Rethinking the Starting Point: Collaborative Pre-Training for Federated Downstream Tasks

Yun-Wei Chu[1], **Dong-Jun Han**[2], **Seyyedali Hosseinalipour**[3], **Christopher G. Brinton**[1]
[1]Purdue University
[2]Yonsei University
[3]University at Buffalo-SUNY
chu198@purdue.edu, djh@yonsei.ac.kr, alipour@buffalo.edu, cgb@purdue.edu

Abstract

A few recent studies have shown the benefits of using centrally pre-trained models to initialize federated learning (FL). However, existing methods do not generalize well when faced with an arbitrary set of downstream FL tasks. Specifically, they often (i) achieve limited accuracy, especially with unseen downstream labels, and (ii) result in significant accuracy variance, failing to provide a balanced performance across clients. To address these challenges, we propose `CoPreFL`, a collaborative/distributed pre-training approach that robustly initializes for downstream FL tasks. `CoPreFL` leverages model-agnostic meta-learning (MAML) that tailors the global model to mimic heterogeneous and unseen FL scenarios, resulting in a pre-trained model that is rapidly adaptable to any FL task. Our MAML procedure integrates performance variance into the meta-objective function, balancing performance across clients rather than solely optimizing for accuracy. Extensive experiments show that `CoPreFL` significantly enhances average accuracy and reduces variance in arbitrary downstream FL tasks with unseen/seen labels, outperforming various pre-training baselines. Additionally, `CoPreFL` proves compatible with different well-known FL algorithms used in downstream tasks, boosting performance in each case.

1 Introduction

Federated learning (FL) has gained prominence as a distributed machine learning framework, enabling collaborative training among clients by periodic aggregations of local models on a server (McMahan et al. 2017; Konecný et al. 2016). Recent research has extensively explored various aspects of FL, such as aggregation schemes (Ji et al. 2019; Wang et al. 2020) or local training techniques (Reddi et al. 2021; Sahu et al. 2018). One aspect that remains understudied, however, is the impact of *model initialization* in FL. While pre-training boosts performance in centralized AI/ML (Radford et al. 2019; Devlin et al. 2019; Dosovitskiy et al. 2021), most FL works still rely on random weight initialization instead of well pre-trained models.

Motivation. Recently studies (Nguyen et al. 2023; Chen et al. 2023) show that initializing FL with *centrally pre-trained models* can enhance average client performance. Yet, these methods face significant challenges, particularly when handling *newly emerging* and/or *heterogeneous* downstream FL tasks unanticipated during pre-training. These include: (i) limited average accuracy (despite outperforming random initialization), due to unseen data and labels, and (ii) high performance variance, leading to imbalanced client accuracy. The histograms in Figure 1 show the performance of various pre-trained models in multiple downstream image classification tasks, illustrating these limitations. While centrally pre-trained models improve average accuracy over random initialization, they introduce significant variance across clients, a well-cited concern in distributed AI/ML (Li et al. 2020; Cho et al. 2022). Moreover, the achievable average accuracy of centralized pre-training remains suboptimal, indicating difficulties in capturing data heterogeneity and diversity in downstream FL tasks.

Goals. Motivated by these limitations, we aim to develop a robust FL pre-training methodology that achieves two main objectives: (i) *improved average accuracy*, and (ii) *reduced performance variance* for balanced client accuracy in downstream tasks. This is challenging as it must work across *any arbitrary set* of downstream FL tasks, including unseen data and labels due to factors like time-varying environments or new clients joining the system. Thus, the pre-trained model must handle unfamiliar classes and data heterogeneity during downstream FL tasks, a challenge overlooked by existing methods (Nguyen et al. 2023; Chen et al. 2023). We summarize our research question as follows:

> *How can we design a pre-training strategy that can simultaneously (i) enhance average accuracy and (ii) reduce performance variance across clients, for an arbitrary set of downstream FL tasks which possess heterogeneity in their data statistics as well as unseen labels?*

Contributions. We propose `CoPreFL`, a Collaborative Pre-training approach for handling an arbitrary set of downstream FL tasks, to address the above question. We make the following key contributions:

- *Distributed pre-training infused with meta-learning*: The core of `CoPreFL` is an FL-inspired pre-training procedure which employs model-agnostic meta-learning (MAML)-based updates on the collaboratively-bulit global model. Through our developed MAML procedure, `CoPreFL` ensures robust initializations, enabling the pre-trained model to easily adapt to unseen

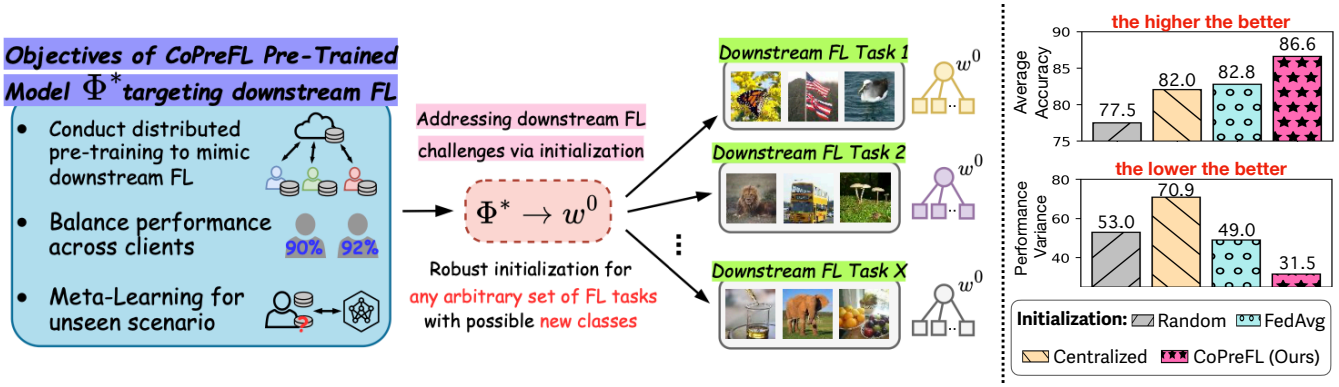

Figure 1: (Left): Overview of CoPreFL, aiming to provide robust initialization for arbitrary downstream FL tasks. (Right): Average accuracy and variance of FL tasks initialized by various pre-trained models. Centralized pre-training achieves limited performance, failing to capture heterogeneous characteristics of unforeseen FL settings. CoPreFL improves both accuracy and variance by strategically mimicking downstream FL scenarios during pre-training.

labels and various data distributions in any downstream FL tasks. This approach differs in purpose and method from prior meta-learning works for FL personalization: since our downstream tasks aim to construct a global model rather than client-specific personalized models, we conduct meta-updates based on the *global model* instead of directly using local models.

- *Meta-objective function incorporating variance*: To enhance average accuracy while improving performance balance among clients in the downstream FL tasks, we explicitly incorporate both expected loss and performance variance into the meta-objective function during pre-training in CoPreFL. In doing so, we introduce a first-order approximation for efficiently computing the gradient of the proposed meta-objective function.
- *Relaxing the assumption of centrally stored pre-training data*: CoPreFL relaxes the assumption made by existing works that all pre-training data is stored centrally. Our pre-training algorithm supports hybrid client-server data storage, where (i) data is exclusively held by distributed clients, or (ii) the server also holds partial data. This is crucial for FL applications with data privacy limitations. Our approach also works with centrally stored pre-training data, as validated by our experiments.
- *Extensive experiments across diverse downstream FL tasks*: We evaluate CoPreFL against various baselines on downstream FL tasks with varying data distributions, seen/unseen labels, and client-server data allocations. Results show notable improvements in accuracy and performance variance when downstream tasks are initialized with CoPreFL. We also show CoPreFL's compatibility with various popular FL algorithms used downstream and its resilience to distributional shifts.

Our work is among the first to consider FL in both the pre-training and downstream stages of distributed learning tasks. We introduce several unique features tailored to FL, including meta-updating the global model during distributed pre-training, hybrid client-server learning, and balancing between average performance and variance across the clients.

2 Related Work

Pre-training for FL. While pre-training is well-studied in centralized AI/ML (Radford et al. 2019; Brown et al. 2020; Devlin et al. 2019; Dosovitskiy et al. 2021), its impact on downstream FL tasks remains underexplored. A few recent works show that starting FL with centrally pre-trained models can improve performance over random initializations (Nguyen et al. 2023; Chen et al. 2023). However, as observed in Figure 1, such strategies often lead to high performance variance and limited average accuracy, as they fail to mimic diverse downstream FL settings. To address this, we propose CoPreFL, a MAML-based pre-training strategy tailored for distributed downstream settings, improving both average accuracy and performance variance while addressing the challenges of heterogeneous/unseen data encountered in downstream FL tasks.

Meta-learning in FL. CoPreFL utilizes meta-learning to create a global model adaptable to any downstream FL tasks, optimizing both performance and variance. This distinguishes it from other meta-learning-based FL, like personalized FL (Chen et al. 2018; Jiang et al. 2019; Fallah, Mokhtari, and Ozdaglar 2020; Chu et al. 2022), which focuses on individual client performance, and few-round FL (Park et al. 2021), which adapts quickly but ignores performance imbalance. Our goal is to develop a pre-trained model that ensures high accuracy and balanced performance for *global models* in downstream FL tasks.

Performance imbalance in FL. Several works in FL address performance imbalance (Mohri, Sivek, and Suresh 2019; Li et al. 2020; Cho et al. 2022) typically by creating a global model that satisfies as many clients as possible (e.g., achieving a uniform accuracy across clients). Such models are more likely to satisfy new clients joining the FL system. These methods can be applied downstream from our pre-training methodology, whose primary objective is to build a robust *initial* model that will lead to higher average and more balanced performance across clients after FL training.

3 Proposed CoPreFL Methodology

3.1 Problem Setup and Pre-Training Objectives

Federated downstream tasks. Referring to Figure 1, CoPreFL aims to provide a robust initialization for any downstream FL tasks. Each task assumes a central server connected to a set of clients G. Starting with an initial model w^0, each FL task iterates between (i) local client training and (ii) global server aggregation over multiple communication rounds. In each downstream round r, every client $g \in G$ downloads the previous global model w^{r-1} from the server, subsequently updates it through multiple iterations of stochastic gradient descent (SGD) using its local dataset, denoted D_g, and uploads their updated models $w^r g$ to the server for aggregation. This aggregation results in a new global model $w^r = \sum_{g \in G} \frac{|D_g|}{|D|} w_g^r$ assuming FedAvg (McMahan et al. 2017) is employed, where $|D|$ is the total data samples across all clients. This process repeats for $r = 1, ..., R$ rounds for each task.

Pre-training scenarios. One of our contributions is relaxing the assumption that all pre-training data is stored centrally. To this end, we consider two distributed pre-training scenarios for CoPreFL:

- **Scenario I:** Pre-training datasets are exclusively available at distributed clients.
- **Scenario II:** A hybrid scenario where the server also holds a small amount of pre-training data.

Scenario I simulates downstream FL tasks where pre-training labels and data may differ from those in downstream tasks. Scenario II represents settings where the server holds data reflecting the broader population distribution (e.g., a self-driving car manufacturer with database of roadway images). Such hybrid FL settings that combine client data with a relatively small portion of server data are becoming popular (Yang, Chen, and Shen 2023; Bian et al. 2023), but remain underexplored for pre-training. Further, as we will discuss in **Remark 2**, our method is still applicable even when all pre-training data is centralized.

Pre-training objectives. Our goal is to design a pre-trained model Φ^* as a robust initialization $w^0 = \Phi^*$ for any downstream FL task. Specifically, given $\mathcal{G}$ as possible client sets in a task, Φ^* is optimized to minimize the following objective function:

$$A(\Phi) = \mathbb{E}_{G \sim p(\mathcal{G})} \left[\frac{1}{|G|} \sum_{g \in G} f(w^R(\Phi, G), D_g) \right], \tag{1}$$

where $p(\mathcal{G})$ represents the probability distribution over $\mathcal{G}$, G is a specific client group (i.e., a specific task) drawn from $p(\mathcal{G})$, $f(\cdot)$ is the per-client loss function for downstream training, $w^R(\Phi, G)$ symbolizes the final R-th round global model derived from client set G when initialized by Φ, and D_g represents the local dataset of client g. $A(\Phi)$ denotes the average FL performance across all clients fir downstream tasks, with each group weighted by likelihood of occurrence.

On the other hand, FL settings can lead to significant performance variations among clients, especially when the aggregated models are biased towards those with larger datasets. This performance variation can be measured by the variance in testing accuracy across participants (Li et al. 2020). Thus, besides improving performance for any FL task, we aim for the final global model $w^R(\Phi^*, G)$ initialized from our pre-trained model Φ^* to achieve balanced testing performance across client set G. Specifically, our second objective for Φ^* is to minimize the variance of the loss distribution across participants in downstream FL tasks, i.e.,

$$F(\Phi) = \mathbb{E}_{G \sim p(\mathcal{G})} \Bigg[\frac{1}{|G|} \sum_{g \in G} f^2(w^R(\Phi^*, G), D_g) - \left(\frac{1}{|G|} \sum_{g \in G} f(w^R(\Phi^*, G), D_g) \right)^2 \Bigg]. \tag{2}$$

Overview of approach. One of our key contributions is balancing (1) and (2). The challenge arises as D_g, $\mathcal{G}$, and $p(\mathcal{G})$ are unknown during pre-training, preventing us from directly optimizing $A(\Phi)$ and $F(\Phi)$. To address this, we develop a model-agnostic meta-learning (MAML) approach in CoPreFL to mimic statistical heterogeneity of downstream FL tasks. This method yields pre-trained models that offer robust initialization for unseen downstream tasks, considering (1) and (2). Detailed in Sections 3.2 and 3.3, we construct a pre-training environment for scenarios I and II that mirrors downstream federated setups, enabling the pre-trained model to handle data heterogeneity across clients and tasks. Our meta-learning-based CoPreFL updates the pre-trained model iteratively over federated rounds using a support set, with a concluding adjustment (meta-update) using a query set treated as unseen knowledge. This enables our pre-trained model to the effectively handle unforeseen FL scenarios downstream while balancing between (1) and (2).

3.2 CoPreFL in Scenario I (Pre-training with Distributed Clients)

We first consider the scenario where pre-training data is distributed across M clients, with no data stored on the server. The detailed procedure of CoPreFL is given in Algorithm 1. In each round $t = 1, ..., T$ of pre-training, a set of clients $m \subset M$ is randomly selected to participate in the current round. Each participating client $j \in m$ splits its local pre-training dataset D_j^p into disjoint support (S_j) and query (Q_j) sets. These steps mimic variations in downstream tasks by changing the participating clients across rounds and holding out query sets, allowing our meta-learning to generalize to unseen downstream scenarios.

Temporary pre-training model construction. In each round t, participating clients $j \in m$ download Φ^{t-1} from the server. Subsequently, clients perform local training using their support sets S_j, yielding a local support loss $\ell_{S_j}^e(\Phi^t)$ per epoch e, defined in line 8 of Algorithm 1, where $\ell(\cdot)$ is the per-datum loss function (e.g., cross-entropy for classification). After all participants finish E epochs, we obtain the updated local model $\Phi_j^{t,E}$. Clients then send their updated models to the server for aggregation, resulting in $\overline{\Phi^t}$ (defined in line 12). This model can be viewed as the temporary pre-training model that will be further refined using query

Algorithm 1: Our Pre-training Method CoPreFL (Pre-training Phase in Scenario I)

Input: A set of clients M in the pre-training phase, with each client i holding its pre-training dataset D_i^p.
for Each pre-training round $t = 1, 2, ..., T$ **do**
  Randomly select a set of clients $m \subset M$ to participate
  Each participant $j \in m$ partitions its own dataset D_j^p into support set S_j and query set Q_j
  for Each participant j in parallel **do**
    Download Φ^{t-1} from the server
    for local epoch $e = 1, 2, ..., E$ **do**
      $\ell_{S_j}^e(\Phi^t) \leftarrow \frac{1}{|S_j|}\sum_{(x,y)\in S_j} \ell(\Phi_j^{t,e}(x), y)$ { Compute local support loss at each epoch}
      $\Phi_j^{t,e} \leftarrow \Phi_j^{t,e-1} - \eta\nabla\ell_{S_j}^e(\Phi^t)$ { Perform SGD local update using support loss}
    end for
  end for
  $\overline{\Phi^t} \leftarrow \sum_{j\in m} \frac{|S_j|}{\sum_{i\in m}|S_i|}\Phi_j^{t,E}$ { Model aggregation to construct temporary global model}
  for Each participant j in parallel **do**
    Download $\overline{\Phi^t}$ from the server
    $\mathcal{L}_{Q_j}(\overline{\Phi^t}) \leftarrow \frac{1}{|Q_j|}\sum_{(x,y)\in Q_j} \ell(\overline{\Phi^t}(x), y)$ { Compute local loss (and gradient) using query set Q_j}
  end for
  Server computes overall meta-loss $\mathcal{L}_Q(\overline{\Phi^t})$ and variance across meta-losses $\sigma_Q^2(\overline{\Phi^t})$ according to (3)
  $\mathcal{L}_{meta}(\overline{\Phi^t}) = \gamma\mathcal{L}_Q(\overline{\Phi^t}) + (1-\gamma)\sigma_Q^2(\overline{\Phi^t})$ { Customized query meta-loss}
  $\Phi^t \leftarrow \overline{\Phi^t} - \zeta\nabla\mathcal{L}_{meta}(\overline{\Phi^t})$ { Meta-learning model update using customized loss}
end for
Output: A pre-trained model for downstream FL tasks: Φ^T

sets, with the objective of obtaining robust global models at the conclusion of downstream task training.

Measuring average performance and variance. Next, the query sets are used to evaluate the performance of the temporary pre-training model on each client, mimicking the scenario where the pre-trained model encounters unseen data, and to conduct meta-updates to promote downstream generalization. CoPreFL aims to strike a balance between the following objectives during pre-training:

$$\begin{aligned}\mathcal{L}_Q(\overline{\Phi^t}) &= \sum_{j\in m}\mathcal{L}_{Q_j}(\overline{\Phi^t}) \text{ and}\\ \sigma_Q^2(\overline{\Phi^t}) &= \frac{1}{|m|}\sum_{j\in m}\Big(\mathcal{L}_{Q_j}(\overline{\Phi^t}) - \frac{1}{|m|}\mathcal{L}_Q(\overline{\Phi^t})\Big)^2,\end{aligned} \tag{3}$$

where $\mathcal{L}_{Q_j}$ represents the loss evaluated using query sets Q_j of participants, $\mathcal{L}_Q$ denotes the overall query loss (characterized by aggregating $\mathcal{L}_{Q_j}$ across all participants), and σ_Q^2 represents the performance variance evaluated using clients' query losses. To balance the performance-variance trade-off, we construct a customized query meta-loss function $\mathcal{L}_{meta}(\overline{\Phi^t})$ to minimize both the overall query loss $\mathcal{L}_Q(\overline{\Phi^t})$ when encountering unseen data and the variance $\sigma_Q^2(\overline{\Phi^t})$ of query losses across participants. Formally, we aim to solve:

$$\min_{\Phi}\mathcal{L}_{meta}(\overline{\Phi^t}) = \min_{\Phi}\left[\gamma\mathcal{L}_Q(\overline{\Phi^t}) + (1-\gamma)\sigma_Q^2(\overline{\Phi^t})\right], \tag{4}$$

where $\gamma \in [0, 1]$ represents a balancer between the average performance and variance. Setting $\gamma = 0$ encourages a more uniform accuracy distribution, aligning with σ_Q^2, but may sacrifice average performance. A larger γ emphasizes the average performance with less consideration for uniformity, optimizing the pre-trained model more towards $\mathcal{L}_Q$.

Model-agnostic meta update. Considering the objective function in (4), each participant j downloads the temporary global model $\overline{\Phi^t}$ and employs its query set Q_j to compute its local query loss $\mathcal{L}_{Q_j}(\overline{\Phi^t})$, as in line 15 in Algorithm 1. The gradients are also computed locally and sent back to the server, as both are necessary to conduct the meta-update. On the server-side, the overall query meta-loss $\mathcal{L}_Q(\overline{\Phi^t})$ and the performance variance $\sigma_Q^2(\overline{\Phi^t})$ are computed, according to (3). Then, CoPreFL updates the temporary pre-training model $\overline{\Phi^t}$ through a gradient step with the customized query meta-loss $\mathcal{L}_{meta}$ and the aggregated received gradients, to align it with (4). To derive the meta-loss $\nabla_{\Phi^{t-1}}\mathcal{L}_{meta}(\overline{\Phi^t})$, we express it through the chain rule as $\nabla_{\overline{\Phi^t}}\mathcal{L}_{meta}(\overline{\Phi^t}) \times \frac{\partial\overline{\Phi^t}}{\partial\Phi^{t-1}}$. Writing $\overline{\Phi^t} = \sum_{j\in m}\frac{|S_j|}{\sum_{i\in m}|S_i|}\Phi_j^{t,E} = \sum_{j\in m}\frac{|S_j|}{\sum_{i\in m}|S_i|}(\Phi^{t,E-1} - \eta\nabla\ell_{S_j}^E(\Phi^t))$, it follows that

$$\begin{aligned}\nabla_{\Phi^{t-1}}\mathcal{L}_{meta}(\overline{\Phi^t}) = \nabla_{\overline{\Phi^t}}\mathcal{L}_{meta}(\overline{\Phi^t})\times\\ \Big(1 - \eta\sum_{j\in m}\frac{|S_j|}{\sum_{i\in m}|S_i|}\frac{\partial}{\partial\Phi^{t-1}}\nabla\ell_{S_j}^E(\Phi^t)\Big).\end{aligned} \tag{5}$$

If we ignore the second derivative term, the meta-loss gradient can be approximated as $\nabla_{\overline{\Phi^t}}\mathcal{L}_{meta}(\overline{\Phi^t})$. This is similar to making a first-order approximation to a meta-update, a common practice in the implementation of MAML variants to reduce complexity (Finn, Abbeel, and Levine 2017).

The server then sends the meta-updated global model Φ^t to a new set of participants to begin the next round of pre-training. After T rounds, the final global model Φ^T serves as the pre-trained model for initializing FL in the downstream tasks, i.e., in Figure 1, clients in any downstream task conduct FL starting from the pre-trained model $w^0 = \Phi^T$.

Remark 1 (Key characteristics of CoPreFL meta-update). Using the *query datasets*, CoPreFL applies a meta-update to the temporary pre-training model–a *global model* developed through FL on the *support datasets*. This differs significantly from existing meta-learning based FL methods, which update *client models* for personalization, as discussed in Section 2. Our method aims to tailor the pre-training model to adapt to any downstream FL tasks, addressing robustness against unseen and heterogeneous data, unlike existing personalization methods. As we will see in Section 4, this leads to notable improvements of CoPreFL over employing these prior methods for pre-training.

3.3 CoPreFL in Scenario II (Hybrid Client-Server Pre-Training)

We next explore a pre-training scenario where the server holds a small dataset D^s drawn from the broader population distribution, alongside client-held data. Unlike scenario I, where client data was split into support and query sets, in

scenario II, all client samples are used as support data, while the server's data serves as the query set.

The procedure of CoPreFL for scenario II is detailed in Algorithm 2 in Appendix B[1]. Here, we highlight the key differences from Algorithm 1. First, the temporary global model $\overline{\Phi^t}$ is aggregated from local models trained on each participant's full local dataset D_j^p. Second, the meta-update of the temporary global model $\overline{\Phi^t}$ utilizes the server's data. To mimic downstream FL tasks, we randomly split the server dataset D^s into $|m|$ parts equally to compute average loss and variance objectives, similar to (3). The temporary global model $\overline{\Phi^t}$ is then updated using meta-loss $\mathcal{L}_{meta}(\overline{\Phi^t})$, calculated through meta-updates on these partitions.

Note that unequal and/or non-uniform partition of the server-side dataset for meta-updating could be alternatives. However, due to the server's lack of prior knowledge about future downstream FL tasks during pre-training, including their dataset sizes and distributions, random query set allocation remains the most viable solution. We show in Section 4 that this partitioning provides significant performance improvements over other pre-training strategies.

Remark 2 (Applications to centralized datasets). Although we present CoPreFL for two distributed scenarios, it is applicable even when all pre-training data is stored at the server (e.g., public datasets). The server can intentionally split the dataset to mimic scenarios I or II and directly apply CoPreFL. We will show in Section 4 that CoPreFL surpasses standard centralized pre-training even in this setup, offering initializations better prepared for downstream data heterogeneity in FL setups.

Remark 3. The theoretical link between pre-training strategies and downstream task performance remains an open problem. This challenge has existed both in centralized-to-centralized (Chang et al. 2020; Dong et al. 2023; Zhang et al. 2022; Yuan et al. 2024) and centralized-to-federated (Nguyen et al. 2023; Chen et al. 2023) transfers from pre-training to downstream, and persists in the distributed-to-federated case we consider. We thus leave theoretical analysis of CoPreFL to future work, and instead validate its effectiveness through extensive experiments. The success of CoPreFL stems from meta-learning, which boosts robustness across diverse downstream tasks.

4 Experiments

4.1 Experimental Setup

Datasets and model. For evaluation, we use CIFAR-100 (Krizhevsky 2009), Tiny-ImageNet (Le and Yang 2015), FEMNIST (Caldas et al. 2018), and PACS (Li et al. 2017), following data splits provided in (Park et al. 2021), and adopt ResNet-18 (He et al. 2015). To model scenarios where downstream task labels are unknown during pre-training, we divide CIFAR-100 into 80 classes for pre-training and 20 for downstream tasks, and Tiny-ImageNet into 160 and 40 classes, respectively. We also explore mixed scenarios involving overlapping classes between pre-training and downstream tasks. Following (Yang, Chen, and Shen 2023; Zhang et al. 2020), we allocate 95% of the samples from the pre-training dataset for clients, and the remaining 5% form the server dataset. For PACS, we adopt a one-domain-leave-out setup (Li et al. 2022; Zhou et al. 2021) using different data domains for pre-training and downstream tasks. Detailed dataset information is available in Appendix C.1.

Pre-training phase. We distribute the pre-training dataset to $|M| = 100$ clients following non-IID data partitions according to a Dirichlet distribution (Morafah et al. 2022; Li, He, and Song 2021), and select $|m| = 20$ participants out of the $|M|$ clients for each FL round. Results with different $|m|$ and IID setups are reported throughout Appendix D. We adopt a standard approach commonly used in meta-learning-based research (Jamal et al. 2020; Shu et al. 2019; Park et al. 2021) for support/query splitting, where we randomly partition each client's data into 80% support and 20% query sets. See Appendix C.1-C.2 for more details.

Downstream FL task and evaluation metrics. To generate each downstream FL task, we randomly select 5 of the 20 classes from the CIFAR-100 dataset and 40 classes from the Tiny-ImageNet dataset, and distribute the corresponding data samples to a set of $|G| = 10$ clients following non-IID Dirichlet data distributions (see Appendix D for IID results). Each participant in the downstream phase utilizes 80% of its local data as training samples, while the remaining 20% is reserved for testing samples. We keep the training procedure consistent for each downstream task (see Appendix C.2 for detailed settings). For each task, we evaluate the final global model using test samples from each client $g \in G$, reporting the accuracy and variance of the accuracy distribution across the clients. We consider a total of $X = 10$ downstream tasks, and the evaluation metrics are reported as averages (with standard deviations) across the tasks.

Data distribution. Data samples are distributed to $|M| = 100$ clients for pre-training and $|G| = 10$ clients for downstream FL tasks using the corresponding dataset based on a Dirichlet(α) distribution with $\alpha = 0.5$, as done in the literature (Morafah et al. 2022; Li, He, and Song 2021).

Baselines for pre-training. We compare CoPreFL with several established FL algorithms, including (i) standard FedAvg (McMahan et al. 2017), (ii) FedMeta (Chen et al. 2018), which employs meta-learning for unseen scenarios, and (iii) q-FFL(q $>$ 0) (Li et al. 2020), designed to balance performance across clients. When applying these baselines in scenario II, in each pre-training round t, after the global model Φ^t has been constructed, we further train with 5 additional iterations on the server dataset. This extended training follows the approach in (Yang, Chen, and Shen 2023; Bian et al. 2023), where the server's data is used to further refine the global model. Similarly, we introduce a baseline called CoPreFL-SGD, which first constructs a global model according to CoPreFL and then further performs SGD iterations using server data on the global model. Finally, we also consider initializations based on (iv) conventional centralized pre-training (Nguyen et al. 2023), and popular FL algorithms like (v) SCAFFOLD (Karimireddy et al. 2019), (vi) FedDyn (Acar et al. 2021), and (vii) PerFedAvg (Fallah, Mokhtari, and Ozdaglar 2020) (see Table 3a).

[1]The full details and additional analyses can be found in the Appendix at: https://arxiv.org/abs/2402.02225

Pre-training	Downstream: Non-IID FedAvg (CIFAR-100)				Downstream: Non-IID FedAvg (Tiny-ImageNet)			
Method	**Acc** ↑	**Variance** ↓	**Lowest 10%** ↑	**Lowest 20%** ↑	**Acc** ↑	**Variance** ↓	**Lowest 10%** ↑	**Lowest 20%** ↑
FedAvg	78.96 ± 2.98	64.80 ± 3.01	62.70 ± 3.35	67.00 ± 2.95	82.94 ± 2.59	37.21 ± 2.81	68.99 ± 2.43	72.29 ± 2.61
FedMeta	82.45 ± 3.07	48.72 ± 2.84	68.97 ± 3.04	72.41 ± 3.06	81.03 ± 2.86	37.58 ± 3.00	69.44 ± 2.61	71.55 ± 2.93
q-FFL	80.01 ± 2.67	88.92 ± 3.31	64.39 ± 2.95	67.48 ± 2.67	84.11 ± 2.49	43.96 ± 2.71	73.87 ± 2.79	76.05 ± 2.61
CoPreFL	**83.29** ± 2.61	**34.69** ± 3.17	**71.58** ± 3.00	**73.20** ± 2.98	**85.23** ± 2.43	**35.40** ± 2.75	**76.77** ± 2.58	**78.46** ± 2.47

(a) Results in `scenario` I using CIFAR-100 and Tiny-ImageNet datasets.

Pre-training	Downstream: Non-IID FedAvg (CIFAR-100)				Downstream: Non-IID FedAvg (Tiny-ImageNet)			
Method	**Acc** ↑	**Variance** ↓	**Lowest 10%** ↑	**Lowest 20%** ↑	**Acc** ↑	**Variance** ↓	**Lowest 10%** ↑	**Lowest 20%** ↑
FedAvg	82.82 ± 3.17	49.00 ± 3.41	69.71 ± 3.25	72.54 ± 3.30	82.87 ± 3.19	48.16 ± 2.94	68.94 ± 3.38	72.91 ± 3.49
FedMeta	82.69 ± 3.05	48.44 ± 2.99	68.84 ± 3.14	71.82 ± 3.27	84.19 ± 2.93	49.70 ± 2.74	70.41 ± 3.16	72.63 ± 3.00
q-FFL	82.14 ± 2.76	73.10 ± 3.08	68.22 ± 3.00	70.64 ± 2.85	83.51 ± 3.05	44.22 ± 3.22	69.91 ± 2.94	73.71 ± 3.14
CoPreFL-SGD	83.63 ± 3.00	41.73 ± 2.85	69.76 ± 2.94	73.46 ± 3.09	84.30 ± 2.77	36.24 ± 3.04	72.83 ± 2.99	75.64 ± 3.18
CoPreFL	**86.63** ± 2.93	**31.58** ± 2.64	**73.05** ± 2.51	**75.82** ± 2.88	**84.72** ± 2.51	**24.80** ± 3.00	**75.84** ± 2.87	**77.31** ± 3.13

(b) Results in `scenario` II using CIFAR-100 and Tiny-ImageNet datasets.

Table 1: Performance on 10 downstream FL tasks with various non-IID FL pre-training methods. Lowest $X\%$ shows the average accuracy of clients with the lowest $X\%$ accuracy. `CoPreFL` achieves the best initialization across scenario, metric, and dataset.

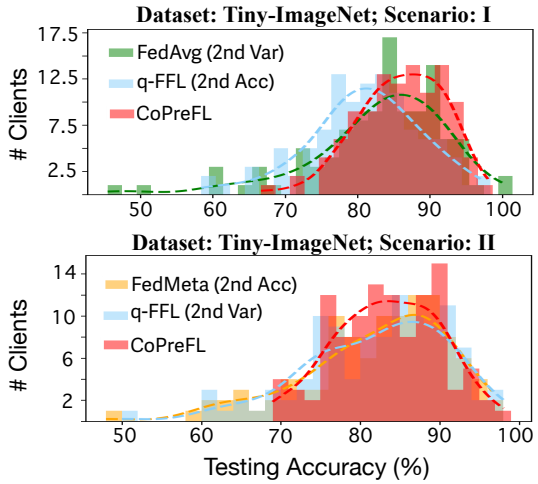

Figure 2: Accuracy distributions in non-IID FL tasks. `CoPreFL` shows higher average accuracy (i.e., right-leaning distribution) and lower performance variance (i.e., narrower distribution) while boosting the worst-performing clients.

4.2 Experimental Results

Main results for scenarios I & II. Tables 1a and 1b show test accuracies for `CoPreFL` across `scenarios` I and II on CIFAR-100 and Tiny-ImageNet datasets. In `scenario` I, `CoPreFL` shows robust initializations for downstream FL tasks, achieving higher average accurace, reduced performance variance across clients, and improved performance for the worst-performing clients (Lowest 10-20%). This highlights the benefits of balancing objectives in (3) during meta-updates. Additional analyses, including varying pre-training participants and downstream data distributions, are in Appendix D.1. In `scenario` II, `CoPreFL` also consistently outperforms baselines by effectively utilizing server data while balancing objectives (3). The benefit of a small server-side dataset, when available, can be seen through the performance gains from Table 1a to 1b. The improvement over `CoPreFL-SGD` suggests that centralized SGD with server data after meta-updating the global model may divert the pre-trained model away from our objectives. This emphasizes the importance of meta-learning on partitioned server data, as outlined in Algorithm 2. For more results including the impact of server dataset sizes, see Appendix D.2.

γ	Acc ↑	Variance ↓
0.0	83.11 ± 2.17	24.70 ± 1.95
0.25	84.04 ± 2.00	35.88 ± 2.59
0.5	85.23 ± 2.43	35.40 ± 2.75
0.75	85.19 ± 2.38	39.31 ± 2.64
1.0	86.33 ± 1.92	39.81 ± 2.30

Table 2: Effect of balancer γ in `scenario` I.

Performance distribution comparison. Figures 2 show the testing accuracy distributions of the final global model across clients in downstream tasks. We visualize `CoPreFL` with methods having the second-best average accuracy and second-lowest variance from Table 1. `CoPreFL` shows narrower distributions, indicating lower variance, and rightward shifts, reflecting higher average accuracy. Notably, `CoPreFL` effectively shifts most low-performing clients from the left tail to the right, improving their accuracy. More results for various scenarios are in Appendix D.3.

Effect of balancer γ in `CoPreFL`. Table 2 presents the performance of `CoPreFL` on Tiny-ImageNet using different balancers γ. A larger γ implies that the pre-trained model prioritizes the devices' average performance, whereas a smaller γ emphasizes performance balance. We see that increasing γ lead to higher average accuracy in downstream FL tasks but also greater variance, indicating performance imbalance. This trend shows that `CoPreFL` allows control over the relative importance between accuracy and balanced performance during pre-training.

Comparison with other initialization methods. Along with the baselines in Table 1b focusing on performance balance or meta-learning, Table 3a also evaluates popular algorithms like SCAFFOLD (Karimireddy et al. 2019), FedDyn (Acar et al. 2021), and PerFedAvg (Fallah, Mokhtari, and Ozdaglar 2020) for pre-training in `scenario` I. We see that `CoPreFL` also outperforms these baselines, validating our meta-learning approach based on (3). These popular FL algorithms struggle with unseen task heterogeneity and performance balance. We also test downstream FedAvg with

Pre-training (Scenario I)	**Downstream:** Non-IID FedAvg	
Method	**Acc** ↑	**Variance** ↓
Random Initialization	75.32 ± 1.68	41.39 ± 3.35
Centralized (Nguyen et al. 2023)	81.30 ± 2.92	69.44 ± 2.33
SCAFFOLD (Karimireddy et al. 2019)	79.15 ± 3.08	57.84 ± 1.95
FedDyn (Acar et al. 2021)	81.23 ± 2.96	53.17 ± 2.85
PerFedAvg (Fallah, Mokhtari, and Ozdaglar 2020)	81.58 ± 1.83	49.73 ± 2.65
CoPreFL	**83.29** ± 2.61	**34.69** ± 3.17

(a) Comparison with other initializations.

Pre-training (Scenario I)	**Downstream**: Non-IID FL			
	FedProx (μ = 1)		q-FFL (q = 2)	
Method	**Acc** ↑	**Variance** ↓	**Acc** ↑	**Variance** ↓
Centralized	82.39 ± 3.17	51.46 ± 2.59	79.26 ± 2.33	47.10 ± 3.05
FedAvg	79.53 ± 2.69	46.15 ± 3.04	79.53 ± 2.38	44.59 ± 2.95
FedMeta	81.77 ± 3.29	63.12 ± 3.62	79.30 ± 3.02	39.63 ± 3.17
q-FFL	83.19 ± 3.03	52.12 ± 2.97	81.38 ± 2.67	37.27 ± 2.85
CoPreFL	**84.31** ± 3.01	**30.55** ± 2.61	**82.71** ± 2.45	**25.39** ± 2.87

(b) Integration with other downstream FL algorithms.

Table 3: Results with (a) other initializations and (b) other downstream FL algorithms on CIFAR-100.

Pre-training	**Downstream:** Non-IID FedAvg			
Method	**Acc** ↑	**Var** ↓	**Lowest 10%**	**Lowest 20%**
Centralized	82.63 ± 2.63	63.57 ± 3.08	67.35 ± 2.57	69.22 ± 3.01
FedAvg	80.19 ± 1.19	51.35 ± 2.44	68.72 ± 1.63	70.15 ± 1.45
FedMeta	83.14 ± 2.07	39.85 ± 1.38	67.29 ± 2.22	71.35 ± 2.53
q-FFL	81.34 ± 1.91	47.98 ± 2.00	69.22 ± 1.85	70.35 ± 2.39
CoPreFL	**84.79** ± 1.25	**30.51** ± 1.72	**70.83** ± 1.59	**72.66** ± 1.61

Table 4: Results with both seen/unseen classes during downstream FL, using the CIFAR-100 dataset in `scenario` I.

Pre-training	**Downstream:** Non-IID FedAvg	
Method	**Acc** ↑	**Variance** ↓
Centralized	86.75 ± 2.89	67.34 ± 2.17
CoPreFL	**87.96** ± 1.95	**30.79** ± 2.79

Table 5: Results with centralized dataset where ImageNet is used for pre-training and CIFAR-100 for downstream FL.

random weights or a centrally pre-trained model, a concept introduced in (Nguyen et al. 2023). While centralized pre-training boosts downstream FL accuracy over random initialization, it introduces high performance variance due to its inability to mimic downstream FL characteristics. Additional details and results are in Appendix D.5.

Compatibility with other downstream FL algorithms. We next explore the ability of our pre-training method to enhance the performance of downstream FL algorithms other than FedAvg. For this, we consider FedProx (Sahu et al. 2018) and q-FFL (Li et al. 2020), more advanced FL algorithms that addresses heterogeneity and performance balance, in each federated downstream task. Table 3b shows the results. Overall, we see that `CoPreFL` consistently achieves superiority in accuracy and variance compared to other pre-training baselines, when combined with different downstream FL algorithms. Details on the implementation and further discussions are provided in Appendix D.6.

Both unseen/seen classes in downstream FL tasks. In addition to the setting without overlapping classes between pre-training and downstream tasks, we explore a mixed scenario where downstream clients hold "seen classes." We use CIFAR-100 and randomly sampled 10 classes from the pre-training and downstream dataset, resulting in 10 seen and 10 unseen classes. We conduct 10 non-IID FedAvg downstream tasks by randomly selecting 5 classes. Pre-trained models are solely trained on the original 80 classes of CIFAR-100. Table 4 shows higher accuracies than Table 1a, as downstream tasks include seen classes. The improvements in each metric further confirm the advantage of `CoPreFL`.

Pre-training	**Downstream:** Non-IID FedAvg	
Method	**Acc** ↑	**Variance** ↓
FedAvg	62.23 ± 2.65	51.29 ± 3.11
FedMeta	64.35 ± 3.07	44.38 ± 2.98
q-FFL	60.79 ± 3.15	27.96 ± 3.03
CoPreFL	**66.83** ± 2.85	**24.31** ± 2.83

Table 6: Results in the domain shift scenario using PACS.

Application to centrally stored public dataset. We also explore the applicability of `CoPreFL` with centrally stored pre-training data, as detailed in Remark 2. We conducted pre-training using the ImageNet_1K dataset and use FedAvg with CIFAR-100 as the downstream task. We intentionally split the dataset according to `scenario` I to mimic the distributed nature of downstream FL. Results in Table 5 show that `CoPreFL` outperforms standard centralized pre-training in both average accuracy and balanced performance, showing `CoPreFL`'s advantage even when a large public dataset is used for pre-training. Further details and additional results are available in Appendix D.7.

Comparison under domain shifts. Our experiments so far focused on the robustness of the pre-trained model to unseen labels. We now evaluate `CoPreFL` on unseen data domains using the PACS dataset. In Table 6, we use 3 domains (Art, Cartoon, and Photo) for pre-training in `scenario` I and conduct downstream FedAvg using the remaining Sketch domain, distributing samples across clients as in Table 1. Results show `CoPreFL` effectively handles domain shifts in downstream FL tasks that differ from the pre-training phase, demonstrating its robust initializations across various types of downstream data heterogeneity. Additional settings and results are in Appendix D.1.

5 Conclusion

We presented `CoPreFL`, a collaborative pre-training method that provides a robust model initialization for an arbitrary set of downstream FL tasks. `CoPreFL` leverages meta-learning to equip the pre-trained model with the ability to handle different forms of data heterogeneity that manifest in downstream FL, while balancing between average performance and variance across clients. We developed `CoPreFL` for different distributed pre-training scenarios, and showed its benefit even for centrally stored public data. Extensive experiments demonstrated the advantages of `CoPreFL` compared with several baselines methods, for a multitude of settings capturing statistical heterogeneity in downstream FL.

Acknowledgments

This work was supported by the Office of Naval Research (ONR) under grant N000142212305, the Air Force Office of Scientific Research (AFOSR) under grant FA9550-24-1-0083, and the National Science Foundation (NSF) under grant CNS-2146171.

References

Acar, D. A. E.; Zhao, Y.; Navarro, R. M.; Mattina, M.; Whatmough, P. N.; and Saligrama, V. 2021. Federated Learning Based on Dynamic Regularization. abs/2111.04263.

Bian, J.; Wang, L.; Yang, K.; Shen, C.; and Xu, J. 2023. Accelerating Hybrid Federated Learning Convergence under Partial Participation. *ArXiv*, abs/2304.05397.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T. J.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *Conference on Neural Information Processing Systems*, abs/2005.14165.

Caldas, S.; Wu, P.; Li, T.; Konecný, J.; McMahan, H. B.; Smith, V.; and Talwalkar, A. 2018. LEAF: A Benchmark for Federated Settings. *ArXiv*, abs/1812.01097.

Chang, W.-C.; Yu, F. X.; Chang, Y.-W.; Yang, Y.; and Kumar, S. 2020. Pre-training Tasks for Embedding-based Large-scale Retrieval. *ArXiv*, abs/2002.03932.

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

Chen, H.-Y.; Tu, C.-H.; Li, Z.; Shen, H. W.; and Chao, W.-L. 2023. On the Importance and Applicability of Pre-Training for Federated Learning. In *The Eleventh International Conference on Learning Representations*.

Cho, Y. J.; Jhunjhunwala, D.; Li, T.; Smith, V.; and Joshi, G. 2022. Maximizing Global Model Appeal in Federated Learning.

Chu, Y.-W.; Hosseinalipour, S.; Tenorio, E.; Cruz, L.; Douglas, K. A.; Lan, A. S.; and Brinton, C. G. 2022. Mitigating Biases in Student Performance Prediction via Attention-Based Personalized Federated Learning. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, abs/1810.04805.

Dong, J.; Wu, H.; Zhang, H.; Zhang, L.; Wang, J.; and Long, M. 2023. SimMTM: A Simple Pre-Training Framework for Masked Time-Series Modeling. *ArXiv*, abs/2302.00861.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *The Ninth International Conference on Learning Representations*, abs/2010.11929.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. E. 2020. Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. In *Neural Information Processing Systems*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Jamal, M. A.; Brown, M. A.; Yang, M.-H.; Wang, L.; and Gong, B. 2020. Rethinking Class-Balanced Methods for Long-Tailed Visual Recognition From a Domain Adaptation Perspective. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7607–7616.

Ji, S.; Pan, S.; Long, G.; Li, X.; Jiang, J.; and Huang, Z. 2019. Learning Private Neural Language Modeling with Attentive Aggregation. *The International Joint Conference on Neural Networks*, 1–8.

Jiang, Y.; Konecný, J.; Rush, K.; and Kannan, S. 2019. Improving Federated Learning Personalization via Model Agnostic Meta Learning. *ArXiv*, abs/1909.12488.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S. J.; Stich, S. U.; and Suresh, A. T. 2019. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *International Conference on Machine Learning*.

Konecný, J.; McMahan, H. B.; Yu, F. X.; Richtárik, P.; Suresh, A. T.; and Bacon, D. 2016. Federated Learning: Strategies for Improving Communication Efficiency. *ArXiv*, abs/1610.05492.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images.

Le, Y.; and Yang, X. S. 2015. Tiny ImageNet Visual Recognition Challenge.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, Broader and Artier Domain Generalization. *2017 IEEE International Conference on Computer Vision (ICCV)*, 5543–5551.

Li, Q.; He, B.; and Song, D. X. 2021. Model-Contrastive Federated Learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10708–10717.

Li, T.; Sanjabi, M.; Beirami, A.; and Smith, V. 2020. Fair Resource Allocation in Federated Learning. In *International Conference on Learning Representations*.

Li, X.; Dai, Y.; Ge, Y.; Liu, J.; Shan, Y.; and Duan, L.-Y. 2022. Uncertainty Modeling for Out-of-Distribution Generalization. *ArXiv*, abs/2202.03958.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of

Deep Networks from Decentralized Data. In *International Conference on Artificial Intelligence and Statistics*.

Mohri, M.; Sivek, G.; and Suresh, A. T. 2019. Agnostic Federated Learning. *International Conference on Machine Learning*, abs/1902.00146.

Morafah, M.; Vahidian, S.; Chen, C.; Shah, M.; and Lin, B. 2022. Rethinking Data Heterogeneity in Federated Learning: Introducing a New Notion and Standard Benchmarks. *NeurIPS 2022 workshop on Federated Learning*, abs/2209.15595.

Nguyen, J.; Wang, J.; Malik, K.; Sanjabi, M.; and Rabbat, M. 2023. Where to Begin? On the Impact of Pre-Training and Initialization in Federated Learning. In *The Eleventh International Conference on Learning Representations*.

Park, Y.; Han, D.-J.; Kim, D.-Y.; Seo, J.; and Moon, J. 2021. Few-Round Learning for Federated Learning. In *Neural Information Processing Systems*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners.

Reddi, S. J.; Charles, Z. B.; Zaheer, M.; Garrett, Z.; Rush, K.; Konecný, J.; Kumar, S.; and McMahan, H. B. 2021. Adaptive Federated Optimization. *The Ninth International Conference on Learning Representations*, abs/2003.00295.

Sahu, A. K.; Li, T.; Sanjabi, M.; Zaheer, M.; Talwalkar, A.; and Smith, V. 2018. Federated Optimization in Heterogeneous Networks.

Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; and Meng, D. 2019. Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting. In *Neural Information Processing Systems*.

Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; and Khazaeni, Y. 2020. Federated Learning with Matched Averaging. *The Eighth International Conference on Learning Representations*, abs/2002.06440.

Yang, K.; Chen, S.; and Shen, C. 2023. On the Convergence of Hybrid Server-Clients Collaborative Training. *IEEE Journal on Selected Areas in Communications*, 41: 802–819.

Yuan, H.; Mu, Z.; Xie, F.; and Lu, Z. 2024. Pre-Training Goal-based Models for Sample-Efficient Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*.

Zhang, X.; Yin, W.; Hong, M.; and Chen, T. 2020. Hybrid Federated Learning: Algorithms and Implementation. *ArXiv*, abs/2012.12420.

Zhang, X.; Zhao, Z.; Tsiligkaridis, T.; and Zitnik, M. 2022. Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency. *ArXiv*, abs/2206.08496.

Zhou, K.; Yang, Y.; Qiao, Y.; and Xiang, T. 2021. Domain Generalization with MixStyle. *ArXiv*, abs/2104.02008.