob, A.; Sani, L.; Kurmanji, M.; Shen, W. F.; Qiu, X.; Cai, D.; Gao, Y.; and Lane, N. D. 2025. DEPT: Decoupled Embeddings for Pre-training Language Models. In *International Conference on Learning Representations*.

Jiang, Y.; Konečnỳ, J.; Rush, K.; and Kannan, S. 2019. Improving Federated Learning Personalization via Model Agnostic Meta Learning. *arXiv preprint arXiv:1909.12488*.

Khodak, M.; Tu, R.; Li, T.; Li, L.; Balcan, M.-F. F.; Smith, V.; and Talwalkar, A. 2021. Federated Hyperparameter Tuning: Challenges, Baselines, and Connections to Weight-Sharing. *Advances in Neural Information Processing Systems*.

Kim, M.; and Hospedales, T. 2023. BayesTune: Bayesian Sparse Deep Model Fine-tuning. *Advances in Neural Information Processing Systems*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Lee, R.; Kim, M.; Li, D.; Qiu, X.; Hospedales, T.; Huszár, F.; and Lane, N. 2023. FedL2P: Federated Learning to Personalize. *Advances in Neural Information Processing Systems*.

Li, X.; Jiang, M.; Zhang, X.; Kamp, M.; and Dou, Q. 2021. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. In *International Conference on Learning Representations*.

Li, Y.; Wang, N.; Shi, J.; Liu, J.; and Hou, X. 2016. Revisiting Batch Normalization For Practical Domain Adaptation. *arXiv preprint arXiv:1603.04779*.

Lim, W. Y. B.; Luong, N. C.; Hoang, D. T.; Jiao, Y.; Liang, Y.-C.; Yang, Q.; Niyato, D.; and Miao, C. 2020. Federated Learning in Mobile Edge Networks: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*.

Mansour, Y.; Mohri, M.; Ro, J.; and Suresh, A. T. 2020. Three Approaches for Personalization with Applications to Federated Learning. *arXiv preprint arXiv:2002.10619*.

Marfoq, O.; Neglia, G.; Bellet, A.; Kameni, L.; and Vidal, R. 2021. Federated Multi-Task Learning Under a Mixture of Distributions. *Advances in Neural Information Processing Systems*.

Matsuda, K.; Sasaki, Y.; Xiao, C.; and Onizuka, M. 2022. An Empirical Study of Personalized Federated Learning. *arXiv preprint arXiv:2206.13190*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*.

Oh, J.; Kim, S.; and Yun, S.-Y. 2022. FedBABU: Towards Enhanced Representation for Federated Image Classification. In *International Conference on Learning Representations*.

Pfeiffer, J.; Vulić, I.; Gurevych, I.; and Ruder, S. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Rust, P.; Pfeiffer, J.; Vulić, I.; Ruder, S.; and Gurevych, I. 2021. How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Shamsian, A.; Navon, A.; Fetaya, E.; and Chechik, G. 2021. Personalized Federated Learning Using Hypernetworks. In *International Conference on Machine Learning*.

Shen, T.; Zhang, J.; Jia, X.; Zhang, F.; Huang, G.; Zhou, P.; Kuang, K.; Wu, F.; and Wu, C. 2020. Federated Mutual Learning. *arXiv preprint arXiv:2006.16765*.

Silva, A.; Tambwekar, P.; and Gombolay, M. 2023. FedPerC: Federated Learning for Language Generation with Personal and Context Preference Embeddings. In *Findings of the Association for Computational Linguistics*.

Song, H.; Zhao, H.; Majumder, S.; and Lin, T. 2024. Increasing Model Capacity for Free: A Simple Strategy for Parameter Efficient Fine-tuning. In *The Twelfth International Conference on Learning Representations*.

Sun, G.; Khalid, U.; Mendieta, M.; Wang, P.; and Chen, C. 2024. Exploring Parameter-Efficient Fine-Tuning to Enable Foundation Models in Federated Learning. In *IEEE International Conference on Big Data*.

Tan, F.; Lee, R.; Łukasz Dudziak; Hu, S. X.; Bhattacharya, S.; Hospedales, T.; Tzimiropoulos, G.; and Martinezs, B. 2024. MobileQuant: Mobile-friendly Quantization for On-device Language Models. In *Conference on Empirical Methods in Natural Language Processing*.

Tribes, C.; Benarroch-Lelong, S.; Lu, P.; and Kobyzev, I. 2024. Hyperparameter Optimization for Large Language Model Instruction-Tuning. In *Edge Intelligence Workshop AAAI*.

Valipour, M.; Rezagholizadeh, M.; Kobyzev, I.; and Ghodsi, A. 2023. DyLoRA: Parameter-Efficient Tuning of Pre-trained Models using Dynamic Search-Free Low-Rank Adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Wang, Z.; Karthikeyan, K.; Mayhew, S.; and Roth, D. 2020. Extending Multilingual BERT to Low-Resource Languages. In *Findings of the Association for Computational Linguistics: EMNLP*.

Wang, Z.; Lipton, Z. C.; and Tsvetkov, Y. 2020. On Negative Interference in Multilingual Models: Findings and A Meta-Learning Treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Welling, M.; and Teh, Y. W. 2011. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *Proceedings of the 28th International Conference on Machine Learning*.

Wu, X.; Liu, X.; Niu, J.; Wang, H.; Tang, S.; and Zhu, G. 2024. FedLoRA: When Personalized Federated Learning Meets Low-Rank Adaptation.

Xu, Y.; Hu, L.; Zhao, J.; Qiu, Z.; Ye, Y.; and Gu, H. 2024. A Survey on Multilingual Large Language Models: Corpora, Alignment, and Bias. *arXiv preprint arXiv:2404.00929*.

Yang, Y.; Liu, X.; Gao, T.; Xu, X.; and Wang, G. 2024a. SA-FedLora: Adaptive Parameter Allocation for Efficient Federated Learning with LoRA Tuning. *arXiv preprint arXiv:2405.09394*.

Yang, Y.; Long, G.; Shen, T.; Jiang, J.; and Blumenstein, M. 2024b. Dual-Personalizing Adapter for Federated Foundation Models. In *Advances in Neural Information Processing Systems*.

Yao, K.; Gao, P.; Li, L.; Zhao, Y.; Wang, X.; Wang, W.; and Zhu, J. 2024. Layer-wise Importance Matters: Less Memory for Better Performance in Parameter-efficient Fine-tuning of Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

Ye, R.; Ge, R.; Zhu, X.; Chai, J.; Du, Y.; Liu, Y.; Wang, Y.; and Chen, S. 2024. FedLLM-Bench: Realistic Benchmarks for Federated Learning of Large Language Models. In *Advances in Neural Information Processing Systems*.

Zhang, Q.; Chen, M.; Bukharin, A.; He, P.; Cheng, Y.; Chen, W.; and Zhao, T. 2023. AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. In *The Eleventh International Conference on Learning Representations*.

Zhang, R.; Qiang, R.; Somayajula, S. A.; and Xie, P. 2024a. AutoLoRA: Automatically Tuning Matrix Ranks in Low-Rank Adaptation Based on Meta Learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Zhang, Y.; Qin, Z.; Wu, Z.; and Deng, S. 2024b. Personalized Federated Fine-Tuning for LLMs via Data-Driven Heterogeneous Model Architectures. *arXiv preprint arXiv:2411.19128*.

Zhao, R.; Morwani, D.; Brandfonbrener, D.; Vyas, N.; and Kakade, S. 2025. Deconstructing What Makes a Good Optimizer for Autoregressive Language Models. In *International Conference on Learning Representations*.

Zhao, W.; Chen, Y.; Lee, R.; Qiu, X.; Gao, Y.; Fan, H.; and Lane, N. D. 2024. Breaking Physical and Linguistic Borders: Multilingual Federated Prompt Tuning for Low-Resource Languages. In *International Conference on Learning Representations*.

Zhou, Y.; Ram, P.; Salonidis, T.; Baracaldo, N.; Samulowitz, H.; and Ludwig, H. 2023. Single-shot General Hyperparameter Optimization for Federated Learning. In *International Conference on Learning Representations*.