Communication-Efficient Personalized Adaptation via Federated-Local Model Merging

Yinan Zou
Purdue University
zou211@purdue.edu

Md Kamran Chowdhury Shisher
University of Colorado Colorado Springs
mshisher@uccs.edu

Christopher G. Brinton
Purdue University
cgb@purdue.edu

Vishrant Tripathi
Purdue University
tripathv@purdue.edu

Abstract

Parameter-efficient fine-tuning methods, such as LoRA, offer a practical way to adapt large vision and language models to client tasks. However, this becomes particularly challenging under task-level heterogeneity in federated deployments. In this regime, personalization requires balancing *general knowledge* with *personalized knowledge*, yet existing approaches largely rely on heuristic mixing rules and lack theoretical justification. In this work, we propose POTARA, a principled framework for federated personalization that constructs a personalized model for each client by merging two complementary models: (i) a federated model capturing *general knowledge*, and (ii) a local model capturing *personalized knowledge*. Through the construct of linear mode connectivity, we show that the expected task loss admits a variance trace upper bound, whose minimization yields closed-form optimal mixing weights that guarantee a tighter bound for the merged model than for either the federated or local model alone. Experiments on vision and language benchmarks show that POTARA consistently improves personalization while reducing communication, leading to a strong performance-communication trade-off.

1 Introduction

Large vision models (LVMs) and large language models (LLMs) have achieved remarkable progress, enabling their widespread adoption across diverse downstream applications [65, 1]. Adapting pre-trained models to domain specific tasks typically requires fine-tuning on task relevant data, but full parameter fine-tuning is computationally expensive and impractical at scale. To mitigate this cost, parameter-efficient fine-tuning methods have been developed, among which low-rank adaptation (LoRA) [27] has become particularly popular. Domain specific fine-tuning of large models often requires large heterogeneous datasets which are typically split across organizations and cannot be centralized due to privacy and security constraints. Federated learning (FL) [36] addresses this issue by enabling collaborative training while keeping raw data local.

Federated large model fine-tuning [64] integrates FL with LoRA, allowing clients to update and communicate only LoRA modules while jointly learning a global model. Despite its promise, federated LoRA fine-tuning faces pronounced *task heterogeneity*: different clients may pursue distinct, and sometimes even conflicting, objectives. In such settings, the knowledge required by each client naturally decomposes into two complementary components: (i) *general knowledge* that can be transferred across clients and tasks, and (ii) *personalized knowledge* that is specific to each client's local data and task.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Challenges. A central challenge is how to balance cross-client *general knowledge* and client-specific *personalized knowledge*. It remains unclear, from a principled theoretical perspective, how much emphasis should be placed on general versus personalized knowledge when forming the final client model. Although recent methods attempt to provide a solution [62, 3, 46, 58], they lack theoretical justification, leaving a gap for theoretically-grounded solutions. Additionally, communication is the primary bottleneck in FL, and LoRA-based training still requires the repeated transmission of parameters across many rounds. These considerations call for an approach that (i) explicitly control the general-personalized trade-off in a principled manner and (ii) achieve favorable performance–communication trade-off in realistic federated deployments.

1.1 Contributions

Motivated by these limitations, this work aims to develop a framework for efficient federated fine-tuning under task heterogeneity. Our key contributions are as follows:

- We propose POTARA[1], a federated LoRA personalization framework to jointly (i) achieve an effective trade-off between federation and personalization, and (ii) balance communication efficiency with performance. POTARA constructs a personalized model for each client by merging two complementary models: a collaborative federated model that captures *general knowledge* across all clients, and a local fine-tuning model that emphasizes *personalized knowledge*.
- To determine the mixing weight between the federated model and the local model, we develop rigorous analysis grounded in linear mode connectivity (LMC). We show that, within an LMC basin around the optimum, the expected excess task losses of the federated model, the local model, and their merged model admits an upper bound in which the key term is a variance trace (see Theorem 3.6). Minimizing this bound yields closed-form optimal mixing weights, which guarantee that the merged model achieves the tighter bound than both the federated model and the local model (see Lemma 3.7).
- We conduct extensive experiments on vision and language benchmarks. The experimental results show that our method achieves superior performance compared to baseline methods while significantly reducing communication overhead, thereby offering a favorable performance-communication trade-off.

1.2 Related Work

Model Merging. Model merging aims to combine multiple task-specific models into a single model without additional training. Wortsman et al. [57] utilized averaged parameters to construct the merged model. To account for parameter importance, Matena and Raffel [35] proposed weighted fusion, where merging weights are derived from the Fisher information matrix (FIM) [19]. Jin et al. [30] formulated model merging as a linear regression problem and obtained a closed-form solution. Instead of averaging weights, Ilharco et al. [29] introduced task vectors by subtracting the pre-trained parameters from each fine-tuned model, and then linearly combining these task vectors to form a merged model. Yadav et al. [59] proposed to reduces conflicts in task vectors [29] by trimming low-magnitude updates, resolving sign disagreements, and merging disjoint parameter subsets with consistent signs. Yu et al. [63] further explored this issue by randomly dropping a subset of fine-tuned weights and rescaling the remainder, yielding sparse task vectors that alleviate destructive interference. Most existing methods [29, 59, 63, 43, 49] select merging weights via heuristic tuning (e.g., grid search), which is computation intensive. In contrast, our approach derives the optimal mixing weights from theory, substantially improving efficiency by eliminating expensive per-client weight sweeps and repeated evaluations.

Personalized Federated Learning. Personalized federated learning aims to overcome the limitation of a single global model by adapting client models to their local data distribution [52]. Common approaches include meta learning [17, 5], partial model personalization [9, 42], multi-task learning [48, 33], hypernetworks [47], clustering [15, 22] and mixtures of global and local models [24, 12]. While Deng et al. [12] studies the mixing weights of the global and local models theoretically, it depends on the distance between the client distribution and the average distribution, which is generally unavailable during federated training. In contrast, our method computes the optimal merging weight

[1] POTARA derives its name from the Potara fusion in Dragon Ball, where two independent entities are merged into a single, and more powerful form.

from the fisher information matrices of LoRA modules of the federated and local models. A related study on personalized FL and model merging, FedMerge [6], proposed a multi-model FL framework in which the server maintains a pool of global models, and each client obtains a personalized model by merging these server-side models. However, the server stores multiple global models and perform gradient-based updates for each model in the pool, both the storage and computation overhead on the server scale linearly with the number of global models. In contrast, our method follows a single-model paradigm where the server maintains only one global model and the server only needs to average the local models, making it substantially more resource efficient for large model fine-tuning. Moreover, the above methods primarily address statistical heterogeneity (e.g., label distribution skew) and are typically evaluated on conventional small model architectures. In contrast, we study a more realistic and challenging regime for large model adaptation, where heterogeneity manifests not only in label diversity but also in task diversity across clients.

Federated Fine-Tuning. FedIT [64] integrated LoRA with FedAvg and is one of the earliest efforts to combine parameter-efficient adaptation with federated optimization. Subsequent works studied federated LoRA under differential privacy [50], the asymmetric roles of LoRA matrices [23], and heterogeneous-rank LoRA matrices [18, 7, 2, 55, 67]. [25, 44] studied personalization under label-distribution heterogeneity while still assuming task-homogeneous adaptation. Considering task heterogeneity, [62, 3, 46] introduced global and local LoRA modules and combine them via a dynamic mixer. Although empirically effective, such designs lack theoretical justification for determining the appropriate mixing weights between these two LoRA modules. In contrast, our method derives the optimal mixing weight from a principled theoretical analysis.

2 Preliminaries and Notation

In this section, we introduce the preliminary concepts and notation used throughout the paper, including LoRA, federated fine-tuning, and linear mode connectivity.

2.1 Parameter-Efficient Fine-Tuning: LoRA

Consider a pre-trained model with parameters $\mathbf{W}_{\mathrm{Pre}} \in \mathbb{R}^{m\times n}$. We denote $\Delta\mathbf{W} \in \mathbb{R}^{m\times n}$ as the trainable update matrix during fine-tuning. Instead of updating all parameters in $\Delta\mathbf{W}$, LoRA [27] decomposes $\Delta\mathbf{W}$ into two low-rank matrices $\mathbf{A} \in \mathbb{R}^{r\times n}$ and $\mathbf{B} \in \mathbb{R}^{m\times r}$, where $r \ll \min(m, n)$. This decomposition allows the fine-tuning to focus on the significantly smaller low-rank matrices $\mathbf{A}$ and $\mathbf{B}$ instead of the full matrix $\Delta\mathbf{W}$. Consequently, the total number of trainable parameters is reduced from $m \times n$ to $r \times (m + n)$. The model parameters after fine-tuning are given by $\mathbf{W}_{\mathrm{FT}} = \mathbf{W}_{\mathrm{Pre}} + \Delta\mathbf{W} = \mathbf{W}_{\mathrm{Pre}} + \mathbf{BA}$, where $\mathbf{A}$ is typically initialized with random Gaussian values and $\mathbf{B}$ is initialized to zero [27, 26].

2.2 Federated LoRA Fine-Tuning

FedIT [64] integrates LoRA [27] with the FedAvg [36] framework. FedIT aims to minimize a global loss function subject to the consensus constraints $\mathbf{B}^{(1)} = \ldots = \mathbf{B}^{(N)}$ and $\mathbf{A}^{(1)} = \ldots = \mathbf{A}^{(N)}$ where $\mathbf{B}^{(i)}$ and $\mathbf{A}^{(i)}$ denotes the LoRA matrices at client i. Consequently, it ultimately yields a single model shared across all clients.

2.3 Linear Mode Connectivity

Prior works [14, 21] show that the minima found by two independently trained neural networks can be connected by a simple path along which the loss or accuracy remains nearly constant. This phenomenon is referred to as mode connectivity. Subsequent studies [37, 20] further show that when neural networks are trained from the same initialization, the resulting well-trained models can be connected by a linear path, known as linear mode connectivity (LMC). We restate the definition of LMC used in [20, 66].

Definition 2.1. [20, 66] Given a dataset $\mathcal{D}$ and two well-trained models $\mathbf{W}_1$ and $\mathbf{W}_2$ such that $f(\mathbf{W}_1; \mathcal{D}) \approx f(\mathbf{W}_2; \mathcal{D})$, we say that $\mathbf{W}_1$ and $\mathbf{W}_2$ are linearly connected if they satisfy

$$f(\lambda_1\mathbf{W}_1 + \lambda_2\mathbf{W}_2; \mathcal{D}) \approx f(\mathbf{W}_1; \mathcal{D}),\ \forall\, \lambda_1, \lambda_2 \geq 0,\ \lambda_1 + \lambda_2 = 1, \tag{1}$$

where $f(\mathbf{W}; \mathcal{D})$ denotes the loss of model $\mathbf{W}$ on dataset $\mathcal{D}$.

In Definition 2.1, two models $\mathbf{W}_1$ and $\mathbf{W}_2$ are linearly connected if the loss of the interpolated mode remains nearly unchanged. In other words, $\mathbf{W}_1$ and $\mathbf{W}_2$ lie within the same flat basin of the loss landscape, and their linear interpolation remains inside this basin [39].

3 Our Method: POTARA

In this section, we propose POTARA, which achieves a principled trade-off between federation and personalization while simultaneously balancing communication efficiency and performance.

3.1 Motivation

In large model fine-tuning, different clients usually hold data from distinct tasks (e.g., question classification versus sentiment analysis) [17, 9, 3, 62, 53]. In such task-heterogeneous settings, consensus-based FL methods often underperform, since enforcing a single shared solution neglects both task diversity and the need for client-specific adaptation. A trivial alternative is to let each client fine-tune independently without communication. However, the number of local data samples is typically insufficient to train a satisfactory model in practical settings. By exchanging information across clients, each client can exploit transferable knowledge learned elsewhere to improve its own model. Accordingly, personalized federated fine-tuning aims to learn a client-specific model that effectively combines *general knowledge* across clients with *personalized knowledge* adapted to the client's local task.

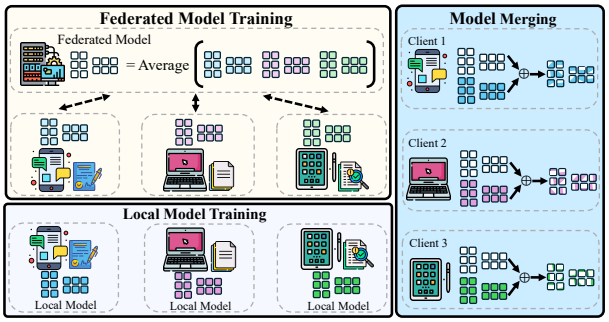


Figure 1: Illustration of POTARA. We train a FedIT model via FL and train local fine-tuning models. We obtain the final personalized model by weighted merging these FedIT and local models.

3.2 Federated-Local Model Merging

Model merging provides a lightweight mechanism for combining complementary capabilities from multiple fine-tuned models without additional joint training. Building on this idea, we propose POTARA, which constructs a personalized model for each client by merging two complementary models:

- A collaborative FedIT model learned through cross-client communication,
- A purely local model fine-tuned only on that client's data.

Intuitively, the FedIT model captures *general knowledge* distilled from all clients. In contrast, the local model emphasizes *personalized knowledge* tailored to the client's task and distribution. By merging them, POTARA yields a client-specific model that simultaneously leverages federation and personalization, offering an effective trade-off between these two objectives.

Let $\boldsymbol{\theta}^{(i)}_{\text{Local}}$ denote the task vector corresponding to the LoRA update of client i's locally fine-tuned model, and let $\boldsymbol{\theta}_{\text{FedIT}}$ denote the task vector corresponding to the LoRA update of the FedIT model [29]. For each client i, we form the merged task vector by a convex combination:

$$\boldsymbol{\theta}^{(i)}_{\text{Merge}} = \lambda^{(i)}_{\text{FedIT}}\, \boldsymbol{\theta}_{\text{FedIT}} + \lambda^{(i)}_{\text{Local}}\, \boldsymbol{\theta}^{(i)}_{\text{Local}}, \tag{2}$$

where $\lambda^{(i)}_{\text{FedIT}}, \lambda^{(i)}_{\text{Local}} \geq 0$ denote the mixing weights for client i and $\lambda^{(i)}_{\text{FedIT}} + \lambda^{(i)}_{\text{Local}} = 1$. Then, the final model for each client is given by

$$\mathbf{W}^{(i)}_{\text{Final}} = \mathbf{W}_{\text{Pre}} + \boldsymbol{\theta}^{(i)}_{\text{Merge}}. \tag{3}$$

A natural question then arises: how to choose the mixing weights $\lambda^{(i)}_{\text{FedIT}}$ and $\lambda^{(i)}_{\text{Local}}$? This choice is critical, as it governs the trade-off between leveraging *general knowledge* from FedIT and preserving *personalized knowledge* from local fine-tuning. Next, we present our weighting strategy for determining $\lambda^{(i)}_{\text{FedIT}}$ and $\lambda^{(i)}_{\text{Local}}$ in a principled manner.

3.3 Determination of Mixing Weights

In the preliminaries, we recalled the standard definition of LMC, which characterizes the phenomenon in terms of the loss remaining nearly constant along the linear interpolation path. To obtain a local geometric characterization of LMC, we introduce the notion of LMC basin below.

Definition 3.1. For a given radius $\delta > 0$, define the LMC basin around the optimum $\mathbf{W}^*$ as

$$\mathcal{U}_\delta = \{\mathbf{W} : \|\mathbf{W} - \mathbf{W}^*\| \leq \delta\}. \tag{4}$$

Two models $\mathbf{W}_1$ and $\mathbf{W}_2$ exhibit linear mode connectivity in $\mathcal{U}_\delta$ if all their convex combination lies entirely in $\mathcal{U}_\delta$, i.e.,

$$\lambda_1 \mathbf{W}_1 + \lambda_2 \mathbf{W}_2 \in \mathcal{U}_\delta, \quad \forall\, \lambda_1, \lambda_2 \geq 0,\ \lambda_1 + \lambda_2 = 1. \tag{5}$$

We assume that the FedIT and each local model exhibit LMC within the LMC basin, i.e., the entire linear interpolation path between them remains inside the basin.

Assumption 3.2. Let $\mathcal{U}_\delta$ be the LMC basin in Definition 3.1. For each client i, we assume that $\mathbf{W}_{\text{Pre}} + \lambda^{(i)}_{\text{FedIT}}\, \boldsymbol{\theta}_{\text{FedIT}} + \lambda^{(i)}_{\text{Local}}\, \boldsymbol{\theta}^{(i)}_{\text{Local}} \in \mathcal{U}_\delta, \quad \forall\, \lambda^{(i)}_{\text{FedIT}}, \lambda^{(i)}_{\text{Local}} \geq 0,\ \lambda^{(i)}_{\text{FedIT}} + \lambda^{(i)}_{\text{Local}} = 1.$

Note that our setting is fine-tuning from a shared pre-trained model, so both the federated and local models can be viewed as relatively small perturbations around an already strong model for related tasks. This suggests that they may remain in a common basin. We provide empirical evidence that supports the LMC assumption under task heterogeneity in Appendices J. Throughout our analysis, we focus on the behavior of the loss landscape within the LMC basin and make the following standard assumptions on optimization.

Assumption 3.3. The task loss function $\mathcal{L}$ is locally Lipschitz smooth within the LMC basin $\mathcal{U}_\delta$. There exist constants $L > 0$ such that

$$\mathcal{L}(\mathbf{y}) \leq \mathcal{L}(\mathbf{x}) + \nabla\mathcal{L}(\mathbf{x})^\top(\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|^2,\ \forall\, \mathbf{x}, \mathbf{y} \in \mathcal{U}_\delta. \tag{6}$$

Equivalently, the Hessian of $\mathcal{L}$ is bounded within the LMC basin, i.e., $\nabla^2\mathcal{L}(\mathbf{x}) \preceq L\,\mathbf{I},\ \forall\, \mathbf{x} \in \mathcal{U}_\delta$.

Assumption 3.4. [38, 4] The Hessian of the task loss within the LMC basin is L_H-Lipschitz continuous, i.e., $\|\nabla^2\mathcal{L}(\mathbf{y}) - \nabla^2\mathcal{L}(\mathbf{x})\| \leq L_H\|\mathbf{y} - \mathbf{x}\|, \forall\, \mathbf{x}, \mathbf{y} \in \mathcal{U}_\delta$, where $L_H > 0$ is a constant.

In the following, we show that the model obtained by merging the FedIT model and the local model yields a personalized performance gain. For clarity, we omit the client index.

We use the Laplace approximation to approximate the task vectors of FedIT (i.e., $\boldsymbol{\theta}_{\text{FedIT}}$) and local fine-tuning (i.e., $\boldsymbol{\theta}_{\text{Local}}$) as Gaussian distributions:

$$\boldsymbol{\theta}_{\text{FedIT}} \sim \mathcal{N}(\boldsymbol{\mu}_{\text{FedIT}}, \boldsymbol{\Sigma}_{\text{FedIT}}), \tag{7}$$

$$\boldsymbol{\theta}_{\text{Local}} \sim \mathcal{N}(\boldsymbol{\mu}_{\text{Local}}, \boldsymbol{\Sigma}_{\text{Local}}), \tag{8}$$

where $\boldsymbol{\mu}_{\text{FedIT}}$ and $\boldsymbol{\mu}_{\text{Local}}$ denote the MAP parameter estimates of the corresponding task vectors, and $\boldsymbol{\Sigma}_{\text{FedIT}} = \mathbf{F}^{-1}_{\boldsymbol{\mu}_{\text{FedIT}}}$ and $\boldsymbol{\Sigma}_{\text{Local}} = \mathbf{F}^{-1}_{\boldsymbol{\mu}_{\text{Local}}}$ are given by the inverses of their Fisher information matrices. The full derivation is provided in Appendix A.1.

We further assume a joint Gaussian distribution of FedIT and local task vectors, which introduces an explicit cross-covariance term.

Assumption 3.5. [45, 10, 35, 56] We assume that $(\boldsymbol{\theta}_{\text{FedIT}}, \boldsymbol{\theta}_{\text{Local}})$ follow a joint Gaussian distribution:

$$\begin{bmatrix} \boldsymbol{\theta}_{\text{FedIT}} \\ \boldsymbol{\theta}_{\text{Local}} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_{\text{FedIT}} \\ \boldsymbol{\mu}_{\text{Local}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\text{FedIT}} & \boldsymbol{\Sigma}_{\text{Cross}} \\ \boldsymbol{\Sigma}_{\text{Cross}}^\top & \boldsymbol{\Sigma}_{\text{Local}} \end{bmatrix} \right), \tag{9}$$

where $\boldsymbol{\Sigma}_{\text{Cross}}$ denotes the cross-covariance matrix between the two task vectors. We assume that FedIT task and local task are correlated, therefore, the cross-covariance matrix satisfies $\boldsymbol{\Sigma}_{\text{Cross}} \succeq 0$. Furthermore, we assume that $\boldsymbol{\Sigma}_{\text{Cross}} \preceq \boldsymbol{\Sigma}_{\text{FedIT}}$ and $\boldsymbol{\Sigma}_{\text{Cross}} \preceq \boldsymbol{\Sigma}_{\text{Local}}$.

The conditions $\boldsymbol{\Sigma}_{\text{Cross}} \preceq \boldsymbol{\Sigma}_{\text{FedIT}}$ and $\boldsymbol{\Sigma}_{\text{Cross}} \preceq \boldsymbol{\Sigma}_{\text{Local}}$ indicate that the cross-covariance is not larger than the marginal variances, i.e., the shared uncertainty does not dominate each model's own uncertainty. We provide further discussion and justification of this assumption in Appendix K.

With the joint Gaussian model in Assumption 3.5, we can derive the Gaussian distribution of the merged task vector:

$$\boldsymbol{\theta}_{\mathrm{Merge}} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathrm{Merge}}, \boldsymbol{\Sigma}_{\mathrm{Merge}}), \tag{10}$$

where

$$\boldsymbol{\mu}_{\mathrm{Merge}} = \lambda_{\mathrm{FedIT}}\boldsymbol{\mu}_{\mathrm{FedIT}} + \lambda_{\mathrm{Local}}\boldsymbol{\mu}_{\mathrm{Local}}, \tag{11}$$

$$\boldsymbol{\Sigma}_{\mathrm{Merge}} = \lambda_{\mathrm{FedIT}}^2\boldsymbol{\Sigma}_{\mathrm{FedIT}} + \lambda_{\mathrm{Local}}^2\boldsymbol{\Sigma}_{\mathrm{Local}} + \lambda_{\mathrm{FedIT}}\lambda_{\mathrm{Local}}(\boldsymbol{\Sigma}_{\mathrm{Cross}} + \boldsymbol{\Sigma}_{\mathrm{Cross}}^\top). \tag{12}$$

We now present an expected excess loss bound for FedIT, local, and their merged model.

Theorem 3.6. *Under Definition 3.1, Assumption 3.2, 3.3, 3.4, and 3.5, for* $i \in \{\mathrm{FedIT}, \mathrm{Local}, \mathrm{Merge}\}$*, it holds that*

$$\mathbb{E}[\mathcal{L}(\mathbf{W}_{\mathrm{Pre}} + \boldsymbol{\theta}_i) - \mathcal{L}(\mathbf{W}^*)] \leq \left(\frac{L}{2} + \frac{L_H}{6}\delta\right)\delta^2 + \left(\frac{L}{2} + \frac{L_H}{6}\delta\right)\mathrm{tr}(\boldsymbol{\Sigma}_i). \tag{13}$$

Theorem 3.6 indicates that the excess loss upper bound is primarily controlled by the trace term. For the merged model, $\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}})$ depends on the mixing weights $(\lambda_{\mathrm{FedIT}}, \lambda_{\mathrm{Local}})$. In the following, we minimize this trace term to obtain the optimal mixing weights in closed form.

Lemma 3.7. *Minimizing the trace term* $\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}})$ *yields the optimal mixing weights, given by*

$$\lambda_{\mathrm{FedIT}}^* = \frac{b-c}{a+b-2c}, \qquad \lambda_{\mathrm{Local}}^* = \frac{a-c}{a+b-2c}, \tag{14}$$

where $a = \mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{FedIT}})$, $b = \mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Local}})$, $c = \mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Cross}})$. *At this optimum, the merged trace is smaller than those of the two individual models:*

$$\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}}) \leq \min\{\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{FedIT}}),\ \mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Local}})\}. \tag{15}$$

In (14), a smaller value of a or b implies lower variance and hence a more reliable model, which leads to a larger weight for that model. At the optimal weights $(\lambda_{\mathrm{FedIT}}^*, \lambda_{\mathrm{Local}}^*)$ in Lemma 3.7, $\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}})$ attains its minimum. Moreover, this minimum satisfies $\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}}) \leq \min\{\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{FedIT}}),\ \mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Local}})\}$. Therefore, by Theorem 3.6 and Lemma 3.7, the merged model admits an excess loss upper bound that is tighter than the bounds for FedIT or local fine-tuning alone.

3.4 Estimation of Fisher Information Matrix

In Section 3.3, we established that $\boldsymbol{\Sigma}_{\mathrm{FedIT}} = \mathbf{F}_{\boldsymbol{\mu}_{\mathrm{FedIT}}}^{-1}$ and $\boldsymbol{\Sigma}_{\mathrm{Local}} = \mathbf{F}_{\boldsymbol{\mu}_{\mathrm{Local}}}^{-1}$, i.e., each posterior variance is given by the inverse Fisher information matrix. To reduce the computation cost, we adopt a diagonal approximation of the Fisher information matrix [35, 32]. The diagonal Fisher matrix evaluated at the posterior mean $\boldsymbol{\mu}_i$ is given by

$$\widehat{\mathbf{F}}_{\boldsymbol{\mu}_i} = \mathrm{diag}\left(\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{y}\sim p_{\boldsymbol{\mu}_i}(\boldsymbol{y}|\boldsymbol{x}_n)}\left[\left(\nabla_{\boldsymbol{\mu}_i}\log p_{\boldsymbol{\mu}_i}(\boldsymbol{y}\mid\boldsymbol{x}_n)\right)^{\odot 2}\right]\right), \tag{16}$$

where N is the mini-batch size to estimate the Fisher information and $(\cdot)^{\odot 2}$ denotes the element-wise square operator. Consequently, the posterior variance can be approximated as $\boldsymbol{\Sigma}_i \approx (\widehat{\mathbf{F}}_{\boldsymbol{\mu}_i})^{-1}$.

To approximate the cross-covariance between the two task vectors, we compute per-sample gradients under $\boldsymbol{\mu}_{\mathrm{FedIT}}$ and $\boldsymbol{\mu}_{\mathrm{Local}}$, estimate their coordinate-wise variances and covariance, and obtain a gradient correlation coefficient ρ_k for each parameter dimension. We then construct a diagonal approximation

$$(\boldsymbol{\Sigma}_{\mathrm{Cross}})_{kk} \ \approx\ \rho_k\sqrt{(\boldsymbol{\Sigma}_{\mathrm{FedIT}})_{kk}(\boldsymbol{\Sigma}_{\mathrm{Local}})_{kk}}, \tag{17}$$

where k denotes the parameter coordinate and ρ_k is clipped to ensure $\boldsymbol{\Sigma}_{\mathrm{Cross}} \succeq 0$ and $\boldsymbol{\Sigma}_{\mathrm{Cross}} \preceq \boldsymbol{\Sigma}_{\mathrm{FedIT}}, \boldsymbol{\Sigma}_{\mathrm{Local}}$. Further details on the merging procedure are provided in the Appendix B.

4 Experiments

In this section, we evaluate and compare the performance of the proposed POTARA against baseline methods on vision and language benchmarks. The baseline methods include local fine-tuning [27], FedIT [64], FFA-LoRA [50], FedSA [23], APFL [12], L2GD [24], FedDPA [62], FedALT [3], FedMosaic [46], PerAda [58], and Fisher Merging [35].

Table 1: Accuracy comparison on personalized CIFAR-100 benchmark. POTARA obtains consistent improvement over the baselines across all clients.

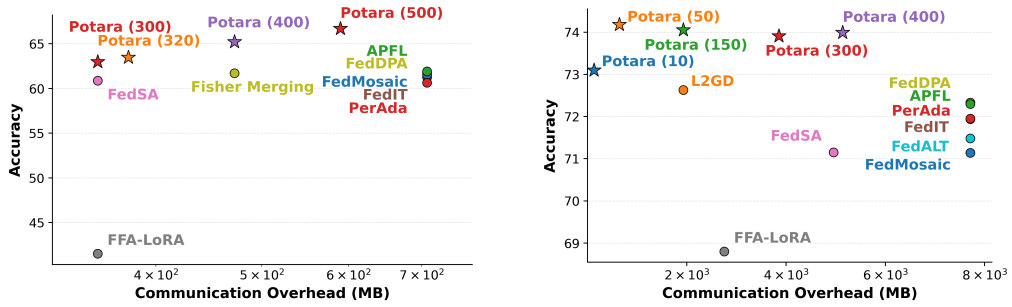
Personalized CIFAR-100 benchmark (ViT-B/16 model)

Methods	Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client 7	Client 8	Client 9	Client 10	Avg.
Local	60.01 ±0.48	59.78 ±0.94	63.26 ±2.68	61.79 ±2.06	61.26 ±1.07	56.46 ±4.77	60.62 ±1.72	61.79 ±1.75	58.75 ±2.82	60.53 ±4.29	60.43 ±0.93
FFA-LoRA	39.96 ±3.87	44.55 ±2.91	42.36 ±6.53	40.14 ±2.98	42.85 ±1.62	38.62 ±0.80	41.50 ±3.63	40.03 ±2.37	44.07 ±1.86	41.08 ±6.12	41.52 ±2.44
FedSA	62.24 ±0.44	63.11 ±1.41	61.79 ±5.53	61.16 ±2.72	59.44 ±0.99	59.17 ±1.94	61.11 ±1.54	59.73 ±0.45	59.71 ±2.02	61.24 ±4.27	60.87 ±0.97
FedIT	62.55 ±2.33	60.53 ±2.37	60.30 ±4.86	59.48 ±3.38	62.07 ±3.63	59.99 ±1.51	61.93 ±3.14	62.73 ±1.52	63.60 ±3.15	59.61 ±3.35	61.28 ±0.29
FedDPA	62.80 ±1.46	60.51 ±2.79	61.22 ±4.81	62.41 ±2.87	61.54 ±2.43	60.31 ±1.71	63.24 ±1.94	62.09 ±2.13	62.74 ±1.82	61.14 ±4.61	61.80 ±0.96
Fisher Merging	60.25 ±3.04	60.27 ±1.96	63.28 ±4.56	59.76 ±1.54	60.99 ±0.56	58.06 ±1.05	59.13 ±2.13	58.66 ±0.71	58.08 ±2.43	57.57 ±4.86	59.60 ±1.13
L2GD	55.50 ±2.43	52.75 ±4.07	57.43 ±9.88	56.27 ±2.29	54.12 ±1.31	52.46 ±2.62	55.39 ±4.07	52.84 ±4.66	56.87 ±6.99	50.63 ±10.88	54.43 ±4.92
FedMosaic	62.88 ±0.89	62.27 ±1.23	62.26 ±4.73	62.05 ±2.37	62.98 ±1.09	58.28 ±0.93	62.36 ±1.99	60.19 ±0.32	61.49 ±2.99	60.59 ±6.45	61.53 ±1.29
APFL	63.36 ±1.36	61.45 ±1.95	63.44 ±2.24	61.48 ±1.61	64.06 ±0.34	58.60 ±1.46	63.11 ±1.87	61.56 ±1.04	61.81 ±0.29	60.14 ±4.30	61.91 ±0.02
PerAda	60.79 ±0.85	61.58 ±0.31	60.36 ±3.25	60.77 ±1.09	62.90 ±0.14	61.22 ±1.68	61.51 ±0.01	58.58 ±3.89	60.35 ±0.02	58.35 ±6.51	60.64 ±0.54
POTARA	63.66 ±2.48	64.95 ±1.06	65.15 ±5.59	62.50 ±2.24	62.47 ±1.95	60.36 ±1.64	63.54 ±2.72	62.58 ±1.21	63.14 ±0.73	61.51 ±3.17	62.98 ±1.47

Table 2: Accuracy comparison on commonsense reasoning benchmark. As in Table 1, POTARA obtains consistent improvements.

Commonsense reasoning benchmark (Llama-3.2-3B-Instruct model)

Methods	ARC-c	ARC-e	BoolQ	HellaSwag	OBQA	PIQA	SIQA	WinoGrande	Avg.
Local	72.13 ±0.90	86.20 ±1.66	66.27 ±2.78	68.93 ±1.54	75.40 ±0.75	77.27 ±1.36	66.13 ±1.64	64.07 ±2.37	72.05 ±0.25
FFA-LoRA	73.27 ±0.50	85.53 ±0.52	64.80 ±0.91	57.73 ±2.78	73.87 ±0.50	79.07 ±1.52	62.00 ±7.09	54.13 ±1.18	68.80 ±0.77
FedSA	71.53 ±1.25	84.67 ±1.05	64.27 ±2.36	64.93 ±2.73	74.87 ±0.93	77.73 ±0.81	66.53 ±1.39	64.67 ±2.84	71.15 ±0.62
FedIT	74.27 ±0.62	85.93 ±1.46	66.87 ±1.18	64.93 ±1.05	74.33 ±1.64	78.80 ±1.77	69.87 ±1.09	60.53 ±1.75	71.94 ±0.42
FedALT	72.73 ±0.46	85.00 ±0.35	64.27 ±0.58	68.33 ±0.92	76.67 ±0.12	79.00 ±0.00	65.53 ±0.58	60.33 ±4.39	71.48 ±0.58
FedDPA	74.73 ±1.25	87.60 ±0.99	63.80 ±1.23	65.73 ±1.39	77.07 ±0.50	80.40 ±1.88	67.87 ±0.66	61.47 ±0.66	72.33 ±0.37
L2GD	74.70 ±0.30	88.30 ±1.10	63.20 ±1.00	68.30 ±0.50	76.90 ±0.70	80.60 ±0.40	69.00 ±1.00	60.10 ±2.90	72.63 ±0.57
FedMosaic	74.20 ±1.40	86.80 ±1.20	61.90 ±7.10	65.80 ±0.60	74.30 ±0.70	77.70 ±1.50	68.20 ±0.20	60.20 ±3.00	71.14 ±0.98
APFL	75.40 ±2.80	85.80 ±0.80	63.80 ±3.40	67.30 ±0.30	74.70 ±1.10	78.40 ±1.20	70.00 ±1.80	63.00 ±1.60	72.30 ±0.25
PerAda	70.80 ±3.20	86.10 ±0.50	65.20 ±3.40	71.50 ±0.70	76.50 ±0.50	79.50 ±0.90	65.90 ±2.30	60.10 ±2.30	71.95 ±0.05
POTARA	75.20 ±1.28	87.80 ±1.31	67.87 ±0.57	71.73 ±2.23	78.47 ±1.04	78.80 ±2.36	69.00 ±1.18	64.53 ±2.00	74.18 ±0.29



(a) Personalized CIFAR-100 benchmark

(b) Commonsense reasoning benchmark

Figure 2: Performance-communication trade-off across different benchmarks. POTARA (t) indicates that the federated model used for merging is the FedIT model trained for t rounds. For example, POTARA (50) merges the FedIT model trained for 50 rounds and the local model. POTARA achieves strong performance with substantially lower communication overhead than the baselines.

4.1 Vision Benchmark

We adopt ViT-B/16 [13] as the pre-trained model. To simulate label heterogeneity, we partition CIFAR-100 [31] across clients using a Dirichlet distribution with concentration parameter $\alpha = 0.5$. For each client, we construct both the training and test splits according to the same Dirichlet-based partition. Unless otherwise specified, we consider 10 clients. All baselines are trained for 600 rounds. For both Fisher Merging and the proposed POTARA, we use the same total training budget as the baselines. Specifically, we first train the FedIT model for 300 rounds and the local fine-tuning model for 300 rounds, resulting in 600 total training rounds, and then merge the two models to obtain the final personalized model. As shown in Table 1, the proposed POTARA achieves the best overall performance among all baseline methods under the same total training budget. Moreover, POTARA reduces the communication cost. See Section 4.3 and Figure 2a for a detailed performance-communication trade-off analysis.

4.2 Language Benchmark

We next evaluate POTARA on LLM and NLP benchmark. We adopt LLaMA-3.2-3B-Instruct [16] as the pre-trained model and construct a task-heterogeneous setting by assigning each client a distinct task from the commonsense reasoning benchmark [28] and report accuracy for all tasks. We also

Table 3: Performance gain of model merging. POTARA consistently improves over both endpoint models used for merging: the local model and the FedIT model.

(a) CIFAR-100

Model	Accuracy (↑)	Gain over Local	Gain over FedIT
Local (300 rounds)	55.89 ± 1.31	–	–
FedIT (300 rounds)	55.42 ± 1.78	–	–
POTARA	$\mathbf{62.98 \pm 1.47}$	$\mathbf{+7.09}$	$\mathbf{+7.56}$

(b) Commonsense reasoning

Model	Accuracy (↑)	Gain over Local	Gain over FedIT
Local (300 rounds)	72.05 ± 0.25	–	–
FedIT (50 rounds)	67.73 ± 0.76	–	–
POTARA	$\mathbf{74.18 \pm 0.29}$	$\mathbf{+2.13}$	$\mathbf{+6.45}$

evaluate POTARA with Qwen3-4B on the FLAN benchmark, with the results provided in Appendix E. Each client has 1,000 training examples and 500 test examples. All federated baseline methods are trained for 600 rounds, while the local fine-tuning baseline is trained for 300 rounds to prevent overfitting. For POTARA, we train FedIT for 50 rounds and local fine-tuning for 300 rounds, and then merge the resulting models to obtain the final personalized model for each client. As reported in Table 2, POTARA consistently outperforms all baselines across the evaluated NLP tasks. Moreover, POTARA uses substantially fewer training rounds than federated baselines while still achieving superior performance. See Section 4.3 and Figure 2b for a detailed communication and computation efficiency analysis. The gains observed across both large vision (ViT) and language models (LLaMA) demonstrate that POTARA generalizes well across model families, parameter scales, and task types.

4.3 Communication and Computation Efficiency

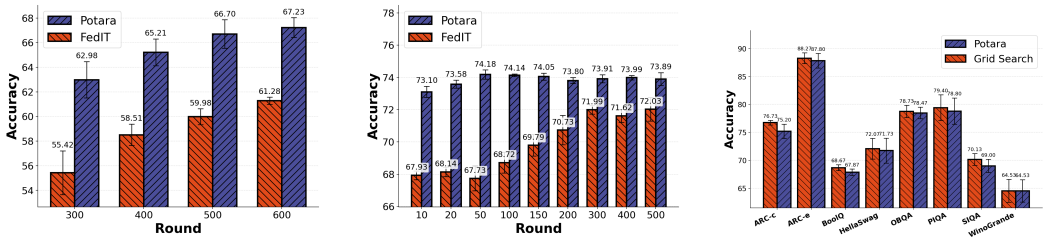
Table 4: Computation Time and Performance Comparison.

Methods	Federated Training	Local Training	FIM Estimation	Covariance Estimation	Total Time	Avg. Acc.
FedIT	5.92 h	N/A	N/A	N/A	5.92 h	71.94
FedDPA	11.1 h	N/A	N/A	N/A	11.1 h	72.33
FedALT	8.50 h	N/A	N/A	N/A	8.50 h	71.48
APFL	11.0 h	N/A	N/A	N/A	11.0 h	72.30
FedMosaic	6.33 h	N/A	N/A	N/A	6.33 h	71.14
PerAda	11.3 h	N/A	N/A	N/A	11.3 h	71.95
POTARA	0.49 h	2.96 h	0.43 h	0.03 h	3.91 h	74.18

In this subsection, we compare the communication and computation cost of POTARA with baselines. For ViT-B/16, LoRA introduces 0.59 MB parameters for the $\mathbf{A}$ matrices and 0.59 MB for the $\mathbf{B}$ matrices in float32 precision. For LLaMA-3.2-3B-Instruct, LoRA introduces 8.26 MB parameters for the $\mathbf{A}$ matrices and 4.59 MB for the $\mathbf{B}$ matrices in float32 precision. We use POTARA (t) to denote our method where the FedIT model obtained at round t is used for merging with the local model at round 300. For example, POTARA (50) merges the FedIT model trained for 50 rounds and local model trained for 300 rounds. As shown in Fig. 2, POTARA consistently achieves a favorable performance-communication trade-off across benchmarks. Increasing the number of federated rounds leads to higher communication cost, but also improves the performance of POTARA by providing a stronger federated model for merging. On CIFAR-100, POTARA (300) achieves around 63% accuracy with only 354 MB communication, outperforming most baselines clustered around 708 MB. With more federated rounds, POTARA (400) and POTARA (500) further improve the accuracy to 65.21% and 66.70%, respectively. On commonsense reasoning, POTARA (50) reaches 74.18% accuracy with only 642.5 MB communication, outperforming baselines that require roughly 5–8×10^3 MB communication. Compared with high-communication baselines such as FedIT, FedALT, FedDPA, APFL, FedMosaic, and PerAda, POTARA attains higher accuracy with substantially lower communication cost. Compared with low-communication baselines such as FedSA, FFA-LoRA, and L2GD, POTARA delivers clear performance gains at comparable or lower overhead. These results demonstrate that POTARA can achieve stronger performance with substantially lower communication cost across different benchmarks. Table 4 further compares the computation time on the commonsense reasoning benchmark. Although POTARA includes additional steps for local training, FIM estimation, and covariance estimation, its total computation time is only 3.91 hours, which is lower than baselines. A detailed computation cost comparison is provided in Appendix D. These results demonstrate that POTARA achieves superior performance while maintaining both communication and computation efficiency.

4.4 Performance Gain of Model Merging

We further analyze the benefit of the model merging by comparing POTARA with the two models used for merging: the FedIT model and the local model. On CIFAR-100, POTARA merges the FedIT model trained for 300 rounds with the local model trained for 300 rounds. As shown in Table 3a, the merged model achieves 62.98 accuracy, improving over the local and FedIT models by 7.09 and 7.56 points, respectively. On the commonsense reasoning benchmark, POTARA merges the FedIT



(a) Effect of federated training round for merging (CIFAR-100).

(b) Effect of federated training round for merging (Commonsense Reasoning).

(c) Client performance comparison between mixing weights computed by our method and mixing weights selected by grid search.

Figure 3: Ablation studies demonstrating the performance of POTARA.

model trained for 50 rounds with the local model trained for 300 rounds, achieving 74.18 accuracy and outperforming the local and FedIT models by 2.13 and 6.45 points, respectively, as shown in Table 3b. These results show that model merging brings consistent gains over both individual models. This suggests that the FedIT model and the local model capture complementary information: the former provides global knowledge obtained from collaborative training, while the latter preserves client-specific knowledge learned from local data. By combining these two sources of knowledge, POTARA achieves substantially better personalized performance than either model alone.

4.5 Ablation Study

Effect of federated training round for merging: We examine the effect of the federated training round used for merging. Specifically, we merge the local model trained for 300 rounds with FedIT models trained for different numbers of rounds. As shown in Figure 3a and Figure 3b, POTARA improves steadily as the FedIT model is trained longer on both benchmarks, indicating that a better-optimized FedIT model provides stronger general knowledge that can be more effectively combined with the local model. On personalized CIFAR-100 benchmark (Figure 3a), increasing the FedIT training rounds yields a clear monotonic gain for the merged model (e.g., from 62.98 to 67.23). On the commonsense reasoning benchmark (Figure 3b), the merged model already achieves strong performance even when using an early FedIT model (e.g., round 50), despite the FedIT still improving substantially with more rounds. This suggests that POTARA can effectively extract useful general knowledge from intermediate FedIT models and combine it with local model.

Effect of mixing weight strategy: We compare client performance using the mixing weights computed by our method with the best-performing weights selected by grid search over $\{0, 0.1, \ldots, 1\}$. As shown in Figure 3c, the two bars are closely matched across clients, demonstrating that our computed weights achieve performance comparable to grid search while eliminating the expensive per-client hyperparameter sweep.

Further experiments: Results with large-scale client settings are reported in Appendix G. Results with different LoRA ranks are reported in Appendix F. Results with a larger pre-trained model and two additional benchmarks are reported in Appendix E and Appendix I. Across all these settings, POTARA consistently achieves performance gains over the baselines.

5 Conclusion

We proposed POTARA, a principled personalization framework for federated fine-tuning that merges a federated model and a local model. We approximate the federated and local task vectors as jointly Gaussian random variables and derive the upper bound on the expected excess task loss of the federated model, the local model, and their merged model, where the dominant term is a trace of the task vector variance. By minimizing this bound, we obtain closed-form optimal mixing weights, guaranteeing that the merged model attains a tighter bound than both constituent models. Experiment results show that POTARA consistently improves client performance across both vision and language benchmarks, outperforming baselines while retaining communication efficiency. One limitation is that POTARA focuses on merging two models. Future work includes extending POTARA to multi-model merging, and dynamic personalization settings where client distributions evolve over time.

Acknowledgments

This work was supported in part by the Office of Naval Research (ONR) under grants N00014-23-C-1016 and N00014-22-1-2305, the National Science Foundation (NSF) under grants CNS-2146171 and CPS-2313109, and by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-24-1-0083.

References

[1] M. Awais, M. Naseer, S. Khan, R. M. Anwer, H. Cholakkal, M. Shah, M.-H. Yang, and F. S. Khan. Foundation models defining a new era in vision: a survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[2] J. Bai, D. Chen, B. Qian, L. Yao, and Y. Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. *Advances in Neural Information Processing Systems*, 2024.

[3] J. Bian, L. Wang, L. Zhang, and J. Xu. Fedalt: Federated fine-tuning through adaptive local training with rest-of-world lora. *AAAI Conference on Artificial Intelligence*, 2026.

[4] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

[5] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

[6] S. Chen, T. Zhou, G. Long, J. Jiang, and C. Zhang. Fedmerge: Federated personalization via model merging. *AAAI Conference on Artificial Intelligence*, 2026.

[7] Y. J. Cho, L. Liu, Z. Xu, A. Fahrezi, and G. Joshi. Heterogeneous lora for federated fine-tuning of on-device foundation models. *Conference on Empirical Methods in Natural Language Processing*, 2024.

[8] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[9] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai. Exploiting shared representations for personalized federated learning. *International Conference on Machine Learning*, 2021.

[10] N. Daheim, T. Möllenhoff, E. Ponti, I. Gurevych, and M. E. Khan. Model merging by uncertainty-based gradient matching. In *The Twelfth International Conference on Learning Representations*.

[11] E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig. Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 2021.

[12] Y. Deng, M. M. Kamani, and M. Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, and S. Gelly. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021.

[14] F. Draxler, K. Veschgini, M. Salmhofer, and F. Hamprecht. Essentially no barriers in neural network energy landscape. *International Conference on Machine Learning*, 2018.

[15] M. Duan, D. Liu, X. Ji, Y. Wu, L. Liang, X. Chen, Y. Tan, and A. Ren. Flexible clustered federated learning for client-level data distribution shift. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2661–2674, 2021.

[16] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[17] A. Fallah, A. Mokhtari, and A. Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 2020.

[18] W. Fang, D.-J. Han, L. Yuan, S. Hosseinalipour, and C. G. Brinton. Federated sketching lora: On-device collaborative fine-tuning of large language models. *arXiv preprint arXiv:2501.19389*, 2025.

[19] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 222(594-604):309–368, 1922.

[20] J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin. Linear mode connectivity and the lottery ticket hypothesis. *International Conference on Machine Learning*, 2020.

[21] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in Neural Information Processing Systems*, 2018.

[22] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 2020.

[23] P. Guo, S. Zeng, Y. Wang, H. Fan, F. Wang, and L. Qu. Selective aggregation for low-rank adaptation in federated learning. *International Conference on Learning Representations*, 2025.

[24] F. Hanzely and P. Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

[25] J. Hao, Y. Wu, A. Payani, M. Lee, and M. Liu. Personalized federated fine-tuning for heterogeneous data: An automatic rank learning approach via two-level lora. *arXiv preprint arXiv:2503.03920*, 2025.

[26] S. Hayou, N. Ghosh, and B. Yu. Lora+: Efficient low rank adaptation of large models. *International Conference on Machine Learning*, 2024.

[27] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 2022.

[28] Z. Hu, L. Wang, Y. Lan, W. Xu, E.-P. Lim, L. Bing, X. Xu, S. Poria, and R. K.-W. Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*, 2023.

[29] G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, and A. Farhadi. Editing models with task arithmetic. *International Conference on Learning Representations*, 2023.

[30] X. Jin, X. Ren, D. Preotiuc-Pietro, and P. Cheng. Dataless knowledge fusion by merging weights of language models. *International Conference on Learning Representations*, 2023.

[31] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[32] S. Lee, J. Liu, Q. Wang, J. Wang, X. Cai, and Y. Wu. Dynamic fisher-weighted model merging via bayesian optimization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4923–4935, 2025.

[33] T. Li, S. Hu, A. Beirami, and V. Smith. Ditto: Fair and robust federated learning through personalization. *International Conference on Machine Learning*, 2021.

[34] D. J. MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[35] M. S. Matena and C. A. Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 2022.

[36] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[37] V. Nagarajan and J. Z. Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 2019.

[38] Y. Nesterov and B. T. Polyak. Cubic regularization of newton method and its global performance. *Mathematical programming*, 108(1):177–205, 2006.

[39] B. Neyshabur, H. Sedghi, and C. Zhang. What is being transferred in transfer learning? *Advances in Neural Information Processing Systems*, 2020.

[40] R. Pascanu and Y. Bengio. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013.

[41] Z. Peng, N. Zou, J. Zeng, G. Li, K. Chen, B. Li, and T. Wang. Hilora: Hierarchical low-rank adaptation for personalized federated learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[42] K. Pillutla, K. Malik, A.-R. Mohamed, M. Rabbat, M. Sanjabi, and L. Xiao. Federated learning with partial model personalization. *International Conference on Machine Learning*, 2022.

[43] B. Qi, H. Sun, W. Long, R. Zhao, X. Gao, et al. Cabs: Conflict-aware and balanced sparsification for enhancing model merging. *International Conference on Learning Representations*, 2025.

[44] J. Qi, Z. Luan, S. Huang, C. Fung, H. Yang, and D. Qian. FDLoRA: Personalized federated learning of large language model via dual lora tuning. *arXiv preprint arXiv:2406.07925*, 2024.

[45] H. Ritter, A. Botev, and D. Barber. A scalable laplace approximation for neural networks. In *International conference on learning representations*, 2018.

[46] M. Seo, T. Kim, H. Lee, J. Choi, and T. Tuytelaars. Co-lora: Collaborative model personalization on heterogeneous multi-modal clients. *International Conference on Learning Representations*, 2026.

[47] A. Shamsian, A. Navon, E. Fetaya, and G. Chechik. Personalized federated learning using hypernetworks. *International Conference on Machine Learning*, 2021.

[48] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar. Federated multi-task learning. *Advances in Neural Information Processing Systems*, 2017.

[49] G. Stoica, P. Ramesh, B. Ecsedi, L. Choshen, and J. Hoffman. Model merging with svd to tie the knots. *International Conference on Learning Representations*, 2025.

[50] Y. Sun, Z. Li, Y. Li, and B. Ding. Improving loRA in privacy-preserving federated learning. *International Conference on Learning Representations*, 2024.

[51] D. Tam, M. Bansal, and C. Raffel. Merging by matching models in task parameter subspaces. *Transactions on Machine Learning Research*, 2024.

[52] A. Z. Tan, H. Yu, L. Cui, and Q. Yang. Towards personalized federated learning. *IEEE transactions on neural networks and learning systems*, 34(12):9587–9603, 2022.

[53] Z. Tian, Z. Zhang, Z. Yang, R. Jin, and H. Dai. Distributed learning over networks with graph-attention-based personalization. *IEEE Transactions on Signal Processing*, 71:2071–2086, 2023.

[54] Y. Wang, H. Fu, R. Kanagavelu, Q. Wei, Y. Liu, and R. S. M. Goh. An aggregation-free federated learning for tackling data heterogeneity. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[55] Z. Wang, Z. Shen, Y. He, G. Sun, H. Wang, L. Lyu, and A. Li. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *Advances in Neural Information Processing Systems*, 2024.

[56] Z. Wang, X. Xu, Y. Liu, Y. Zhang, P. Lin, S. Feng, X. Yang, D. Wang, and H. Schütze. Why do more experts fail? a theoretical analysis of model merging. *arXiv preprint arXiv:2505.21226*, 2025.

[57] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *International Conference on Machine Learning*, 2022.

[58] C. Xie, D.-A. Huang, W. Chu, D. Xu, C. Xiao, B. Li, and A. Anandkumar. Perada: Parameter-efficient federated learning personalization with generalization guarantees. *IEEE/CVF conference on computer vision and pattern recognition*, pages 23838–23848, 2024.

[59] P. Yadav, D. Tam, L. Choshen, C. A. Raffel, and M. Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 2023.

[60] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[61] F.-E. Yang, C.-Y. Wang, and Y.-C. F. Wang. Efficient model personalization in federated learning via client-specific prompt generation. *IEEE/CVF International Conference on Computer Vision*, 2023.

[62] Y. Yang, G. Long, T. Shen, J. Jiang, and M. Blumenstein. Dual-personalizing adapter for federated foundation models. *Advances in Neural Information Processing Systems*, 2024.

[63] L. Yu, B. Yu, H. Yu, F. Huang, and Y. Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *International Conference on Machine Learning*, 2024.

[64] J. Zhang, S. Vahidian, M. Kuo, C. Li, R. Zhang, T. Yu, G. Wang, and Y. Chen. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6915–6919. IEEE, 2024.

[65] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

[66] Z. Zhou, Y. Yang, X. Yang, J. Yan, and W. Hu. Going beyond linear mode connectivity: The layerwise linear feature connectivity. *Advances in Neural Information Processing Systems*, 2023.

[67] Y. Zou, Z. Wang, Y. Shi, and Y. Zhou. Joint rank optimization and bandwidth allocation for heterogeneous federated lora fine-tuning. *IEEE Transactions on Vehicular Technology*, 2025.

Appendix

A Theoretical Results **15**

A.1 Derivation of Gaussian Distribution Approximation 15

A.2 Proof of Theorem 3.6 16

A.3 Proof of Lemma 3.7 17

A.4 Proof of Lemmas 18

B Merging Procedure **19**

C Uplink Communication Cost Comparison **20**

D Computation Cost Comparison **20**

E Experiments on FLAN benchmark **21**

E.1 Performance Comparison 21

E.2 FLAN Benchmark 21

F Experiments with Different Ranks **22**

G Experiments with Increased Number of Clients **23**

H Sensitivity under Different Mini-batch Sizes **23**

I Experiment on DomainNet Benchmark **23**

J Empirical Verification of the LMC Assumption **24**

K Discussion on Gaussian Approximation Assumption **24**

L Implementation Details **26**

L.1 Details on Hyperparameters 26

L.2 Implementation on Baselines 26

M Difference between Fisher Merging and POTARA **26**

A Theoretical Results

A.1 Derivation of Gaussian Distribution Approximation

The gaussian distribution approximation has been used in prior works [35, 51], but the full derivation is omitted. For completeness, we provide the full derivation.

The training of the individual task vector for task i can be formulated as a maximum a posteriori (MAP) estimation problem [35, 56] as follows

$$\boldsymbol{\mu}_i \;=\; \arg\max_{\boldsymbol{\theta}_i} p(\boldsymbol{\theta}_i \,|\, \mathcal{D}_i), \tag{18}$$

where $\mathcal{D}_i$ denotes the dataset associated with task i. Here, task i may correspond either to the FedIT objective or to the local fine-tuning objective.

We perform a second-order Taylor expansion of the log-posterior around a point $\hat{\boldsymbol{\theta}}$:

$$\log p(\boldsymbol{\theta}_i \,|\, \mathcal{D}_i) \approx \log p(\hat{\boldsymbol{\theta}} \,|\, \mathcal{D}_i) + (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}})^\top \nabla_{\boldsymbol{\theta}_i} \log p(\boldsymbol{\theta}_i \,|\, \mathcal{D}_i)|_{\boldsymbol{\theta}_i = \hat{\boldsymbol{\theta}}} - \frac{1}{2}(\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}})^\top \mathbf{H}_i (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}), \tag{19}$$

where $\mathbf{H}_i = -\nabla^2_{\boldsymbol{\theta}_i} \log p(\boldsymbol{\theta}_i \,|\, \mathcal{D}_i)|_{\boldsymbol{\theta}_i = \hat{\boldsymbol{\theta}}}$.

We assume that the parameter $\boldsymbol{\mu}_i$ of the trained neural network is a local maximum of the posterior distribution. By setting $\hat{\boldsymbol{\theta}} = \boldsymbol{\mu}_i$, we have $\nabla_{\boldsymbol{\theta}_i} \log p(\boldsymbol{\theta}_i \,|\, \mathcal{D}_i)|_{\boldsymbol{\theta}_i = \boldsymbol{\mu}_i} = 0$.

Then, we obtain

$$\log p(\boldsymbol{\theta}_i \,|\, \mathcal{D}_i) \approx \log p(\boldsymbol{\mu}_i \,|\, \mathcal{D}_i) - \frac{1}{2}(\boldsymbol{\theta}_i - \boldsymbol{\mu}_i)^\top \mathbf{H}_i (\boldsymbol{\theta}_i - \boldsymbol{\mu}_i), \tag{20}$$

where $\mathbf{H}_i = -\nabla^2_{\boldsymbol{\theta}_i} \log p(\boldsymbol{\theta}_i \,|\, \mathcal{D}_i)|_{\boldsymbol{\theta}_i = \boldsymbol{\mu}_i}$.

Applying the Laplace approximation [34, 11] around the local optimum $\boldsymbol{\mu}_i$, the posterior distribution $p(\boldsymbol{\theta}_i \,|\, \mathcal{D}_i)$ admits the following Gaussian approximation:

$$p(\boldsymbol{\theta}_i \,|\, \mathcal{D}_i) \approx \mathcal{N}(\boldsymbol{\theta}_i; \boldsymbol{\mu}_i, \mathbf{H}_i^{-1}) \tag{21}$$

Equivalently, we write

$$\boldsymbol{\theta}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad \text{where } \boldsymbol{\Sigma}_i = \mathbf{H}_i^{-1}. \tag{22}$$

Given a trained neural network model $p_{\boldsymbol{\mu}_i}(\boldsymbol{y} \mid \boldsymbol{x})$, the Fisher information matrix [35, 19] is defined as

$$\mathbf{F}_{\boldsymbol{\mu}_i} = \mathbb{E}_{\boldsymbol{x}} \Big[\mathbb{E}_{\boldsymbol{y} \sim p_{\boldsymbol{\mu}_i}(\boldsymbol{y}|\boldsymbol{x})} \big[\nabla_{\boldsymbol{\mu}_i} \log p_{\boldsymbol{\mu}_i}(\boldsymbol{y} \mid \boldsymbol{x}) \nabla_{\boldsymbol{\mu}_i} \log p_{\boldsymbol{\mu}_i}(\boldsymbol{y} \mid \boldsymbol{x})^\top \big] \Big]. \tag{23}$$

It can be shown that the Fisher information matrix coincides with the Hessian at modes of the distribution [40].

A.2 Proof of Theorem 3.6

Recall that the joint distribution of $(\boldsymbol{\theta}_{\mathrm{FedIT}}, \boldsymbol{\theta}_{\mathrm{Local}})$ is given by

$$\begin{bmatrix}\boldsymbol{\theta}_{\mathrm{FedIT}} \\ \boldsymbol{\theta}_{\mathrm{Local}}\end{bmatrix} \sim \mathcal{N}\Bigg(\begin{bmatrix}\boldsymbol{\mu}_{\mathrm{FedIT}} \\ \boldsymbol{\mu}_{\mathrm{Local}}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{\mathrm{FedIT}} & \boldsymbol{\Sigma}_{\mathrm{Cross}} \\ \boldsymbol{\Sigma}_{\mathrm{Cross}}^{\top} & \boldsymbol{\Sigma}_{\mathrm{Local}}\end{bmatrix} \Bigg). \tag{24}$$

where $\boldsymbol{\Sigma}_{\mathrm{Cross}}$ denotes the cross-covariance between the two task vectors with $\boldsymbol{\Sigma}_{\mathrm{Cross}} \succeq \mathbf{0}$, $\boldsymbol{\Sigma}_{\mathrm{Cross}} \preceq \boldsymbol{\Sigma}_{\mathrm{FedIT}}$ and $\boldsymbol{\Sigma}_{\mathrm{Cross}} \preceq \boldsymbol{\Sigma}_{\mathrm{Local}}$.

Then, the merged task vector follows the following Gaussian distribution:

$$\boldsymbol{\theta}_{\mathrm{Merge}} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathrm{Merge}}, \boldsymbol{\Sigma}_{\mathrm{Merge}}) \tag{25}$$

where

$$\boldsymbol{\mu}_{\mathrm{merge}} = \lambda_{\mathrm{FedIT}}\boldsymbol{\mu}_{\mathrm{FedIT}} + \lambda_{\mathrm{Local}}\boldsymbol{\mu}_{\mathrm{Local}}, \tag{26}$$

$$\boldsymbol{\Sigma}_{\mathrm{merge}} = \lambda_{\mathrm{FedIT}}^2\boldsymbol{\Sigma}_{\mathrm{FedIT}} + \lambda_{\mathrm{Local}}^2\boldsymbol{\Sigma}_{\mathrm{Local}} + \lambda_{\mathrm{FedIT}}\lambda_{\mathrm{Local}}(\boldsymbol{\Sigma}_{\mathrm{Cross}} + \boldsymbol{\Sigma}_{\mathrm{Cross}}^{\top}), \tag{27}$$

$$\lambda_{\mathrm{FedIT}}, \lambda_{\mathrm{Local}} \geq 0, \tag{28}$$

$$\lambda_{\mathrm{FedIT}} + \lambda_{\mathrm{Local}} = 1. \tag{29}$$

To analyze the effect of model merging, we expand the task loss $\mathcal{L}$ around its optimum $\mathbf{W}^*$. Let $\boldsymbol{\phi} = \mathbf{W}_{\mathrm{Pre}} + \boldsymbol{\theta}_{\mathrm{Merge}} - \mathbf{W}^*$ and $\mathbf{H} = \nabla^2\mathcal{L}(\mathbf{W}^*)$, where $\mathbf{W}_{\mathrm{Pre}}$ is the pre-trained model that can be regarded as a constant vector. By Assumption 3.4 and Lemma 2.2 in [4], we have

$$\mathcal{L}(\mathbf{W}_{\mathrm{Pre}} + \boldsymbol{\theta}_{\mathrm{Merge}}) \leq \mathcal{L}(\mathbf{W}^*) + \frac{1}{2}\,\boldsymbol{\phi}^{\top}\mathbf{H}\,\boldsymbol{\phi} + \frac{L_H}{6}\|\boldsymbol{\phi}\|_2^3. \tag{30}$$

By taking expectation, we obtain

$$\mathbb{E}\big[\mathcal{L}(\mathbf{W}_{\mathrm{Pre}} + \boldsymbol{\theta}_{\mathrm{Merge}}) - \mathcal{L}(\mathbf{W}^*)\big] \leq \frac{1}{2}\,\mathbb{E}\big[\boldsymbol{\phi}^{\top}\mathbf{H}\boldsymbol{\phi}\big] + \frac{L_H}{6}\mathbb{E}[\|\boldsymbol{\phi}\|_2^3]. \tag{31}$$

Using Lemma A.1 (i.e., $\mathbb{E}[\boldsymbol{\phi}^{\top}\mathbf{H}\boldsymbol{\phi}] = (\mathbb{E}\boldsymbol{\phi})^{\top}\mathbf{H}(\mathbb{E}\boldsymbol{\phi}) + \mathrm{tr}(\mathbf{H}\,\mathrm{Cov}(\boldsymbol{\phi}))$), we obtain

$$\begin{aligned}
&\mathbb{E}\big[\mathcal{L}(\mathbf{W}_{\mathrm{Pre}} + \boldsymbol{\theta}_{\mathrm{Merge}}) - \mathcal{L}(\mathbf{W}^*)\big] \\
&\leq \frac{1}{2}\,\mathbb{E}\big[\boldsymbol{\phi}^{\top}\mathbf{H}\boldsymbol{\phi}\big] + \frac{L_H}{6}\mathbb{E}[\|\boldsymbol{\phi}\|_2^3] \\
&= \frac{1}{2}(\mathbb{E}\boldsymbol{\phi})^{\top}\mathbf{H}(\mathbb{E}\boldsymbol{\phi}) + \frac{1}{2}\mathrm{tr}(\mathbf{H}\,\mathrm{Cov}(\boldsymbol{\phi})) + \frac{L_H}{6}\mathbb{E}[\|\boldsymbol{\phi}\|_2^3] \\
&= \frac{1}{2}(\mathbf{W}_{\mathrm{Pre}} + \boldsymbol{\mu}_{\mathrm{Merge}} - \mathbf{W}^*)^{\top}\mathbf{H}(\mathbf{W}_{\mathrm{Pre}} + \boldsymbol{\mu}_{\mathrm{Merge}} - \mathbf{W}^*) + \frac{1}{2}\mathrm{tr}(\mathbf{H}\,\boldsymbol{\Sigma}_{\mathrm{Merge}}) + \frac{L_H}{6}\mathbb{E}\big[\|\boldsymbol{\phi}\|_2^3\big],
\end{aligned} \tag{32}$$

where $\boldsymbol{\mu}_{\mathrm{Merge}} = \mathbb{E}[\boldsymbol{\theta}_{\mathrm{Merge}}]$ and $\boldsymbol{\Sigma}_{\mathrm{Merge}} = \mathrm{Cov}(\boldsymbol{\theta}_{\mathrm{Merge}})$.

By Assumption 3.2, we have $\|\boldsymbol{\phi}\|_2 \leq \delta$, and therefore $\|\boldsymbol{\phi}\|_2^3 \leq \delta\,\|\boldsymbol{\phi}\|_2^2$. Taking expectation yields $\mathbb{E}\|\boldsymbol{\phi}\|_2^3 \leq \delta\,\mathbb{E}\|\boldsymbol{\phi}\|_2^2$. By the second-moment decomposition, we have

$$\mathbb{E}\|\boldsymbol{\phi}\|_2^2 = \|\mathbb{E}\boldsymbol{\phi}\|_2^2 + \mathrm{tr}(\mathrm{Cov}(\boldsymbol{\phi})) = \|\mathbf{W}_{\mathrm{Pre}} + \boldsymbol{\mu}_{\mathrm{Merge}} - \mathbf{W}^*\|_2^2 + \mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}}). \tag{33}$$

Combining (32)–(33), we conclude that

$$\begin{aligned}
&\mathbb{E}\big[\mathcal{L}(\mathbf{W}_{\mathrm{Pre}} + \boldsymbol{\theta}_{\mathrm{Merge}}) - \mathcal{L}(\mathbf{W}^*)\big] \\
&\leq \frac{1}{2}(\mathbf{W}_{\mathrm{Pre}} + \boldsymbol{\mu}_{\mathrm{Merge}} - \mathbf{W}^*)^{\top}\mathbf{H}(\mathbf{W}_{\mathrm{Pre}} + \boldsymbol{\mu}_{\mathrm{Merge}} - \mathbf{W}^*) + \frac{1}{2}\mathrm{tr}(\mathbf{H}\,\boldsymbol{\Sigma}_{\mathrm{Merge}}) \\
&\quad + \frac{L_H}{6}\delta\,\Big(\|\mathbf{W}_{\mathrm{Pre}} + \boldsymbol{\mu}_{\mathrm{Merge}} - \mathbf{W}^*\|_2^2 + \mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}})\Big).
\end{aligned} \tag{34}$$

Then, we have

$$\begin{aligned}\mathbb{E}\big[\mathcal{L}(\mathbf{W}_{\mathrm{Pre}}+\boldsymbol{\theta}_{\mathrm{Merge}})-\mathcal{L}(\mathbf{W}^*)\big] &\le \frac{L}{2}\|\mathbf{W}_{\mathrm{Pre}}+\boldsymbol{\mu}_{\mathrm{Merge}}-\mathbf{W}^*\|^2+\frac{L}{2}\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}})\\ &\quad+\frac{L_H}{6}\delta\left(\|\mathbf{W}_{\mathrm{Pre}}+\boldsymbol{\mu}_{\mathrm{Merge}}-\mathbf{W}^*\|_2^2+\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}})\right)\\ &\le \frac{L}{2}\delta^2+\frac{L}{2}\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}})+\frac{L_H}{6}\delta^3+\frac{L_H}{6}\delta\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}})\\ &=\left(\frac{L}{2}+\frac{L_H}{6}\delta\right)\delta^2+\left(\frac{L}{2}+\frac{L_H}{6}\delta\right)\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}}) \qquad (35)\end{aligned}$$

where the first inequality is due to Assumption 3.3, and the second inequality is due to Assumption 3.2.

Similarly, for each individual task vector we have

$$\mathbb{E}\big[\mathcal{L}(\mathbf{W}_{\mathrm{Pre}}+\boldsymbol{\theta}_i)-\mathcal{L}(\mathbf{W}^*)\big]\le\left(\frac{L}{2}+\frac{L_H}{6}\delta\right)\delta^2+\left(\frac{L}{2}+\frac{L_H}{6}\delta\right)\mathrm{tr}(\boldsymbol{\Sigma}_i),\quad i\in\{\mathrm{FedIT},\mathrm{Local}\}. \qquad (36)$$

A.3 Proof of Lemma 3.7

Let

$$a=\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{FedIT}}),\quad b=\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Local}}),\quad c=\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Cross}}). \qquad (37)$$

Hence

$$\begin{aligned}\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}}) &= \mathrm{tr}(\lambda_{\mathrm{FedIT}}^2\boldsymbol{\Sigma}_{\mathrm{FedIT}}+\lambda_{\mathrm{Local}}^2\cdot\boldsymbol{\Sigma}_{\mathrm{Local}}+\lambda_{\mathrm{FedIT}}\lambda_{\mathrm{Local}}(\boldsymbol{\Sigma}_{\mathrm{Cross}}+\boldsymbol{\Sigma}_{\mathrm{Cross}}^\top))\\ &= \lambda_{\mathrm{FedIT}}^2\cdot a+\lambda_{\mathrm{Local}}^2\cdot b+2\lambda_{\mathrm{FedIT}}\lambda_{\mathrm{Local}}\cdot c \qquad (38)\end{aligned}$$

Using the Lemma A.2, we have

$$|c| \;=\; |\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Cross}})| \;\le\; \sqrt{\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{FedIT}})\,\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Local}})} \;=\; \sqrt{ab}. \qquad (39)$$

We define a function $g(\cdot)$ with respect to λ_{FedIT} as follows

$$g(\lambda_{\mathrm{FedIT}})=\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}})=a\lambda_{\mathrm{FedIT}}^2+b(1-\lambda_{\mathrm{FedIT}})^2+2\lambda_{\mathrm{FedIT}}(1-\lambda_{\mathrm{FedIT}})c, \qquad (40)$$

where $0\le\lambda_{\mathrm{FedIT}}\le 1$.

This quadratic form is convex in λ_{FedIT}, because $g^{''}(\lambda_{\mathrm{FedIT}})=2(a+b-2c)\ge 2(a+b-2\sqrt{ab})=2(\sqrt{a}-\sqrt{b})^2\ge 0$.

Then, we obtain the optimal λ_{FedIT} and the optimum as

$$\lambda_{\mathrm{FedIT}}^*=\frac{b-c}{a+b-2c}\in[0,1], \qquad (41)$$

$$g(\lambda_{\mathrm{FedIT}}^*)=b-\frac{(b-c)^2}{a+b-2c}=a-\frac{(a-c)^2}{a+b-2c}. \qquad (42)$$

Due to $\boldsymbol{\Sigma}_{\mathrm{Cross}}\preceq\boldsymbol{\Sigma}_{\mathrm{FedIT}}$ and $\boldsymbol{\Sigma}_{\mathrm{Cross}}\preceq\boldsymbol{\Sigma}_{\mathrm{Local}}$, it follows that $\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Cross}})\le\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{FedIT}})$ and $\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Cross}})\le\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Local}})$. Combining both inequalities yields $c\le\min(a,b)$. Therefore, the optimum is smaller than a and b, i.e., $g(\lambda_{\mathrm{FedIT}}^*)\le\min\{a,b\}$.

When $a+b=2c$, the quadratic objective becomes $g(\lambda_{\mathrm{FedIT}})=(a-b)\lambda_{\mathrm{FedIT}}+b$, so the optimum is attained at one of the two endpoints. This degenerate case is rare, since it requires a sufficiently large correlation term, which is unlikely under task heterogeneity.

So far, we have shown that, at the optimal mixing weights $\lambda_{\mathrm{FedIT}}^*$ and $\lambda_{\mathrm{Local}}^*$, $\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Merge}})$ is smaller than $\min\{\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{FedIT}}),\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{FedIT}})\}$. Therefore, the merged model admits an excess loss upper bound that is tighter than the bounds for FedIT or local fine-tuning alone.

A.4 Proof of Lemmas

Lemma A.1. *Let* $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{\phi}]$ *and* $\boldsymbol{\Sigma} = \mathrm{Cov}(\boldsymbol{\phi}) = \mathbb{E}\big[(\boldsymbol{\phi} - \boldsymbol{\mu})(\boldsymbol{\phi} - \boldsymbol{\mu})^\top\big]$. *We have*

$$\mathbb{E}[\boldsymbol{\phi}^\top \mathbf{H} \boldsymbol{\phi}] = (\mathbb{E}\boldsymbol{\phi})^\top \mathbf{H} (\mathbb{E}\boldsymbol{\phi}) + \mathrm{tr}(\mathbf{H}\,\mathrm{Cov}(\boldsymbol{\phi})) \tag{43}$$

Proof.

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\phi}^\top \mathbf{H} \boldsymbol{\phi}] &= \mathbb{E}\big[\mathrm{tr}(\boldsymbol{\phi}^\top \mathbf{H} \boldsymbol{\phi})\big] = \mathbb{E}\big[\mathrm{tr}(\mathbf{H} \boldsymbol{\phi} \boldsymbol{\phi}^\top)\big] = \mathrm{tr}\big(\mathbf{H}\,\mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]\big) \\
&= \mathrm{tr}\big(\mathbf{H}(\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\top)\big) = \mathrm{tr}(\mathbf{H}\boldsymbol{\Sigma}) + \mathrm{tr}(\mathbf{H}\boldsymbol{\mu}\boldsymbol{\mu}^\top) \\
&= \mathrm{tr}(\mathbf{H}\boldsymbol{\Sigma}) + \boldsymbol{\mu}^\top \mathbf{H} \boldsymbol{\mu}
\end{aligned} \tag{44}$$

where we used $\mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top] = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\top$ and $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$. □

Lemma A.2. *Assume* $\begin{bmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^\top & \boldsymbol{\Sigma}_2 \end{bmatrix} \succeq 0$. *Then*

$$\big|\mathrm{tr}(\boldsymbol{\Sigma}_{12})\big| \;\le\; \sqrt{\mathrm{tr}(\boldsymbol{\Sigma}_1)\,\mathrm{tr}(\boldsymbol{\Sigma}_2)} \tag{45}$$

Proof. Since $\begin{bmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^\top & \boldsymbol{\Sigma}_2 \end{bmatrix} \succeq 0$, there exists a matrix $\mathbf{R}$ with $\|\mathbf{R}\|_2 \le 1$ such that

$$\boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_1^{1/2} \mathbf{R}\, \boldsymbol{\Sigma}_2^{1/2}. \tag{46}$$

By applying trace,

$$\mathrm{tr}(\boldsymbol{\Sigma}_{12}) = \mathrm{tr}\big(\boldsymbol{\Sigma}_1^{1/2} \mathbf{R}\, \boldsymbol{\Sigma}_2^{1/2}\big). \tag{47}$$

Using the trace inequality $|\mathrm{tr}(\mathbf{A}^\top \mathbf{B})| \le \|\mathbf{A}\|_F \|\mathbf{B}\|_F$ and $\|\mathbf{R}X\|_F \le \|\mathbf{R}\|_2 \|X\|_F$, we have

$$\begin{aligned}
\big|\mathrm{tr}(\boldsymbol{\Sigma}_{12})\big| &= \big|\mathrm{tr}\big((\boldsymbol{\Sigma}_1^{1/2})^\top (\mathbf{R}\boldsymbol{\Sigma}_2^{1/2})\big)\big| \\
&\le \|\boldsymbol{\Sigma}_1^{1/2}\|_F \, \|\mathbf{R}\boldsymbol{\Sigma}_2^{1/2}\|_F \\
&\le \|\boldsymbol{\Sigma}_1^{1/2}\|_F \, \|\mathbf{R}\|_2 \, \|\boldsymbol{\Sigma}_2^{1/2}\|_F \\
&\le \|\boldsymbol{\Sigma}_1^{1/2}\|_F \, \|\boldsymbol{\Sigma}_2^{1/2}\|_F.
\end{aligned} \tag{48}$$

Note that $\|\boldsymbol{\Sigma}_1^{1/2}\|_F^2 = \mathrm{tr}((\boldsymbol{\Sigma}_1^{1/2})^\top \boldsymbol{\Sigma}_1^{1/2}) = \mathrm{tr}(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_1^{1/2}) = \mathrm{tr}(\boldsymbol{\Sigma}_1)$. Similarly, $\|\boldsymbol{\Sigma}_2^{1/2}\|_F^2 = \mathrm{tr}(\boldsymbol{\Sigma}_2)$. Therefore,

$$\big|\mathrm{tr}(\boldsymbol{\Sigma}_{12})\big| \le \sqrt{\mathrm{tr}(\boldsymbol{\Sigma}_1)\,\mathrm{tr}(\boldsymbol{\Sigma}_2)}.$$

□

B Merging Procedure

Step 1: Fisher Information Matrix Estimation.

We compute the diagonal Fisher information matrices $\hat{\mathbf{F}}_{\boldsymbol{\mu}_{\mathrm{FedIT}}}$ and $\hat{\mathbf{F}}_{\boldsymbol{\mu}_{\mathrm{Local}}}$ using (16), and let $\boldsymbol{\Sigma}_{\mathrm{FedIT}} \approx \hat{\mathbf{F}}^{-1}_{\boldsymbol{\mu}_{\mathrm{FedIT}}}$ and $\boldsymbol{\Sigma}_{\mathrm{Local}} \approx \hat{\mathbf{F}}^{-1}_{\boldsymbol{\mu}_{\mathrm{Local}}}$.

Step 2: Covariance Estimation. Let $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N}$ be a random data batch from training set of client i. For two well-trained models with parameters $\boldsymbol{\mu}_{\mathrm{FedIT}}$ and $\boldsymbol{\mu}_{\mathrm{Local}}$, the per-sample gradients are given by

$$\mathbf{g}^{(n)}_{\mathrm{FedIT}} \triangleq \nabla_{\boldsymbol{\mu}_{\mathrm{FedIT}}} \ell(\boldsymbol{\mu}_{\mathrm{FedIT}}; \mathbf{x}_n, \mathbf{y}_n), \qquad \mathbf{g}^{(n)}_{\mathrm{Local}} \triangleq \nabla_{\boldsymbol{\mu}_{\mathrm{Local}}} \ell(\boldsymbol{\mu}_{\mathrm{Local}}; \mathbf{x}_n, \mathbf{y}_n), \tag{49}$$

where ℓ is the per-sample task loss. For each parameter coordinate $k \in \{1, \ldots, d\}$ where d denotes the number of trainable parameters, compute the mean and variance of per-sample gradient, and covariance as

$$\bar{g}_{\mathrm{FedIT},k} = \frac{1}{N}\sum_{n=1}^{N} g^{(n)}_{\mathrm{FedIT},k}, \quad \bar{g}_{\mathrm{Local},k} = \frac{1}{N}\sum_{n=1}^{N} g^{(n)}_{\mathrm{Local},k}, \tag{50}$$

$$\mathrm{Var}_{\mathrm{FedIT}}(k) = \frac{1}{N}\sum_{n=1}^{N}\big(g^{(n)}_{\mathrm{FedIT},k} - \bar{g}_{\mathrm{FedIT},k}\big)^2, \quad \mathrm{Var}_{\mathrm{Local}}(k) = \frac{1}{N}\sum_{n=1}^{N}\big(g^{(n)}_{\mathrm{Local},k} - \bar{g}_{\mathrm{Local},k}\big)^2, \tag{51}$$

$$\mathrm{Cov}_{\mathrm{Cross}}(k) = \frac{1}{N}\sum_{n=1}^{N}\big(g^{(n)}_{\mathrm{FedIT},k} - \bar{g}_{\mathrm{FedIT},k}\big)\big(g^{(n)}_{\mathrm{Local},k} - \bar{g}_{\mathrm{Local},k}\big). \tag{52}$$

The gradient correlation coefficient for coordinate k is defined as

$$\rho_k \triangleq \frac{\mathrm{Cov}_{\mathrm{Cross}}(k)}{\sqrt{\mathrm{Var}_{\mathrm{FedIT}}(k)\,\mathrm{Var}_{\mathrm{Local}}(k)}}, \qquad \rho_k \in [-1, 1]. \tag{53}$$

We adopt a diagonal approximation for the cross-covariance,

$$(\boldsymbol{\Sigma}_{\mathrm{Cross}})_{kk} \triangleq \rho_k \sqrt{(\boldsymbol{\Sigma}_{\mathrm{FedIT}})_{kk}\,(\boldsymbol{\Sigma}_{\mathrm{Local}})_{kk}}, \qquad k \in \{1, \ldots, d\}. \tag{54}$$

Then, we clip ρ_k to ensure $\boldsymbol{\Sigma}_{\mathrm{Cross}} \succeq 0$ and $\boldsymbol{\Sigma}_{\mathrm{Cross}} \preceq \boldsymbol{\Sigma}_{\mathrm{FedIT}}, \boldsymbol{\Sigma}_{\mathrm{Local}}$:

$$\rho_k \triangleq \max\big\{0,\ \min\{\rho_k,\ \rho_k^{\max}\}\big\}, \quad \text{where } \rho_k^{\max} = \min\left\{\sqrt{\frac{(\boldsymbol{\Sigma}_{\mathrm{FedIT}})_{kk}}{(\boldsymbol{\Sigma}_{\mathrm{Local}})_{kk}}},\ \sqrt{\frac{(\boldsymbol{\Sigma}_{\mathrm{Local}})_{kk}}{(\boldsymbol{\Sigma}_{\mathrm{FedIT}})_{kk}}}\right\}. \tag{55}$$

This yields a PSD diagonal $\boldsymbol{\Sigma}_{\mathrm{Cross}}$ with each element bounded by the corresponding entries of $\boldsymbol{\Sigma}_{\mathrm{FedIT}}$ and $\boldsymbol{\Sigma}_{\mathrm{Local}}$.

Step 3: Mixing Weights Calculation. The scalars a, b, and c are calculated as

$$a = \mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{FedIT}}),\ b = \mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Local}}), \tag{56}$$

$$c = \mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{Cross}}) = \sum_{k=1}^{d}(\boldsymbol{\Sigma}_{\mathrm{Cross}})_{kk} = \sum_{k=1}^{d}\rho_k\sqrt{(\boldsymbol{\Sigma}_{\mathrm{FedIT}})_{kk}\,(\boldsymbol{\Sigma}_{\mathrm{Local}})_{kk}}. \tag{57}$$

C Uplink Communication Cost Comparison

We denote T as the number of communication rounds. FedIT uploads both $\mathbf{A}$ and $\mathbf{B}$ each round, yielding per-client cost $\mathcal{O}(r(m+n)T)$. FedSA uploads only $\mathbf{A}$, yielding $\mathcal{O}(rnT)$. FFA-LoRA uploads only $\mathbf{B}$, yielding $\mathcal{O}(rmT)$. The communication cost of FedDPA, FedALT, PerAda, and APFL are $\mathcal{O}(r(m+n)T)$ because all of them uploads the local LoRA module each round. FedMosaic requires each client to upload the local last-layer gradients for computing client similarity coefficients. Since this gradient aggregation is performed periodically, its communication cost is negligible compared with the cost of transmitting LoRA modules. As a result, the overall communication cost of FedMosaic is $\mathcal{O}(r(m+n)T)$. The uplink communication cost of L2GD is $\mathcal{O}(rp(m+n)(1-p)T)$, where p denotes the probability of performing an aggregation step, since uplink transmission occurs only when the coin toss switches from a local update step to an aggregation step. For POTARA, clients upload both $\mathbf{A}$ and $\mathbf{B}$ only during the FedIT phase, so the cost is $\mathcal{O}(r(m+n)T_{\text{FedIT}})$, where T_{FedIT} denotes the number of communication rounds required to train the FedIT model. The communication cost comparison is summarized in Table 5. POTARA achieves consistent performance gains even under a reduced communication budget. A detailed performance-communication trade-off is reported in Section 4.3.

Table 5: Communication Cost Comparison.

Methods	**Communication Overhead**
FedIT	$\mathcal{O}(r(m+n)T)$
FedSA	$\mathcal{O}(rnT)$
FFA-LoRA	$\mathcal{O}(rmT)$
FedALT	$\mathcal{O}(r(m+n)T)$
FedDPA	$\mathcal{O}(r(m+n)T)$
L2GD	$\mathcal{O}(rp(m+n)(1-p)T)$
FedMosaic	$\mathcal{O}(r(m+n)T)$
APFL	$\mathcal{O}(r(m+n)T)$
PerAda	$\mathcal{O}(r(m+n)T)$
POTARA	$\mathcal{O}(r(m+n)T_{\text{FedIT}})$

D Computation Cost Comparison

We compare the proposed method with baselines in terms of computation time and accuracy. The results are reported in Table 4. All the experiments are conducted on NVIDIA A100 GPU with 40 GB of memory. FedDPA, FedMosaic, APFL, and PerAda are dual-adapter based methods. Among them, FedDPA, APFL, and PerAda update both the global adapter and the local adapter during training, which substantially increases the training cost. As a result, these methods require nearly twice the computation time of FedIT. Although FedMosaic mainly updates the local adapter with frozen global adapter, it is still more time-consuming than FedIT because it additionally computes gradient-based similarity coefficients and updates the gating parameter during training. FedALT also needs to update the local adapter and the mixer weight while keeping the global adapter frozen, which incurs additional cost than FedIT. In contrast, POTARA achieves the lowest total computation time among all compared methods and requires only 3.91 hours in total. This is significantly lower than FedIT, which requires 5.92 hours, and much lower than FedDPA, APFL, and PerAda, whose total computation times are around 11 hours. The additional estimation steps in POTARA introduce only a small computational overhead, while avoiding expensive dual-adapter federated training. More importantly, the computational efficiency of POTARA does not come at the cost of performance. POTARA achieves the highest accuracy of 74.18, outperforming baselines. These results demonstrate that POTARA provides a more favorable efficiency-accuracy trade-off than baselines. Overall, the results show that the proposed post-training merging strategy is computationally efficient and effective.

We further compares POTARA with FedIT under the same total training cost in Table 6. On CIFAR-100, POTARA uses the FedIT checkpoint trained for 300 communication rounds and merges it with the local model trained for 300 rounds, resulting in a total training cost of 600 rounds. Under the same training cost, FedIT trained alone for 600 rounds achieves 61.28 accuracy, whereas POTARA

achieves a higher accuracy of 62.98 while requiring only 300 communication rounds. A similar trend is observed on the commonsense reasoning benchmark. POTARA can terminate federated training earlier at 100 rounds and merge the resulting FedIT model with the local model trained for 300 rounds, yielding a total training cost of 400 rounds and an accuracy of 74.10. In contrast, FedIT trained alone for 400 rounds achieves only 71.60 accuracy. These results show that, under the same total training cost, POTARA achieves better performance than FedIT while substantially reducing communication overhead.

Table 6: Comparison between POTARA and FedIT under the same total training cost. For POTARA, the total training cost includes federated training and local training, while communication cost only comes from federated training.

Benchmark	Method	Training Cost	Communication Cost	Accuracy
CIFAR-100	FedIT	600 rounds	600 rounds	61.28
	POTARA	300 FedIT + 300 local	300 rounds	**62.98**
Commonsense Reasoning	FedIT	400 rounds	400 rounds	71.60
	POTARA	100 FedIT + 300 local	100 rounds	**74.10**

E Experiments on FLAN benchmark

E.1 Performance Comparison

We evaluate POTARA on the FLAN benchmark [8], which consists of over 60 datasets covering more than 12 NLP task categories. To simulate an extreme heterogeneity scenario, we consider eight clients, each trained on a completely different dataset. This setting is highly task-heterogeneous and is substantially more extreme than the CIFAR-100 and commonsense reasoning setups. Dataset details, task descriptions, and client assignments are summarized in Appendix E.2. We adopt Qwen3-4B-Instruct-2507 [60] as the pre-trained model and report ROUGE-1 for all tasks. Baselines are trained for 600 rounds, except local fine-tuning which runs for 300 rounds to mitigate overfitting. For POTARA, we train FedIT for 300 rounds and local fine-tuning for 300 rounds, and merge the two resulting models to obtain the final model. As shown in Table 7, POTARA achieves the best overall ROUGE-1 on the FLAN benchmark and consistently performing strongly across diverse task categories, indicating that merging global knowledge with personalized knowledge is especially effective under extreme task heterogeneity. Meanwhile, POTARA attains these improvements while reducing communication cost because it merges an intermediate FedIT model with local model.

Table 7: ROUGE-1 performance on FLAN benchmark (Qwen3-4B-Instruct model).

FLAN benchmark (Qwen3-4B-Instruct model)									
Methods	**Closed-book QA**	**Reading Comp.**	**Coreference**	**Paraphrase**	**Commonsense**	**NLI**	**Reading**	**Sentiment**	**Avg.**
Local	69.53 ±0.75	69.38 ±2.81	88.01 ±0.70	80.00 ±3.56	87.62 ±0.78	85.78 ±2.18	73.87 ±0.66	76.16 ±1.69	78.79 ±0.99
FFA-LoRA	71.00 ±0.66	70.54 ±0.66	77.18 ±1.34	79.89 ±0.57	85.65 ±1.18	86.33 ±1.18	73.57 ±0.39	77.06 ±0.74	77.65 ±0.70
FedSA	69.54 ±0.95	69.54 ±1.09	87.78 ±2.61	81.89 ±0.87	84.05 ±1.16	86.00 ±1.19	72.72 ±0.65	76.47 ±0.98	78.50 ±0.25
FedIT	69.96 ±0.57	71.37 ±0.63	83.19 ±0.51	80.44 ±1.40	85.37 ±0.96	86.33 ±1.25	73.78 ±0.85	76.68 ±0.56	78.39 ±0.63
FedDPA	70.03 ±0.19	70.57 ±0.34	81.89 ±1.37	81.44 ±0.79	86.93 ±1.71	87.55 ±0.88	74.01 ±0.02	76.45 ±0.88	78.61 ±0.29
FedALT	68.07 ±1.51	72.50 ±0.77	84.84 ±0.51	82.44 ±1.03	85.17 ±0.07	86.78 ±0.57	73.43 ±0.86	77.40 ±1.02	78.83 ±0.26
L2GD	70.45±0.48	69.80±1.63	83.24±1.73	79.83±1.50	84.29±1.58	86.34±1.34	72.13±0.62	76.39±0.12	77.81±0.42
FedMosaic	69.33±0.94	70.39±1.18	77.79±2.67	81.00±2.00	85.56±2.81	82.83±2.50	72.90±1.14	76.20±0.01	77.01±1.65
APFL	69.55±1.02	69.79±2.55	82.44±2.18	77.84±0.84	84.74±1.65	85.50±0.83	73.17±0.82	75.86±0.30	77.37±0.45
PerAda	70.08±0.69	70.28±0.93	87.77±1.14	80.67±1.00	87.00±2.06	85.00±1.67	73.24±1.03	75.85±0.05	78.74±0.14
POTARA	70.61 ±0.54	70.61 ±2.06	88.26 ±1.77	81.56 ±0.42	88.80 ±0.23	86.78 ±1.10	75.18 ±0.78	76.10 ±1.01	79.74 ±0.27

E.2 FLAN Benchmark

We follow the experimental setup in [3, 62] to simulate a client-level task heterogeneity setting using the FLAN datasets [8]. The FLAN collection comprises over 12 natural language processing tasks, each of which contains multiple datasets. These tasks are generated from diverse contextual factors and inherently exhibit complex and heterogeneous data distribution shifts. We consider an extreme client heterogeneity setting where 8 clients are assigned completely different NLP tasks. To simulate a realistic environment where each client has limited access to large-scale datasets, we adopt a downsampling strategy that assigns 2000 training samples and 300 test samples to each client. The specific task and dataset assignments are summarized in Table 8.

Table 8: FLAN benchmark.

Task	Dataset	Description
Closed-book QA	arc_challenge	Answer multiple-choice questions.
Reading Comp. w/ Commonsense	cosmos_qa	Reading comprehension with commonsense.
Coreference Resolution	definite_pronoun_resolution	Resolve ambiguous pronouns.
Paraphrase Detection	glue_qqp	Identify semantic equivalence.
Commonsense Reasoning	hellaswag	Choose the most plausible continuation.
Natural Language Inference	mnli	Infer relations between sentence pairs.
Reading Comprehension	squad_v1	Extract answers from passages.
Sentiment Analysis	sst2	Sentence-level sentiment classification.

F Experiments with Different Ranks

We evaluate the proposed method under different LoRA ranks and report the results on personalized CIFAR-100 and commonsense reasoning benchmarks in Tables 9 and 10, respectively. As shown in Table 9, increasing the LoRA rank generally improves the performance of all methods. Note that our method consistently achieves the best performance across all ranks. In particular, on personalized CIFAR-100, our method obtains accuracies of 58.26, 65.21, and 72.66 for ranks 4, 8, and 16, respectively, outperforming the strongest baseline at each rank. Table 10 shows the results on the commonsense reasoning benchmark. Our method again achieves the highest average accuracy for all ranks, with accuracies of 74.18, 73.68, and 75.55 for ranks 8, 16, and 32, respectively. These results indicate that the proposed method is not sensitive to a specific rank choice and remains effective under different parameter-efficient adaptation budgets. Overall, the consistent improvements across both vision and language benchmarks demonstrate the adaptability and robustness of our method.

Table 9: Performance comparison on Personalized CIFAR-100 benchmark with different ranks.

Personalized CIFAR-100 benchmark

Rank	Local	FFA-LoRA	FedSA	FedIT	FedDPA	Fisher Merging	FedMosaic	PerAda	L2GD	APFL	Potara
4	55.24	33.77	55.22	55.28	55.26	55.19	55.22	54.34	51.14	57.64	58.26
8	60.43	41.52	60.87	61.28	61.80	61.72	61.53	60.64	53.43	61.91	65.21
16	63.05	51.62	66.67	68.27	68.66	69.95	67.75	67.83	66.51	69.96	72.66

Table 10: Performance comparison on commonsense reasoning benchmark with different ranks.

Commonsense reasoning benchmark

Rank	Local	FFA-LoRA	FedSA	FedIT	FedDPA	FedMosaic	PerAda	L2GD	APFL	Potara
8	72.05	68.80	71.15	71.94	72.33	71.14	71.95	72.63	72.30	74.18
16	72.20	70.75	72.00	72.35	72.25	71.85	72.47	72.13	72.00	73.68
32	72.33	71.12	72.78	72.27	72.35	71.63	72.10	72.63	71.67	75.55

G Experiments with Increased Number of Clients

In this section, we evaluate the scalability of POTARA by increasing the number of clients across different benchmarks. For personalized CIFAR-100, we first increase the number of clients to 20, with each client following a distinct label distribution. For commonsense reasoning, we increase the number of clients to 16, where every two clients share the same task but are assigned different training samples. As shown in Tables 11 and 12, POTARA consistently outperforms all baselines under these larger-client settings, demonstrating robust performance gains as the federation scale increases. To further examine scalability, we increase the number of clients to 50 on personalized CIFAR-100. The results in Table 13 show that POTARA continues to maintain its advantage in average performance, further validating its effectiveness in larger-scale federated settings.

Table 11: Performance comparison on personalized CIFAR-100 benchmark with 20 clients.

Client	Local	FFA-LoRA	FedSA	FedIT	FedDPA	Fisher Merging	FedMosaic	L2GD	APFL	PerAda	POTARA
Client 1	63.53 ±2.22	44.37 ±1.97	61.28 ±2.27	59.52 ±0.73	67.41 ±2.92	62.00 ±2.46	64.62 ±2.60	60.97 ±4.96	62.24±0.58	57.95 ±2.30	62.39 ±3.91
Client 2	62.28 ±5.39	49.25 ±6.36	61.16 ±5.72	61.35 ±6.12	65.95 ±6.74	61.39 ±9.09	65.04 ±5.84	63.22 ±4.65	64.62±6.26	63.70 ±6.38	70.18 ±5.12
Client 3	60.40 ±0.23	45.24 ±1.43	58.25 ±2.07	64.25 ±2.42	65.39 ±3.34	59.64 ±2.42	65.44 ±1.71	64.39 ±2.12	64.33±0.60	63.91 ±2.28	70.09 ±1.34
Client 4	58.45 ±2.62	39.74 ±0.34	57.29 ±1.19	56.98 ±2.56	59.52 ±0.55	62.54 ±0.71	57.63 ±1.91	59.14 ±0.93	58.55±1.69	60.39 ±1.43	63.63 ±1.26
Client 5	62.07 ±2.18	50.88 ±4.72	62.73 ±2.31	66.00 ±3.74	68.64 ±1.89	66.22 ±0.01	66.05 ±1.93	65.06 ±2.53	65.22±0.59	61.74 ±2.11	67.94 ±4.36
Client 6	64.14 ±0.67	47.09 ±2.28	61.07 ±0.85	65.06 ±2.94	65.34 ±2.65	59.56 ±0.25	64.63 ±1.55	63.00 ±3.19	62.04±0.70	58.93 ±3.96	67.00 ±0.09
Client 7	58.82 ±0.95	40.24 ±1.49	58.12 ±0.59	57.96 ±1.45	58.85 ±3.59	61.12 ±2.24	58.42 ±5.12	57.97 ±0.07	59.31±5.34	60.10 ±0.13	61.66 ±0.95
Client 8	62.78 ±1.61	44.65 ±0.02	59.21 ±0.43	63.37 ±3.22	64.41 ±1.21	63.31 ±0.11	61.06 ±0.65	62.07 ±0.12	64.75±2.06	61.29 ±1.14	65.35 ±2.42
Client 9	62.58 ±1.40	42.76 ±2.31	60.06 ±1.32	59.63 ±4.14	60.85 ±3.94	65.22 ±0.23	59.65 ±0.71	56.71 ±2.44	64.21±1.44	58.23 ±1.12	66.83 ±0.98
Client 10	62.14 ±1.33	44.13 ±2.60	59.58 ±4.50	62.31 ±2.31	65.35 ±1.79	62.25 ±0.61	61.45 ±4.46	59.74 ±1.48	60.25±1.79	62.54 ±1.54	66.92 ±1.45
Client 11	60.16 ±1.79	44.74 ±1.14	59.23 ±2.08	64.24 ±1.03	62.61 ±1.03	57.62 ±0.73	60.63 ±1.44	60.34 ±0.98	61.86±2.67	57.31 ±3.37	61.96 ±0.85
Client 12	57.86 ±1.87	44.80 ±2.32	55.35 ±0.39	65.66 ±1.49	60.01 ±0.94	59.58 ±2.15	63.15 ±2.97	61.04 ±1.97	59.29±3.10	58.56 ±2.80	63.40 ±1.89
Client 13	61.04 ±1.68	46.04 ±7.54	61.00 ±1.22	61.74 ±1.95	64.73 ±3.67	60.20 ±4.24	63.10 ±4.80	60.01 ±1.72	63.73±2.67	63.53 ±0.08	66.48 ±1.17
Client 14	60.64 ±2.41	43.08 ±1.47	58.89 ±4.14	60.73 ±3.29	59.39 ±8.12	64.56 ±4.91	58.33 ±5.48	58.76 ±1.16	61.27±6.05	61.54 ±1.89	63.97 ±4.32
Client 15	59.61 ±1.47	37.66 ±0.69	57.57 ±1.00	61.24 ±1.12	60.62 ±2.62	59.30 ±1.39	57.44 ±0.02	56.45 ±0.77	57.81±2.32	57.97 ±1.18	63.62 ±2.54
Client 16	57.85 ±0.45	42.66 ±0.24	57.34 ±0.59	60.27 ±3.41	58.12 ±0.53	59.75 ±2.71	61.15 ±3.39	58.37 ±1.15	59.74±2.71	59.44 ±4.75	63.39 ±1.66
Client 17	62.09 ±4.21	43.70 ±8.42	58.79 ±2.54	64.45 ±5.90	61.59 ±4.52	63.83 ±3.02	61.40 ±5.63	58.00 ±7.75	61.32±5.88	61.11 ±2.42	64.84 ±3.86
Client 18	58.45 ±2.12	43.16 ±0.08	57.26 ±0.93	59.52 ±0.82	57.10 ±1.57	59.87 ±3.55	61.37 ±1.10	60.15 ±4.22	58.72±1.61	58.47 ±3.53	64.88 ±2.82
Client 19	62.75 ±3.68	50.26 ±1.38	63.88 ±1.85	66.43 ±1.42	66.52 ±2.21	66.39 ±2.62	69.22 ±0.73	67.32 ±0.42	64.91±0.84	62.34 ±1.05	71.42 ±0.54
Client 20	58.68 ±0.28	43.53 ±2.69	60.81 ±0.45	63.67 ±3.51	62.98 ±1.16	60.39 ±2.02	63.34 ±0.40	62.07 ±0.72	64.53±0.79	59.86 ±1.29	67.26 ±0.28
Average	60.81 ±0.57	44.40 ±1.09	59.45 ±0.95	62.22 ±0.93	62.77 ±0.63	61.73 ±1.45	62.16 ±1.67	60.74 ±0.27	61.95±1.52	60.35 ±0.64	65.66 ±0.82

Table 12: Performance comparison on commonsense reasoning benchmark with 16 clients.

Client	Local	FFA-LoRA	FedSA	FedIT	FedALT	FedDPA	FedMosaic	L2GD	APFL	PerAda	POTARA
Client 1 (ARC-c)	71.80 ±0.20	73.00 ±0.80	70.60 ±0.40	73.60 ±0.00	73.50 ±0.70	74.50 ±0.90	73.70 ±0.70	75.10 ±0.70	74.50±0.10	71.40 ±1.80	74.80 ±0.20
Client 2 (ARC-c)	70.50 ±0.50	73.10 ±0.30	72.20 ±1.20	73.60 ±0.00	72.40 ±0.00	74.50 ±0.10	74.40 ±0.80	75.80 ±0.00	74.10±0.70	73.00 ±0.60	75.20 ±1.00
Client 3 (ARC-e)	85.80 ±0.20	87.50 ±0.50	86.20 ±1.20	87.30 ±0.70	85.90 ±0.70	86.90 ±0.30	87.30 ±1.70	88.00 ±0.20	87.80±0.20	87.40 ±0.60	88.50 ±0.30
Client 4 (ARC-e)	86.70 ±0.50	87.10 ±0.50	85.90 ±1.30	87.30 ±0.70	85.00 ±0.20	87.40 ±0.60	88.20 ±2.00	87.70 ±1.10	88.10±0.50	87.80 ±1.60	87.90 ±0.30
Client 5 (BoolQ)	65.20 ±3.00	67.00 ±0.60	61.60 ±5.60	64.30 ±3.50	66.70 ±0.10	66.90 ±0.10	66.80 ±1.80	66.70 ±0.30	58.50±5.70	64.30 ±0.10	65.60 ±0.60
Client 6 (BoolQ)	67.90 ±1.50	63.50 ±1.70	64.90 ±0.70	64.30 ±3.50	65.50 ±0.30	64.20 ±3.00	67.60 ±0.00	62.60 ±5.00	61.70±3.50	62.40 ±4.40	66.70 ±1.70
Client 7 (HellaSwag)	68.50 ±1.50	58.50 ±0.10	66.70 ±1.90	66.90 ±1.10	69.00 ±0.20	65.00 ±0.00	62.10 ±1.10	59.50 ±1.50	59.70±2.30	71.00 ±0.80	69.70 ±1.30
Client 8 (HellaSwag)	68.10 ±0.10	59.70 ±0.50	63.90 ±2.50	66.90 ±1.10	68.00 ±1.20	68.10 ±0.70	67.70 ±1.70	60.30 ±0.90	58.70±3.90	68.60 ±3.20	68.30 ±0.90
Client 9 (OBQA)	75.90 ±0.50	71.30 ±2.90	72.90 ±0.30	76.70 ±0.10	75.80 ±1.00	78.00 ±1.80	76.70 ±0.90	75.60 ±0.60	75.40±0.80	74.80 ±2.60	78.70 ±0.50
Client 10 (OBQA)	74.90 ±2.50	73.80 ±0.40	73.50 ±1.30	76.70 ±0.10	76.00 ±1.00	77.80 ±1.00	75.50 ±1.70	75.60 ±1.20	75.00±0.20	76.70 ±1.50	77.90 ±0.70
Client 11 (PIQA)	80.20 ±0.80	79.40 ±1.80	77.40 ±0.40	80.60 ±0.60	79.80 ±0.20	80.20 ±0.60	80.90 ±0.70	80.90 ±0.90	77.60±2.20	80.10 ±0.70	81.90 ±0.90
Client 12 (PIQA)	79.40 ±1.00	77.60 ±2.60	78.30 ±0.70	80.60 ±0.60	79.20 ±1.60	81.20 ±0.60	76.20 ±3.40	80.60 ±0.00	80.40±1.80	77.50 ±1.90	81.90 ±0.30
Client 13 (SIQA)	67.70 ±1.10	67.60 ±0.20	67.00 ±0.80	70.50 ±0.10	70.80 ±3.00	71.30 ±1.50	69.10 ±0.70	65.70 ±4.30	69.60±1.20	69.60 ±0.60	73.00 ±1.60
Client 14 (SIQA)	67.80 ±1.20	67.70 ±0.70	68.30 ±0.70	70.50 ±0.10	66.50 ±0.50	69.20 ±1.80	68.50 ±0.90	64.60 ±5.00	68.70±0.50	69.50 ±0.70	72.40 ±1.80
Client 15 (WinoGrande)	67.00 ±0.20	52.20 ±2.60	66.20 ±0.40	62.00 ±0.80	63.40 ±0.00	62.10 ±0.50	60.70 ±1.10	55.00 ±3.80	52.50±0.90	62.90 ±0.70	62.80 ±0.60
Client 16 (WinoGrande)	64.80 ±1.00	51.30 ±1.70	64.30 ±1.30	62.00 ±0.80	63.20 ±0.40	61.70 ±0.10	62.40 ±1.80	54.40 ±4.80	55.80±0.80	61.30 ±1.10	63.60 ±0.20
Average	72.64 ±0.34	69.39 ±0.38	71.25 ±0.35	72.74 ±0.24	72.55 ±0.53	73.06 ±0.23	72.37 ±0.02	70.50 ±0.94	69.88±0.38	72.40 ±0.30	74.31 ±0.58

Table 13: Performance comparison on personalized CIFAR-100 benchmark with 50 clients.

Method	Local	FFA-LoRA	FedSA	FedIT	FedDPA	Fisher Merging	FedMosaic	PerAda	L2GD	POTARA
Avg Acc	57.14	45.41	53.96	62.30	62.41	62.02	63.30	61.43	61.36	64.91

H Sensitivity under Different Mini-batch Sizes

We conduct a sensitivity analysis by varying the mini-batch size used to estimate the diagonal Fisher information and cross-covariance. As shown in the table, the performance remains stable across different mini-batch sizes, indicating that POTARA is robust to this estimation parameter.

I Experiment on DomainNet Benchmark

To evaluate Potara on the dataset with domain shift, we add experiments on DomainNet, which contains images from six distinct domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. Following prior work [41, 54, 61], we construct a 10-class subset using the most frequent categories. We consider 12 clients, with two clients assigned to each domain. To introduce additional

Table 14: Sensitivity under different mini-batch sizes.

Mini-batch size	30	40	50	60
Avg Acc	74.18	74.40	74.53	74.63

within-domain data heterogeneity, the class proportions for each client are sampled from a Dirichlet distribution with concentration parameter of 0.2. We also include HiLoRA [41] as an additional baseline. For POTARA, the FedIT and local models are trained for 200 and 100 rounds, respectively, and are subsequently merged, while all baseline methods are trained for 300 rounds to ensure a comparable overall training budget. Following setting in [41], the number of rounds per tier for HiLoRA is set to $T_{\text{root}} = 150$, T_{cluster}, and T_{leaf}. The results show that Potara consistently outperforms baselines, further demonstrating its effectiveness under heterogeneous visual domains.

Table 15: Performance comparison on DomainNet benchmark.

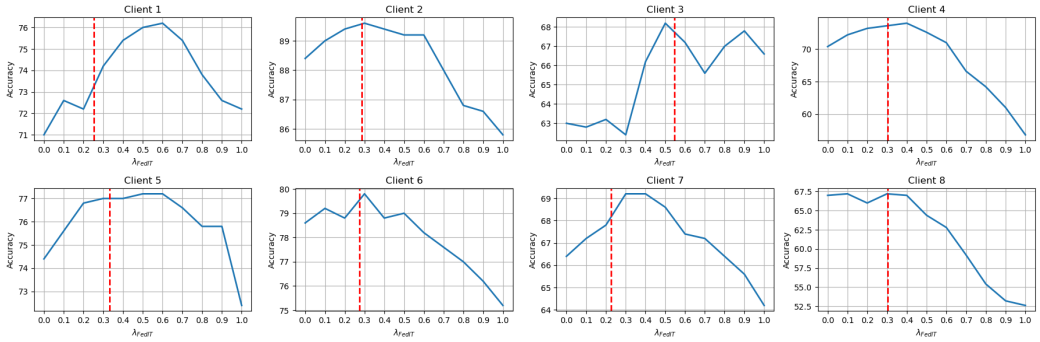
Method	FFA-LoRA	FedSA	FedIT	FedDPA	L2GD	FedMosaic	APFL	PerAda	HiLoRA	POTARA
Avg Acc	83.15 $\pm$0.78	86.62 $\pm$0.03	87.32 $\pm$0.20	85.74 $\pm$0.02	86.70 $\pm$0.20	87.52 $\pm$0.09	87.57 $\pm$0.27	86.13 $\pm$0.02	87.71 $\pm$0.45	89.00 $\pm$0.11

J Empirical Verification of the LMC Assumption

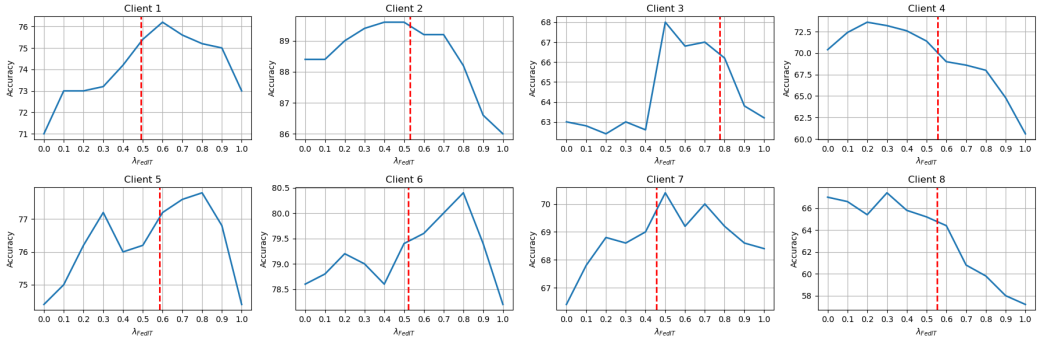
We note that our setting is fine-tuning from a shared pre-trained model, so both the FedIT and local models can be viewed as relatively small perturbations around an already strong model for related tasks. This suggests that they may remain in a common basin. To provide empirical evidence for the LMC assumption, we evaluate accuracy along the linear interpolation path. Specifically, we vary $\lambda_{\text{FedIT}} \in \{0, 0.1, \ldots, 1\}$ and evaluate client performance under different mixing weights, where $\lambda_{\text{FedIT}} = 1$ corresponds to using the FedIT model alone and $\lambda_{\text{FedIT}} = 0$ corresponds to the local model. We conduct this analysis on commonsense reasoning benchmark with LLaMA where each client has a unique and distinct dataset. Figure 4 shows that the performance of nearly all clients exhibits a clear unimodal trend with respect to λ_{FedIT}, with intermediate mixing weights outperforming both extreme settings. This observation suggests that the two models lie in a shared basin and appropriate interpolation can improve performance over either endpoint, motivating the use of model merging. Moreover, as the FedIT model is trained for more communication rounds, the peak of the unimodal curves shifts noticeably toward larger λ_{FedIT} for nearly all clients. Consistently, the mixing weights estimated by our method (indicated by red dashed lines) also move to the right as the FedIT round increases. This behavior indicates that a better-trained FedIT model contributes more reliable general knowledge to the merged model. This rightward shift further suggests that the Fisher information associated with the FedIT model increases as training proceeds, which in turn assigns a larger weight to the FedIT model during model merging. Note that the location of the best weight via grid search is sensitive to stochastic training, evaluation noise, the granularity of the search grid, and most crucially, the size of the evaluation set as well as the the degree of task heterogeneity. There is generally no uniquely identifiable or exactly optimal mixing weight. Instead, model interpolation typically yields a neighborhood of weights that achieve similarly strong performance. As shown in Figure 4, the interpolation curves are often relatively flat around their maxima, which is consistent with the linear mode connectivity and flat basin behavior commonly observed in model merging. Consequently, the single best weight identified by grid search is itself an uncertain estimate - one sample from a flat high-performing region rather than a unique optimum. Moreover, locating this high-performing region through grid search incurs substantial computational overhead. The goal of Potara is therefore not to exactly reproduce a noisy grid-search maximizer, but to efficiently identify a client-specific weight within the high-performing interpolation neighborhood. The strong empirical performance of our resulting models supports this claim and demonstrates that Potara provides a robust and computationally efficient alternative to exhaustive grid search.

K Discussion on Gaussian Approximation Assumption

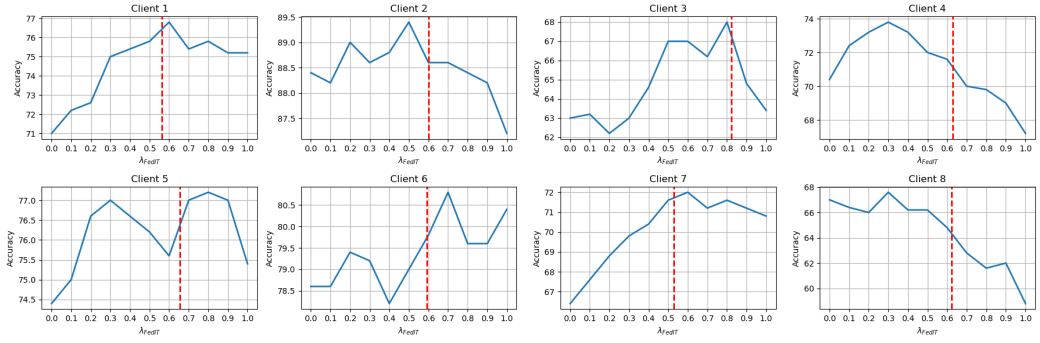
The Gaussian approximation is a standard analytical tool in Bayesian deep learning [45] and model merging [10, 35, 56]. Our analysis follows this line of work and uses the Gaussian approximation to



(a) Merging FedIT model at round 50 with local model at round 300 under varying mixing weight.



(b) Merging FedIT model at round 150 with local model at round 300 under varying mixing weight.



(c) Merging FedIT model at round 300 with local model at round 300 under varying mixing weight.

Figure 4: Client performance under various mixing weights. For each client, we construct a personalized model by merging a FedIT checkpoint at round 50,150, or 300 with a fixed local model at round 300, and sweep the mixing weight λ_{FedIT} (x-axis). Curves report the resulting client accuracy. The red dashed line marks the mixing weight calculated by our method for each client.

characterize the uncertainty of the FedIT and local task vectors. The joint Gaussian approximation is motivated by the fact that the FedIT model is obtained from aggregating client updates, which are in turn computed from local client data. As a result, the FedIT model indirectly incorporates information from the local training process, making it reasonable to model a nonzero correlation between the FedIT and local models. From this perspective, the joint Gaussian assumption is introduced to capture this dependence through the cross-covariance term.

L Implementation Details

L.1 Details on Hyperparameters

Unless stated otherwise, the hyperparameters used in this work are as follows.

Table 16: Hyperparameter setting.

Hyperparameter	ViT-B/16 & CIFAR-100	LLaMA-3.2-3B-Instruct & Commonsense Reasoning	Qwen3-4B-Instruct-2507 & FLAN
Batch size	10	4	4
Learning rate	0.001	0.0001	0.0001
LoRA dropout rate	0.1	0.05	0.05
Local iteration number	5	4	4
Target module	["query", "value"]	["q_proj", "k_proj", "v_proj"]	["q_proj", "k_proj", "v_proj"]
Number of data samples to compute mixing weights	30	30	30

L.2 Implementation on Baselines

For baseline implementation, FedMosaic is designed to handle both data heterogeneity and model heterogeneity. Since our experiments focus on the setting where all clients share the same model architecture, we implement FedMosaic with the standard LoRA layers. We implement FedDPA using its FedDPA-T variant, in which the global and local adapters are alternately updated within each communication round. For L2GD, the probability of performing an aggregation step is set to 0.5. Since Fisher Merging requires estimating Fisher information for all model parameters, it is computationally prohibitive for LLMs with over 1B parameters. Therefore, we exclude Fisher Merging from the commonsense reasoning and FLAN benchmarks.

M Difference between Fisher Merging and POTARA

Our proposed method POTARA differs from Fisher merging in two fundamental aspects.

First, these two methods operate in different parameter spaces, which leads to vastly different computational costs. Fisher merging is performed in the full parameter space of the model. When the model size is small (e.g., fewer than hundred million parameters), the computation of the Fisher information matrix is manageable and does not introduce significant overhead. However, for LLMs, whose parameter size typically exceeds one billion, computing the Fisher information for all parameters becomes computationally prohibitive. In contrast, our method performs merging in the task vector space and computes the Fisher information only for the LoRA parameters, whose dimensionality is several orders of magnitude smaller than that of the full model. As a result, our approach incurs substantially lower computational overhead and remains practical for LLMs.

Second, Fisher merging assumes independence between the models being merged and ignores their correlation structure. Specifically, Fisher merging combines models by weighting them according to their individual Fisher information matrices, without considering correlation between models. In contrast, our method explicitly models the correlation between the federated model and the local model, which allows us to derive a more principled and optimal mixing weight.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction (i.e., Section 1) accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitation is included in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The assumptions for the theoretical results are clearly stated in Section 3, and the proofs are provided in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper discloses all necessary information to reproduce the main experimental results in Section 4 and Appendix L.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The datasets are all open-source. We will release the code upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The details about the setting of our experiments are reported in Section 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: See standard deviation reported in Section 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please see Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The focus of this work is on the technical advancement of federated personalization. While this research has potential societal impacts, it primarily addresses technical challenges and does not necessitate a specific discussion on societal consequences.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All models, datasets and references are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.