Bi-level Personalization for Federated Foundation Models: A Task-vector Aggregation Approach

Yiyuan Yang[1], Guodong Long[1], Qinghua Lu[2], Liming Zhu[2], Jing Jiang[1]
[1]University of Technology Sydney, Australia
[2]CSIRO's Data61, Australia
Yiyuan.Yang-1@student.uts.edu.au, {guodong.long, jing.jiang}@uts.edu.au,
{Qinghua.Lu, Liming.Zhu}@data61.csiro.au

Abstract

Federated foundation models represent a new paradigm to jointly fine-tune pre-trained foundation models across clients. It is still a challenge to fine-tune foundation models for a small group of new users or specialized scenarios, which typically involve limited data compared to the large-scale data used in pre-training. In this context, the trade-off between personalization and federation becomes more sensitive. To tackle these, we proposed a bi-level personalization framework for federated fine-tuning on foundation models. Specifically, we conduct personalized fine-tuning on the client-level using its private data, and then conduct a personalized aggregation on the server-level using similar users measured by client-specific task vectors. Given the personalization information gained from client-level fine-tuning, the server-level personalized aggregation can gain group-wise personalization information while mitigating the disturbance of irrelevant or interest-conflict clients with non-IID data. The effectiveness of the proposed algorithm has been demonstrated by extensive experimental analysis in benchmark datasets.

Code — https://github.com/Lydia-yang/FedBip
Extended version — https://arxiv.org/abs/2509.12697

Introduction

Considering the exhaustion of publicly available data and the growing importance of data privacy, recent studies have begun to explore the adaptation of foundation models within the federated learning (FL) framework to leverage decentralized private data for collaborative learning (Zhuang, Chen, and Lyu 2023). This emerging direction, known as Federated Foundation Models (FedFM), primarily focuses on fine-tuning pre-trained foundation models in small-scale federated settings, in contrast to the massive scale of pre-training. Typically, FedFM fine-tuning involves a limited number of new users or specialized scenarios, aiming to align the general knowledge of the pre-trained model with specific local contexts. Thus, the emphasis in FedFM fine-tuning shifts toward personalization and local adaptation, posing unique challenges in designing effective and efficient methods.

To enable effective personalization in FedFM, various methods have been proposed, including the incorporation of additional personalized modules to align client-specific distribution for personalization (Chen et al. 2024; Yang et al. 2024) and methods that explicitly decouple global and personalized components to facilitate more flexible adaptation (Guo et al. 2024). While effective, these methods primarily focus on client-level personalization, relying on uniform or heuristically fixed federated aggregation at server-side for global model updates. However, in highly heterogeneous FedFM settings, such conventional federation may undermine personalization performance. Specifically, aggregating irrelevant or interest-conflict tasks among clients may dilute task-specific knowledge, while neglecting to amplify contributions from similar tasks in clients may hinder the benefit from effective knowledge sharing for group-wise personalization gaining. Furthermore, many existing approaches introduce additional personalized modules in clients, which often incur non-trivial computational and storage overhead. This presents a significant challenge in FedFM, where foundation models are typically large-scale and clients may have limited resources. Therefore, it is vital to consider the trade-off between personalization and federation for efficient and effective personalization in FedFM.

Although several aggregation-weighting strategies have been proposed in conventional FL to balance personalization and federation by adjusting aggregation weights based on client heterogeneity or similarity inferred from local model parameters (Huang et al. 2021; Zhang et al. 2023a; Rehman et al. 2023), they may be suboptimal for FedFM due to the limited parameter variation across clients, a unique characteristic of foundation model fine-tuning. Foundation models are typically pre-trained on large-scale, diverse datasets, providing a strong and generalizable initialization for various downstream tasks fine-tuning. As a result, local fine-tuning in FedFM usually involves small parameter updates, leading to highly similar model parameters across clients. This significantly limits the effectiveness of conventional FL methods that rely on parameter divergence to infer client similarity or discrepancy. Consequently, there is a pressing need to develop new methods tailored to FedFM, which can effectively balance personalization and federation under high data heterogeneity and constrained parameter variance.

To fill this gap, we propose a bi-level personalization framework for FedFM fine-tuning, named FedBip, which jointly leverages both client-level personalization and

server-level personalization to enhance model adaptation in heterogeneous settings. At the client level, FedBip fine-tunes the foundation model on each client's local dataset, enabling task-specific adaptation and personalization. At the server level, to further improve personalization, we introduce a task-vector aggregation mechanism to promote collaboration among similar tasks and mitigate interference from irrelevant or conflicting ones. Specifically, this mechanism computes pairwise task similarities between clients and assigns client-specific aggregation weights accordingly, so that each client can emphasize contributions from similar clients for group-wise knowledge sharing, while down-weighting the influence of unrelated or conflicting tasks. Additionally, to address the challenge of limited parameter variation in FedFM, we adopt task vector for aggregation inspired by prior work (Ilharco et al. 2022). The task vector is computed as the difference between the local fine-tuned model and its initialization at each communication round, serving as a compact and informative representation of the client's task for effective similarity estimation even under constrained update spaces. In this way, FedBip effectively balances personalization and federation by adaptively aligning aggregation with task similarity, enabling both individual task adaptation and group-wise knowledge transfer. Extensive experiments on benchmarks demonstrate the effectiveness of our method. Our contribution can be summarized as:

- A novel bi-level personalization framework tailored for federated foundation models, enabling personalized learning at both the client and server levels.
- An innovative server-side personalization method based on task vector aggregation, enhancing global model adaptability across heterogeneous clients.
- A new lightweight joint fine-tuning strategy for small-scale adaptation of pre-trained foundation models, improving efficiency and performance in federated settings.
- A comprehensive empirical analysis demonstrating the effectiveness and robustness of the proposed framework across diverse benchmark datasets and scenarios.

Related Work

Federated Foundation Model

Federated foundation models (Zhuang, Chen, and Lyu 2023; Yu, Muñoz, and Jannesari 2023; Ren et al. 2024) have been proposed to adapt the capabilities of large-scale pre-trained models in FL settings while preserving data privacy and enabling collaborative learning across decentralized clients. FedIT (Zhang et al. 2023b) serves as an early attempt to adapt FedAvg for large-scale foundation models. Based on this, a growing body of research has emerged to tackle various challenges in FedFM, including efficiency (Zhang et al. 2023c; Yang et al. 2025), privacy (Han et al. 2024a), and heterogeneity (Cho et al. 2023; Sun et al. 2024; Wang et al. 2024). As the diversity of client data and tasks increases, personalization becomes a main concern in FedFM. Based on the underlying personalization techniques used, existing methods can be broadly categorized into two main types: (1) methods that introduce additional personalized modules (Chen et al. 2024; Yang et al. 2024), enabling each client to adapt to its local data distribution, while still benefiting from shared global knowledge; (2) methods that explicitly decouple global and personalized components (Guo et al. 2024), allowing for more flexible and resource-efficient control over global collaboration and local adaption. However, these approaches primarily focus on client-level personalization, often overlooking the potential degradation of personalization performance from uninformed server-side aggregation strategies. To fill this gap, our work explores bi-level personalization in FedFM by jointly considering both client-level and server-level personalization, extending the scope of research in this area.

Aggregation-Weighting Methods in FL

In conventional FL, there are also various studies exploring aggregation-weighting strategies to balance local adaptation and federation by adjusting the contribution of each client during aggregation. These approaches can be broadly categorized into two classes: global model aggregation-weighting methods and personalized aggregation-weighting methods. For global model aggregation-weighting methods, the goal is to mitigate inter-client divergence by assigning different aggregation weights across clients to reduce the impact of conflicting updates, thereby enhancing global model performance. For example, some approaches, such as FedDisco (Ye et al. 2023) and L-DAWA (Rehman et al. 2023), explicitly adjust the server's aggregation weights by quantifying the divergence between the local and global model distributions or parameter spaces to mitigate inter-client divergence. Other methods like FedLAW (Li et al. 2023) and FedAWA (Shi et al. 2025) propose to learn the aggregation weights by optimizing the divergence between local client models and the global model, thereby enabling more adaptive aggregation. In contrast, personalized aggregation methods further enhance performance by employing client-specific aggregation-weighting strategies, enabling better alignment with local objectives. FedAMP (Huang et al. 2021) introduces an attention-based mechanism that enables each client to perform personalized aggregation by weighing peer models based on parameter similarity. Similarly, pFedLA (Ma et al. 2022) employs a hypernetwork to learn a layer-wise aggregation policy, assigning distinct weights across layers to better personalize the aggregated model. Methods such as FedPHP (Li et al. 2021b) and APPLE (Luo and Wu 2022) further aggregate client models locally with client-specific weights during local model updates. Additionally, approaches like PartialFed (Sun et al. 2021), and FedALA (Zhang et al. 2023a) adopt adaptive strategies to blend the local and global models, thereby generating personalized models based on the extent of divergence or task relevance. However, all these methods are based on conventional FL and do not consider the unique characteristics of FedFM, while our method is the first to leverage the capacity of pre-trained foundation models to realize the adaptive personalized aggregation-weighting for balanced personalization and federated learning in FedFM.

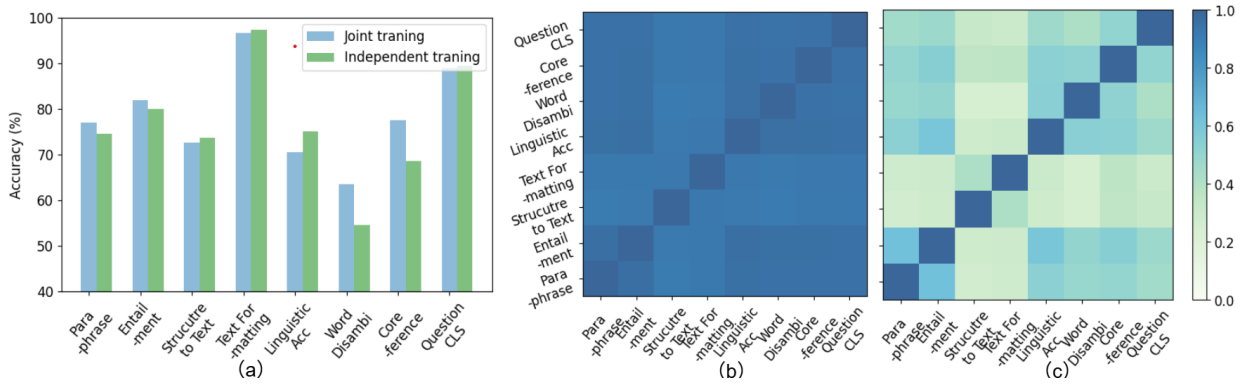

Figure 1: (a) is the performance comparison between jointly training and independently training. (b) and (c) are parameter and vector similarity between models fine-tuned on different tasks, respectively.

Task Arithmetic

Task Arithmetic (Ilharco et al. 2022) is proposed as an efficient model editing method for foundation models, enabling flexible adaptation and modification without full retraining. It introduces the concept of "task vector" as a compact representation of the transformation required to adapt a pre-trained model to a specific task through fine-tuning. By performing simple task vector arithmetic, it can create models with new capabilities for tasks. Several studies have explored the integration of task arithmetic into FL. For example, the study (Morafah et al. 2024) incorporates task arithmetic to merge knowledge distilled from heterogeneous models, while other work (Tao et al. 2024) demonstrates the effectiveness of task arithmetic from a theoretical perspective of FL. However, none of the existing works have explored the use of task arithmetic in FedFM, and our work fills this gap by introducing task-vector aggregation tailored to the unique challenges of FedFM.

Preliminary

Task Vector

Task vector is introduced by (Ilharco et al. 2022) as an efficient method for model editing, which effectively encodes the necessary information required to perform a specific task and can serve as task-specific representations. Formally, given a pre-trained model θ_{pre} and its fine-tuned model θ^i_{ft} on task i, the task vector is obtained by the element-wise difference between the pre-trained and fine-tuned model, formulated as $\tau_i = \theta^i_{ft} - \theta_{pre}$. These task vectors can be used to transfer, combine, or remove task-specific knowledge through simple vector arithmetic. For example, by adding a task vector as $\theta_{pre} + \tau_i$, one can enable fast adaptation or transfer of the pre-trained model to task i.

Task Vector in PEFT. To reduce computational overhead, parameter-efficient fine-tuning (PEFT) methods (Han et al. 2024b) have been proposed, which achieve efficient learning by updating only a small subset of model parameters while keeping the majority of parameters frozen. This can be formalized as $\theta = (\Delta\theta, \theta_{pre})$, where $\Delta\theta$ denotes the trainable subset and θ_{pre} remains fixed. Suppose the initial tunable parameters are $\Delta\theta^0$ and the fine-tuned parameters are $\Delta\theta^t$, then the task vector for task i in the PEFT setting is $\tau_i = \Delta\theta^i - \Delta\theta^0$.

Federated Foundation Models

Federated foundation models (Zhuang, Chen, and Lyu 2023) are proposed to adapt large-scale foundation models within the FL framework by leveraging private data from distributed clients for model training. Given K clients, each with its local dataset D_k, the overall process of FedFM is:

$$\text{Client: } \theta_k = \min_{\theta} f_k(\theta; D_k), \quad \text{Server: } \theta = \sum_{k=1}^{K} p_k \theta_k, \quad (1)$$

where $f_k(\theta; D_k)$ is the local objective function computed on client k's dataset D_k, and p_k is the aggregated weight assigned to client k. As foundation models usually contains millions or even billions of parameters compared to traditional models, recent studies have incorporated PEFT method in FedFM for efficient learning, where the objective is to fine-tune only a small learnable subset of parameters, formulates as $\Delta\theta_k = \min_{\Delta\theta} f_k(\Delta\theta, \theta_{pre}; D_k)$ on clients and $\Delta\theta = \sum_{k=1}^{K} p_k \Delta\theta_k$ on the server.

Motivation

As foundation models are pre-trained on large-scale and diverse data to acquire generalizable capabilities across various downstream tasks, the objective of FedFM fine-tuning is to adapt these generalized models to client-specific data in a privacy-preserving way, enabling better personalization for new users or scenarios. However, the unique characteristics of FedFM, such as high data heterogeneity and strong pre-trained initialization, introduce new challenges that go beyond those encountered in conventional FL.

Personalization and Federation in FedFM. Let $(\boldsymbol{x}, \boldsymbol{y})$ denote a data pair sampled from the distribution P and clients $i, j \in [K]$ and $i \neq j$. In conventional FL, it primarily addresses heterogeneity with label shifts $P_i(\boldsymbol{y}) \neq P_j(\boldsymbol{y})$ or feature shifts $P_i(\boldsymbol{x}) \neq P_i(\boldsymbol{x})$. In contrast, FedFM often involves more complex heterogeneity, such as task shift

or domain shift, denoted as $P_i(\boldsymbol{x}, \boldsymbol{y}) \neq P_j(\boldsymbol{x}, \boldsymbol{y})$. This arises from the inherent nature of foundation models designed to support a wide range of tasks across diverse domains (Touvron et al. 2023). Under such settings, conventional federation may lead to suboptimal performance, as it fails to account for task similarity and conflicts and treats all client contributions uniformly. This uniform aggregation can degrade personalized knowledge acquired from local fine-tuning, particularly when updates from irrelevant or conflicting tasks dominate, and hinder the exploitation of group-wise knowledge from similar tasks to enhance personalization. To empirically examine this, we conduct an experimental analysis comparing two training setups: (1) jointly training all tasks, and (2) fine-tuning each task independently. As shown in Figure 1 (a), we observe that for several tasks (e.g., Linguistic Acceptability), the independent performance is higher than that of joint training, indicating that inter-task conflicts exist under high heterogeneity. These findings suggest that naively aggregating all client models in conventional FL is suboptimal in such scenarios. Therefore, it is essential to develop new aggregation strategies that can account for task similarity and conflicts in FedFM.

Limited Parameter Variation in FedFM. As demonstrated in prior work (Li et al. 2019), under standard assumptions of convexity and smoothness, the performance of a model trained for T rounds can be bounded by $f(\theta_T) - f(\theta^*) \leq O(\frac{f(\theta_0)-f(\theta^*)}{T})$, where θ^* denotes the optimal model. In conventional FL, training typically begins from scratch, with random initialization $\theta_0 = \mathcal{N}(0, \sigma^2)$. In contrast, foundation models in FedFM are first pre-trained on large-scale datasets, then fine-tuning starts from the pre-trained weight $\theta_0 = \theta_{pre}$. Since pre-training imparts generalizable knowledge to provide a strong and generalized initialization for fine-tuning, the bound tends to be tighter for foundation models in FedFM compared to models trained from scratch in conventional FL, corresponding to relatively smaller model updates for optimization in FedFM. This limited variation between model parameters in FedFM poses significant challenges for adapting conventional FL methods, which typically rely on model parameter divergence to guide personalized aggregation and may fail to reflect meaningful task-level similarity or discrepancies, thereby undermining the effectiveness of these approaches. This issue becomes even more pronounced when using PEFT techniques, which tend to learn more similar parameters in a significantly lower-dimensional and restricted subspace. To empirically validate this, we compute the cosine similarity between model parameters fine-tuned on different tasks. As shown in Figure 1 (b), the results reveal that parameter differences are insufficient to reflect the underlying task divergence, highlighting the need for novel methods in FedFM.

Method

As previously analyzed, conventional uniform federation may degrade the personalization performance in FedFM, due to the disturbance from irrelevant or conflicting tasks and the underutilization of group-wise knowledge from similar tasks. Additionally, the conventional FL methods for balancing personalization and federation often fail to generalize to FedFM, as they rely on parameter similarities across clients, which are often insufficient to capture meaningful task-level distinctions with limited parameter variation in FedFM. To address these challenges, we propose a bi-level personalization framework with task-vector aggregation, as shown in Figure 2. During learning, each client performs local fine-tuning of the foundation model on its private dataset to achieve task-specific adaptation, representing the client-level personalization. Subsequently, the server conducts task-vector aggregation, computing client-specific aggregation weights based on inter-task relationships derived from task vectors for better personalization and federation balance, representing the server-level personalization.

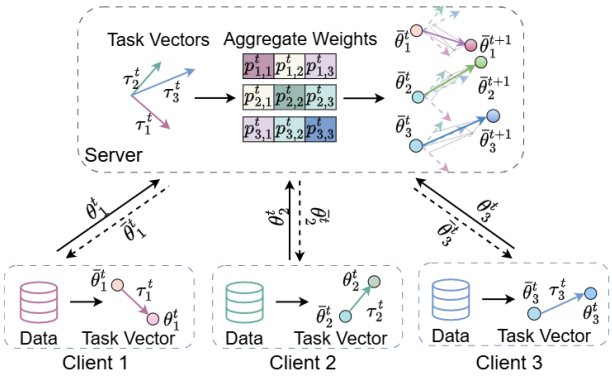

Figure 2: The overall framework of FedBip, consists of two components: client-level personalization, where each client fine-tunes the foundation model on its local data and computes a corresponding task vector; and server-level personalization, where the server performs personalized aggregation by computing task vector similarities and assigning client-specific aggregation weights to guide updates for each client.

Client-level Personalization

For client-level personalization, the objective is to adapt the foundation model to the specific task of each client. To achieve this, we follow the standard federated learning paradigm, where each client fine-tunes the model on its local dataset to obtain a personalized model. More specifically, during each communication round t, each client i receives its aggregated model $\bar{\theta}_i^t$ from the server as the initialization, and then fine-tunes it on its local dataset D_i for E local epochs, resulting in the personalized model θ_i^t, which is subsequently sent back to the server.

$$\theta_i^t = \min_{\theta} f_i(\theta; D_k), \quad \text{initialized with } \theta = \bar{\theta}_i^t \tag{2}$$

Server-level Personalization

Since conventional federated aggregation may degrade the personalization performance achieved at the client level, we propose a task-vector aggregation strategy to better balance personalization and federation, while enabling group-wise knowledge sharing for server-level personalization.

To balance personalization and federation, we seek to amplify the aggregation weights for clients with similar tasks,

while down-weighting those from irrelevant or conflicting clients. Conventional FL methods often achieve this by using each client's model parameter as its representation to infer task similarity or divergence. However, in FedFM, due to the limited parameter variation, such approaches may fail as discussed in Motivation. To overcome this limitation, we propose using task vectors as more informative representations to guide aggregation weights, inspired by prior work (Ilharco et al. 2022). Since each client in FedFM typically holds a local dataset corresponding to a different task, the task vector between the locally fine-tuned model and the aggregated model can capture task-specific characteristics, even under the limited parameter variation imposed by FedFM. To empirically validate this, we compute the cosine similarity between task vectors obtained from models fine-tuned on different tasks. As illustrated in Figure 1 (c), the results show that task vectors effectively reflect inter-task divergence, indicating their potential to serve as a signal for task-aware aggregation in FedFM.

Additionally, considering that the relationships between different tasks of clients are inherently non-uniform, we adopt a task-vector aggregation for each client instead of directly aggregating model parameters into a single global model. This approach offers two key advantages: (1) it enables more precise and adaptive updates by leveraging the most recent task-specific information for each client, thereby facilitating client-specific model updates; and (2) it mitigates the risk of error accumulation across communication rounds, as the aggregation focuses solely on the current round's task vectors, rather than recursively averaging full model weights. Moreover, since all clients fine-tune their models from a shared pre-trained initialization, the resulting task vectors reside in a common representation space, making it both meaningful and valid to perform vector arithmetic. As a result, the aggregated task vector can be reliably added to each client's previously personalized aggregated model, ensuring consistent and effective model adaptation.

More speicialy, on the server side, task vectors are first computed for each client i as:

$$\tau_i^t = \theta_i^t - \bar{\theta}_i^t, \forall i \in [K], \tag{3}$$

where $\bar{\theta}_i^t$ denotes the maintained aggregated model from the previous communication round, and the task vector τ_i^t captures the task-specific adaptation information of client i. To perform task-aware aggregation, we compute a set of similarity-based weights for each client. For client i, the aggregation weight associated with each peer client $k \in \{1, ..., K\}$ is computed as:

$$p_{i,k}^t = \frac{g(\tau_i^t, \tau_k^t)}{\sum_{j=1}^K g(\tau_i^t, \tau_j^t)}, \tag{4}$$

where $g(\cdot, \cdot)$ is a similarity function, and the weights are normalized across all clients to ensure $\sum_{k=1}^K p_{i,k}^t = 1$. Here, we choose cosine similarity as it is invariant to vector magnitude. Finally, the server computes a personalized aggregated model for client i as $\bar{\theta}_i^{t+1} = \bar{\theta}_i^t + \sum_{k=1}^K p_{i,k}^t \tau_k^t$, which is sent back to client i for next round. The overall process of FedBip is in Algorithm 1

Algorithm 1: FedBip

Input: Clients K, local datasets $\{D_1, ..., D_K\}$, local epoch E, communication rounds T
Output: Personalized models $\theta_1, ..., \theta_K$

Clients initialize local model
for $t = 1, ..., T$ **do**
  Server sends aggregated $\bar{\theta}_0^t, ..., \bar{\theta}_K^t$ to each client
  for each client $i \in [K]$ in parallel **do**
    $\theta_i^t \leftarrow$ClientUpdate$(\bar{\theta}_i^t, D_k, E)$
    Client i sends θ_i^t to the server
  end for
  for $i \in [K]$ **do**
    if FedBip **then**
      Server obtains $\tau_1^t, ..., \tau_K^t$ by Equation 3
      Server obtains $p_{i,1}^t, ..., p_{i,K}^t$ by Equation 4
      Server aggregates $\bar{\theta}_i^{t+1} = \bar{\theta}_i^t + \sum_{k=1}^K p_{i,k}^t \tau_k^t$
    end if
    if FedBip-L **then**
      Server obtains $\tau_{1,l}^t, ..., \tau_{K,l}^t$ by Equation 5
      Server obtains $p_{i,1,l}^t, ..., p_{i,K,l}^t$ by Equation 6
      Server aggregates $\bar{\theta}_{i,l}^{t+1} = \bar{\theta}_{i,l}^t + \sum_{k=1}^K p_{i,k,l}^t \tau_{k,l}^t$
    end if
  end for
end for
return $\theta_1, ..., \theta_K$

Layer-Wise Extension of FedBip Previous studies (Ma et al. 2022; Rehman et al. 2023; Lee, Zhang, and Avestimehr 2023) have demonstrated that model divergence often varies across different layers, highlighting layer-wise heterogeneity as a vital factor in FL. This phenomenon is also pronounced in foundation models, where different layers serve distinct roles—for instance, lower layers in transformer-based models tend to encode local features, while higher layers capture more abstract information (Chefer, Gur, and Wolf 2021). Motivated by this, we extend FedBip to support layer-wise task-vector aggregation, enabling finer-grained adjustment over the aggregation process. Specifically, for each round t, and each layer l, we compute the layer-wise task vector and corresponding aggregation weights as:

$$\tau_{i,l}^t = \theta_{i,l}^t - \bar{\theta}_{i,l}^t, \tag{5}$$

$$p_{i,k,l}^t = \frac{g(\tau_{i,l}^t, \tau_{k,l}^t)}{\sum_{j=1}^K g(\tau_{i,l}^t, \tau_{j,l}^t)}, \tag{6}$$

where $\theta_{i,l}^t$ denotes the parameters of the l-th layer of the fine-tuned model from client i and $\bar{\theta}_{i,l}^t$ denotes the parameters of the l-th layer of the previous aggregated model for client i. The server then performs layer-wise task-vector aggregation for client i as $\bar{\theta}_{i,l}^{t+1} = \bar{\theta}_{i,l}^t + \sum_{k=1}^K p_{i,k,l}^t \tau_{k,l}^t$.

Remark. FedBip is modular and flexible, allowing integration with other federated client-side optimization techniques and application to FedFM with PEFT technologies, without modifying the overall framework.

Methods	Para -phrase	Entail -ment	Structure to Text	Text For -matting	Linguistic Acc	Word Dis	Core -ference	Question CLS	Average	Client Comp.over	Server Exe.time
FedIT	77.00	82.50	71.99	96.61	76.00	62.00	73.26	83.00	79.05	0.281 TFLOPS	0.1s
FedAWA	70.00	85.50	71.55	96.82	75.50	63.00	73.86	93.50	78.72	0.281 TFLOPS	1.3s
L-DAWA	74.00	83.00	70.57	**96.97**	75.50	64.50	76.47	92.50	79.19	0.281 TFLOPS	0.2s
FedAMP	**83.50**	82.50	72.30	96.96	78.50	52.50	76.60	93.50	79.55	0.281 TFLOPS	2.2s
FedALA	75.00	84.50	71.13	96.51	76.00	**67.40**	76.6	90.50	79.71	0.562 TFLOPS	0.1s
FedBip	80.00	85.00	72.20	96.74	79.50	60.50	**82.20**	93.50	81.21	0.281 TFLOPS	2.4s
FedBip-L	81.00	**87.00**	**73.23**	96.61	**80.50**	62.00	79.89	**95.00**	**81.90**	0.281 TFLOPS	4.8s
Modularity											
FFA-LoRA	62.50	60.00	63.10	96.30	72.00	52.00	**61.83**	74.50	67.78	0.281 TFLOPS	0.1s
+FedBip	**72.50**	**74.50**	**69.42**	**96.40**	**79.00**	**53.75**	58.56	**86.00**	**73.77**	0.281 TFLOPS	2.4s
FedDPA	80.00	**87.50**	73.38	96.74	78.50	**64.00**	**79.84**	94.00	81.74	0.562 TFLOPS	0.1s
+FedBip	**84.50**	86.50	**73.76**	**96.81**	**78.50**	62.00	79.52	**94.50**	**82.01**	0.562 TFLOPS	2.4s

Table 1: Results of different models on NLP tasks with modularity and efficiency results.

Methods	Art	CliPart	Product	Real World	Average
FedAVG	84.71	86.36	94.44	94.26	90.02
FedAWA	86.36	86.09	94.31	94.46	90.31
L-DAWA	84.43	85.98	94.37	94.81	89.90
FedAMP	87.99	90.11	96.18	95.84	92.53
FedALA	**89.64**	90.89	96.18	96.39	93.33
FedBip	88.56	91.10	**97.97**	96.01	93.41
FedBip-L	88.02	**91.62**	97.94	**97.39**	**93.74**
Modularity					
FedProx	84.15	86.17	93.95	94.43	89.68
+ FedBip	**90.19**	**92.30**	**98.13**	**97.15**	**94.44**
Ditto	87.19	90.36	96.84	95.19	92.39
+FedBip	**88.02**	**91.96**	**97.72**	**96.84**	**93.63**

Table 2: Results of different models on CV tasks.

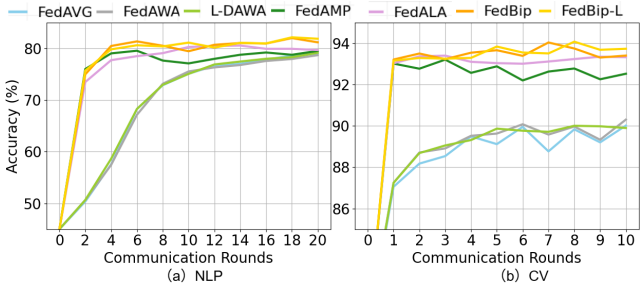

Figure 3: Accuracy via communication rounds.

Experiment

Experiment Setting

Datasets. To comprehensively evaluate the effectiveness of FedBip, we conduct experiments in both the computer vision (CV) and natural language processing (NLP) domains. For CV, we use the OfficeHome dataset (Venkateswara et al. 2017), which comprises images across four distinct domains with 65 categories to simulate cross-domain heterogeneity. For NLP, we utilize the Flan (Wei et al. 2021), which contains a diverse collection of instruction-following datasets, and we select eight distinct tasks as the federated dataset to simulate cross-task heterogeneity.

Baselines and Implementation. We compare our methods with below baselines based on the same model architecture: 1) conventional global aggregation methods: FedAVG (McMahan et al. 2017) for CV and FedIT (Zhang et al. 2023b) for NLP; 2) global model aggregation adjustment methods: FedAWA (Shi et al. 2025) and L-DAWA (Rehman et al. 2023); 3) personalized aggregation methods: FedAMP (Huang et al. 2021) and FedALA (Zhang et al. 2023a). To simulate data heterogeneity, we distribute clients based on domain (in CV) or task (in NLP). For both settings, to better evaluate the effectiveness of methods, we assume that all clients are activated for every communication round and set round $T = 10$ for CV and $T = 20$ for NLP. For CV, we use the ViT backbone from the CLIP model (Radford et al. 2021) and apply full fine-tuning. For NLP, we employ LoRA (Hu et al. 2022) as the PEFT method for LLaMA-7B.

Main Results

We evaluate FedBip against other baselines under two settings: full fine-tuning on cross-domain heterogeneity in CV and parameter-efficient fine-tuning on cross-task heterogeneity in NLP. As shown in Table 1 and Table 2, FedBip consistently achieves the best performance on average, demonstrating the effectiveness of our method for considering both client-level and server-level personalization. Moreover, the layer-wise extension (FedBip-L) further improves performance over FedBip, highlighting the importance of capturing layer-specific differences across clients. In addition, personalized aggregation methods generally outperform global aggregation baselines, emphasizing the necessity of aligning the model with client-specific distributions to handle heterogeneous tasks or domains. In particular, for tasks such as Linguistic Acceptability, where independent fine-tuning outperforms joint training (as shown in Figure 1 (a)), our method significantly improves performance. This provides further empirical evidence that FedBip effectively balances the federation and personalization in FedFM.

Modularity. FedBip is designed as a modular framework that can be integrated with client-side optimization FL algorithms to enhance performance. As shown in Table 1 and

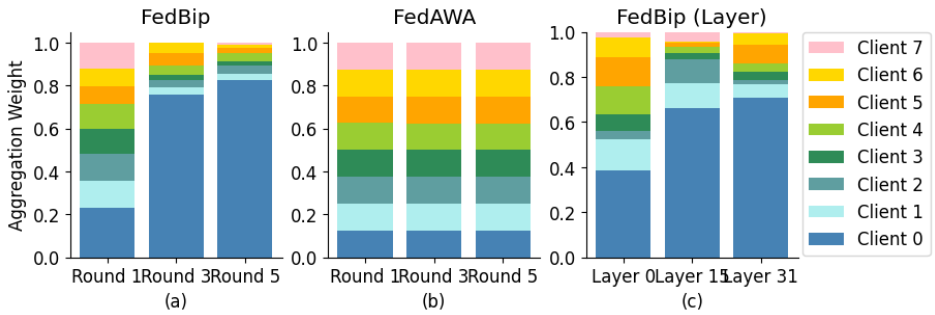

Figure 4: Client 0's aggregation weights calculated by FedBip, FedAWA and FedBip-L.

Table 2, we integrate FedBip with several representative FL baselines, including LoRA-based methods (FFA-LoRA (Sun et al. 2024), FedDPA (Yang et al. 2024)) in NLP, and global method FedProx (Li et al. 2020) and personalized method Ditto (Li et al. 2021a) in CV. The results show consistent average performance gains across all combinations, demonstrating the strong modularity of our approach.

Flexibility. FedBip is highly flexible, supporting different transformer-based foundation models and fine-tuning methods. As illustrated in Table 1 and Table 2, FedBip is successfully deployed across models of varying scales and domains, including ViT (for CV) and LLaMA (for NLP), and performs well under both full fine-tuning and LoRA-based PEFT. The consistent performance improvements over baseline methods confirm that FedBip is a flexible and effective framework for addressing personalization in FedFM.

Analysis

Convergence Analysis. To analyze convergence, we compare average test accuracy versus communication rounds in Figure 3. The results show that personalized aggregation methods, including FedBip, converge faster than global aggregation methods and reach stable performance after only a few rounds. Moreover, our method consistently outperforms all baselines throughout the training process.

Efficiency Analysis. We evaluate the computation cost on the client side and execution time on the server side. As shown in Table 1, FedBip introduces no additional computation overhead for clients and only a slight increase in server-side execution time, demonstrating that FedBip is a lightweight and efficient framework for FedFM fine-tuning.

Ablation Analysis. To evaluate scalability, we vary client number $K \in \{8, 24, 40\}$ in Table 3, where FedBip consistently outperforms other baselines, demonstrating its effectiveness and robustness under increasing client heterogeneity and numbers. To evaluate impact of $g(\cdot, \cdot)$ in task-vector aggregation, we compare different similarity metrics in Table 4, where cosine similarity performs best due to its magnitude invariance and better robustness for high-dimensional vectors. Table 5 evaluates different weighting and aggregation strategies, showing computing aggregation weights and performing aggregation directly on task vectors yields the best performance. This aligns with our earlier analysis that parameter-based similarity fails to capture task divergence, and parameter aggregation may lead to error accumulation.

#Client	8	24	40
FedIT	79.05	65.07	66.67
FedAWA	78.72	64.69	66.49
L-DAWA	79.19	64.64	65.78
FedAMP	79.55	66.37	68.09
FedALA	79.71	67.24	71.53
FedBip	**81.21**	**68.91**	**73.00**

Table 3: Ablation study of different client numbers.

Similarity Metric	L2	Pearson	Cosine
FedBip	79.55	80.77	81.21

Table 4: Ablation of similarity metric.

Weighting Aggregation	Parameter Parameter	Parameter Vector	Vector Vector
	79.63	80.15	81.21

Table 5: Ablation of weighting and aggregation strategies.

Aggregation Weight Analysis. We analyze the aggregation weights computed by FedBip, FedAWA, and FedBip-L in Figure 4. (a) shows that weights of FedBip increasingly favor personalization over rounds, and FedBip effectively assigns higher weights to similar tasks while reducing influence of irrelevant ones compared with (b). (c) illustrates the variation in layer-wise aggregation weights of FedBip-L, indicating that lower layers share more common knowledge, whereas higher layers retain more personalized information.

Conclusion

FedFM fine-tuning aims to adapt pre-trained foundation models for a small group of new users across distributed clients, where balancing federation and personalization presents a critical challenge. To address this, we propose FedBip, a bi-level personalization framework that incorporates client-level personalization through local fine-tuning and server-level personalization via task-vector aggregation. Experimental results on both CV and NLP benchmarks demonstrate the effectiveness of FedBip, paving the way for future exploration of more advanced methods across diverse modalities and large-scale federated learning scenarios.

References

Chefer, H.; Gur, S.; and Wolf, L. 2021. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 782–791.

Chen, H.; Zhang, Y.; Krompass, D.; Gu, J.; and Tresp, V. 2024. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 11285–11293.

Cho, Y. J.; Liu, L.; Xu, Z.; Fahrezi, A.; Barnes, M.; and Joshi, G. 2023. Heterogeneous lora for federated fine-tuning of on-device foundation models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*.

Guo, P.; Zeng, S.; Wang, Y.; Fan, H.; Wang, F.; and Qu, L. 2024. Selective Aggregation for Low-Rank Adaptation in Federated Learning. *arXiv preprint arXiv:2410.01463*.

Han, S.; Buyukates, B.; Hu, Z.; Jin, H.; Jin, W.; Sun, L.; Wang, X.; Wu, W.; Xie, C.; Yao, Y.; et al. 2024a. Fedsecurity: A benchmark for attacks and defenses in federated learning and federated llms. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5070–5081.

Han, Z.; Gao, C.; Liu, J.; Zhang, J.; and Zhang, S. Q. 2024b. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; and Zhang, Y. 2021. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 7865–7873.

Ilharco, G.; Ribeiro, M. T.; Wortsman, M.; Gururangan, S.; Schmidt, L.; Hajishirzi, H.; and Farhadi, A. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Lee, S.; Zhang, T.; and Avestimehr, A. S. 2023. Layer-wise adaptive model aggregation for scalable federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 8491–8499.

Li, T.; Hu, S.; Beirami, A.; and Smith, V. 2021a. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, 6357–6368. PMLR.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2: 429–450.

Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2019. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*.

Li, X.-C.; Zhan, D.-C.; Shao, Y.; Li, B.; and Song, S. 2021b. Fedphp: Federated personalization with inherited private models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 587–602. Springer.

Li, Z.; Lin, T.; Shang, X.; and Wu, C. 2023. Revisiting weighted aggregation in federated learning with neural networks. In *International Conference on Machine Learning*, 19767–19788. PMLR.

Luo, J.; and Wu, S. 2022. Adapt to adaptation: Learning personalization for cross-silo federated learning. In *IJCAI: proceedings of the conference*, volume 2022, 2166.

Ma, X.; Zhang, J.; Guo, S.; and Xu, W. 2022. Layer-wised model aggregation for personalized federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10092–10101.

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

Morafah, M.; Kungurtsev, V.; Chang, H.; Chen, C.; and Lin, B. 2024. Towards diverse device heterogeneous federated learning via task arithmetic knowledge integration. *Advances in Neural Information Processing Systems*, 37: 127834–127877.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmLR.

Rehman, Y. A. U.; Gao, Y.; De Gusmão, P. P. B.; Alibeigi, M.; Shen, J.; and Lane, N. D. 2023. L-dawa: Layer-wise divergence aware weight aggregation in federated self-supervised visual representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16464–16473.

Ren, C.; Yu, H.; Peng, H.; Tang, X.; Li, A.; Gao, Y.; Tan, A. Z.; Zhao, B.; Li, X.; Li, Z.; et al. 2024. Advances and Open Challenges in Federated Learning with Foundation Models. *arXiv preprint arXiv:2404.15381*.

Shi, C.; Zhao, H.; Zhang, B.; Zhou, M.; Guo, D.; and Chang, Y. 2025. FedAWA: Adaptive Optimization of Aggregation Weights in Federated Learning Using Client Vectors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 30651–30660.

Sun, B.; Huo, H.; Yang, Y.; and Bai, B. 2021. Partialfed: Cross-domain personalized federated learning via partial initialization. *Advances in Neural Information Processing Systems*, 34: 23309–23320.

Sun, Y.; Li, Z.; Li, Y.; and Ding, B. 2024. Improving loRA in privacy-preserving federated learning. *arXiv preprint arXiv:2403.12313*.

Tao, Z. S.; Mason, I.; Kulkarni, S.; and Boix, X. 2024. Task arithmetic through the lens of one-shot federated learning. *arXiv preprint arXiv:2411.18607*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5018–5027.

Wang, Z.; Shen, Z.; He, Y.; Sun, G.; Wang, H.; Lyu, L.; and Li, A. 2024. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *arXiv preprint arXiv:2409.05976*.

Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Yang, Y.; Long, G.; Shen, T.; Jiang, J.; and Blumenstein, M. 2024. Dual-Personalizing Adapter for Federated Foundation Models. *arXiv preprint arXiv:2403.19211*.

Yang, Y.; Long, G.; Zhou, T.; Lu, Q.; Ye, S.; and Jiang, J. 2025. Federated Adapter on Foundation Models: An Out-Of-Distribution Approach. *arXiv preprint arXiv:2505.01075*.

Ye, R.; Xu, M.; Wang, J.; Xu, C.; Chen, S.; and Wang, Y. 2023. Feddisco: Federated learning with discrepancy-aware collaboration. In *International Conference on Machine Learning*, 39879–39902. PMLR.

Yu, S.; Muñoz, J. P.; and Jannesari, A. 2023. Federated Foundation Models: Privacy-Preserving and Collaborative Learning for Large Models. *arXiv preprint arXiv:2305.11414*.

Zhang, J.; Hua, Y.; Wang, H.; Song, T.; Xue, Z.; Ma, R.; and Guan, H. 2023a. Fedala: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11237–11244.

Zhang, J.; Vahidian, S.; Kuo, M.; Li, C.; Zhang, R.; Wang, G.; and Chen, Y. 2023b. Towards Building the Federated GPT: Federated Instruction Tuning. *arXiv preprint arXiv:2305.05644*.

Zhang, Z.; Yang, Y.; Dai, Y.; Wang, Q.; Yu, Y.; Qu, L.; and Xu, Z. 2023c. FedPETuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In *Annual Meeting of the Association of Computational Linguistics 2023*, 9963–9977. Association for Computational Linguistics (ACL).

Zhuang, W.; Chen, C.; and Lyu, L. 2023. When foundation model meets federated learning: Motivations, challenges, and future directions. *arXiv preprint arXiv:2306.15546*.