Prototype-Driven Data-Free Heterogeneous Federated Learning

Abstract

This paper proposes Data-free Heterogeneous Federated learning (DHF), a framework designed to address the challenges of model and data heterogeneity in federated learning. DHF uses Personalized Model Adaptation (PMA) to tailor models for individual clients, effectively handling diverse data distributions and model architectures. PMA enhances local model adaptability and performance through the aggregation of data prototypes on the server and a novel strategy for aggregating clients' classification heads. To maintain a robust global model, DHF employs global model enhancement, which includes an ensemble distillation technique for efficient knowledge transfer and an innovative indicator for guiding synthetic data generation. This data-free approach not only preserves privacy but also improves overall model reliability. Extensive experiments across various heterogeneous scenarios confirm DHF's superior performance, positioning it as a strong candidate for advancing federated learning in complex environments.

Keywords: federated learning, data heterogeneity.

1 Introduction

The advent of advanced information technology and pervasive Internet access has propelled us into the big data era, characterized by the massive and rapid generation, collection, storage, and analysis of data. This wealth of information has significantly fueled the development of deep learning models [2, 41]. However, it also raises substantial challenges in protecting data privacy [31, 35], making it imperative to strike a balance between data utilization and the preservation of privacy rights. Federated learning, a sophisticated machine learning paradigm, addresses these challenges by enabling models to be trained on local devices, sharing only model updates instead of raw data [6, 33]. This approach bolsters data privacy and facilitates collaborative learning across varied data sources, enhancing model generalization. Consequently, federated learning has gained traction across multiple domains [18, 29].

Nonetheless, federated learning grapples with significant challenges, particularly data heterogeneity — disparities in data distributions and attributes across sources [20, 42]. These variations impede model generalization, diminish training efficiency, and disrupt parameter consistency. To counteract these issues, various methods have been proposed, including regularization to enhance global model generalization [12, 24], contrastive learning [22], and personalized models through parameter decoupling [3, 10, 34] and meta-learning [13]. However, these approaches often assume a standardized model architecture, neglecting the issue of model heterogeneity [46].

Model heterogeneity arises from the diverse computational and communication capacities of users' devices, rendering a standardized model architecture suboptimal and potentially stifling the performance of more capable devices and servers [16]. To mitigate this, novel methods have been developed. Partial training-based approaches alleviate computational and communication demands by enabling users to train sub-models derived from the global model [1, 4, 11, 14]. However, these techniques often curtail users' autonomy by relying on server-directed sub-model extraction. In contrast, knowledge distillation methods afford users greater flexibility in constructing their model architectures [9, 17]. Yet, many of these techniques do not retain a global model on the server, which could compromise federated learning's generalization and transfer abilities [17, 39, 47]. Those that do maintain a global model often depend on public data for training [8, 27]. Although data-free alternatives have been suggested, generating high-quality synthetic data remains a challenge [40, 48].

Designing a federated framework that customizes models for individual clients while maintaining a robust global model is challenging in environments with both model and data heterogeneity. To address this, we introduce Data-Free Heterogeneous Federated Learning (DHF). DHF uses Personalized Model Adaptation (PMA) to tailor models to each client despite diverse data and model configurations. PMA regulates clients' feature extraction by aggregating data prototypes on the server and introduces a novel aggregation strategy for clients' classification heads to enhance performance. For the global model, DHF uses Global Model Enhancement (GME), which includes an ensemble distillation technique to effectively transfer knowledge from clients to the global model. Besides, it introduces an innovative indicator to guide synthetic data generation, improving the model's robustness. Extensive experiments validate DHF's effectiveness in managing both model and data heterogeneity. Our key contributions include:

- **DHF Framework**: Delivers tailored models for individual clients in diverse environments while maintaining a robust global model.
- **New Data-Free Method**: Introduces a smart, indicator-driven technique to generate high-quality synthetic data.
- **Experimental Validation**: Extensive experiments showcase DHF's effectiveness across a wide range of challenging, heterogeneous scenarios..

2 Related works

To address data heterogeneity, current approaches include generalized and personalized federated learning. Generalized federated learning aims to create a robust global model suitable for all users. Methods like FedProx [24], FedDyn [12], and SCAFFOLD [19] introduce additional mechanisms to handle data variability and improve convergence. FedProx adds a proximal term to local optimization, FedDyn incorporates dynamic control, and SCAFFOLD uses control variables to synchronize updates among participants. MOON [22] applies contrastive learning to enhance training. Knowledge distillation techniques are used in FedGen [51] and FedFTG [49]: FedGen simulates heterogeneous data through generative models, while FedFTG fine-tunes the global model to reduce the impact of data variability. Personalized federated learning focuses on customizing models for individual users. Techniques such as FedPer [3], FedRep [10], FedBABU [34], and FedRoD [7] combine local models with global knowledge while maintaining personalization. FedFomo [50] and

APPLE [30] use weighted aggregation of models from other clients for enhanced personalization. Ditto [23] and pFedMe [38] leverage multi-task learning, while Per-FedAvg [13] and FedMeta [5] apply model-agnostic meta-learning (MAML) to build an initial meta-model and perform additional local training. FedPHP [25] introduces the concept of "inHerited Private Models" (HPMs) to integrate historical personalized models into global updates, guiding personalization. FedAMP [15] proposes an attentive message-passing mechanism to support similar tasks. However, these methods often assume uniform model architectures, limiting their generalizability.

To address model heterogeneity, current approaches primarily fall into two categories: partial training and knowledge distillation. Partial training methods like Federated Dropout [4] randomly select sub-models for training, increasing model diversity but potentially causing instability due to varying model components. HeteroFL [11] achieves training stability by statically selecting sub-models, though it is constrained by the server model's size and can lead to uneven data distribution. FedRolex [1] improves upon this by rotating sub-model selection, allowing the server model to be more complex than client models, which often results in better performance compared to other partial training methods. Knowledge distillation methods such as FedDF [27], and FedET [9] leverage unlabeled public datasets for ensemble distillation. While these methods enhance model performance, they require additional data, which can be a practical challenge. FedProto [39] uses prototypes for knowledge transfer but lacks a global model, limiting its effectiveness. Methods like FML [37], PFML [45], and FedKD [43] employ mutual learning strategies, introducing shared models during local training. Lastly, DENSE [48] and DFRD [40] explore data-free ensemble distillation, but their efficiency and effectiveness in generating auxiliary data still need improvement.

3 Method

We consider a scenario where there are N clients, each with its own dataset, denoted by $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N$, where $\mathcal{D}_k = \{(x_i, y_i)|i = 1, \ldots, |\mathcal{D}_k|\}$. These datasets are heterogeneous and have non-IID characteristics. In each epoch t, a certain percentage of clients (denoted by $\mathcal{A}^t$) are randomly selected to participate. This activation percentage is denoted by the symbol ρ ($0 \leq \rho \leq 1$). Each user has his own local model ω_k, which can be divided into two parts [ϕ_k, h_k], where ϕ_k is the feature extraction part, which can map the input to a low-dimensional representation space, and h_k can map the low-dimensional space to the output space.

To address the challenges of data heterogeneity and model heterogeneity and minimize communication overhead, our approach is divided into two parts. The first part is personalized learning, which aims to customize personalized models for each user, and the architectures of these models can be different (except for the last layer of the feature extraction part). Then, we use the current knowledge to train a generator using a new evaluation metric to generate high-quality public data to assist in transferring the knowledge of each client's local model to the global model.

3.1 Personalized Learning Component

Our goal is to customize personalized models for each user when their model architecture is different to solve the problem of data heterogeneity. The goal can be defined as:

$$\min_{\{\omega_1,\cdots,\omega_N\}} \sum_{k\in\mathcal{A}^t} \frac{|\mathcal{D}_k|}{|\mathcal{D}|}\mathcal{L}_k\left(\omega_k\right) \tag{1}$$

where, $\mathcal{D} = \cup_k \mathcal{D}_k$ is the set of all client data. ω_k is the local model of client k, and the model architecture of each client can be different except for the last layer of feature extraction. $\mathcal{L}_k$ is the local loss function of client k.

3.1.1 Local Training

On the server side, we maintain a generator G tasked with producing pseudo data that mimics the distribution of the client data. The generation process is defined as $\widetilde{x} = G(z;\theta)$, where θ represents the generator's parameters, $z \sim \mathcal{N}(0,1)$ is standard Gaussian noise, and y is a one-hot label randomly generated from a uniform distribution. The objective of G is to create synthetic data $\widetilde{x}$ that closely resembles the client training data and is associated with the corresponding labels y. The training of the generator is guided by three key principles: authenticity, validity, and diversity. To fulfill these requirements, we model the training process with three distinct loss functions: 1. **Authenticity**: The generated data must not only appear realistic but also align with the knowledge embedded in the client models. We use prototypes as a reference because they are derived from model statistics and can mitigate the effects of model instability. We aim for the features extracted from the generated data by the model's feature extraction part to be as close as possible to the corresponding prototype. The authenticity loss is formulated as:

$$\mathcal{L}_{aut} = \mathcal{L}(x^j, \overline{C}^j; \widetilde{\phi}^j) \tag{2}$$

Here, $\widetilde{\phi}^j$ is the feature extraction component derived from the ensemble formula. This loss function ensures that the generated data closely mimics real data, preserving its statistical and distributional properties. By doing so, we guarantee that the generator's output is consistent with actual data at the feature level, enhancing the credibility and utility of the generated data. 2. **Validity**: Even if the data appears authentic, it must also be capable of effectively transferring knowledge. We want the generated data to expose the knowledge gaps of the student model, facilitating the transfer of knowledge. The validity loss is defined as:

$$\mathcal{L}_{trans} = \mathcal{L}_D(\omega(x^j), \widetilde{\omega}^j(x^j)) + \mathcal{L}_D(\phi(x^j), \widetilde{\phi}^j(x^j)) \tag{3}$$

This loss function encourages the student and teacher models to differ in their outputs and feature spaces, enabling the global model to learn from the client's knowledge. This design ensures that the generated data serves a practical purpose in training the student model, aiding in the capture and understanding of important features and patterns within the client data. 3. **Diversity**: To ensure the generated data encompasses the diversity of client data, we require that the generated data points exhibit clear differences across categories and avoid the generator falling into a local optimum that could lead to model collapse. The diversity loss is defined as:

$$\mathcal{L}_{div} = e^{\sum_{i,j\in\{1,\ldots,B\}}(-||\widetilde{x}_i-\widetilde{x}_j||_2 * ||z_i-z_j||_2)/B^2} \tag{4}$$

Where B is the batch size. This loss function uses the distance between the input noise Z and the generator's output $\widetilde{x}$ as a weight. If the noise Z has a large distance, we also require the generated data $\widetilde{x}_i$ to have significant differences, ensuring that the generated data has sufficient spread in the feature space, capturing the diversity

of client data. This prevents the generator from producing overly similar data samples, ensuring that the model is exposed to a rich and diverse set of data features during training. By integrating these loss functions, we can train a generator that produces realistic data, effectively transfers knowledge, and maintains diversity, providing high-quality data support for the training of the global model. This approach allows us to leverage synthetic data without direct access to client data, enhancing the overall performance and adaptability of the model.

In this study, we employ prototypes as the medium for knowledge transfer. Consequently, we introduce a localized loss function for each client, which is delineated as:

$$\mathcal{L}_k = \mathcal{L}_s(x_i, y_i; \omega_k) + \lambda \cdot \mathcal{L}_R(x_i^j, \overline{C}^j; \phi_k) \tag{5}$$

In this equation, $\mathcal{L}_s$ denotes the classification loss, where we utilize the cross-entropy loss. $\mathcal{L}_R$ serves as a regularization term to quantify the distance, and in this instance, we employ the L2 norm. The notation x_i^j signifies that the data point x_i is associated with the label j, while $\overline{C}^j$ refers to the prototype corresponding to the same label as x_i. Our objective is to ensure that the extracted features are in close proximity to their respective prototypes, thereby facilitating the absorption of global knowledge in the feature extraction phase. Additionally, each client is responsible for computing its own prototypes, which is achieved through the following computation:

$$C_k^j = \sum_{i \in \mathcal{D}_k^j} \frac{1}{|\mathcal{D}_k^j|} f_k(\phi_k; x_i) \tag{6}$$

Here, $\mathcal{D}_k^j$ denotes the dataset with label j at client k, $|\mathcal{D}_k^j|$ indicates the number of data points in the dataset, and f_k denotes the feature extraction function. As local training progresses to completion, each client will possess a C_k array whose length corresponds to the diversity of data types present.

3.1.2 Server Aggregating

To establish a comprehensive global knowledge system, we initiate the process by amalgamating the prototypes across all clients. Subsequently, we employ a weighted aggregation strategy that considers the categorical distribution within each client's dataset to construct a unified global prototype. The precise formula for this aggregation is delineated below:

$$\overline{C}^j = \sum_{k \in \mathcal{A}^t} \frac{|\mathcal{D}_k^j|}{|\mathcal{D}^j|} C_k^j \tag{7}$$

where, $|\mathcal{D}^j|$ denotes the cumulative count of data points labeled j across all participating clients. This method ensures that the contribution of each client's prototype to the global prototype is commensurate with the size of their respective datasets, effectively capturing the data abundance of various clients for specific categories.

Indeed, no matter how effectively the prototype is crafted, its influence is inherently limited to the feature extraction phase, failing to exert direct constraints on the classification layer. To enhance personalization and bridge this gap, we introduce a novel strategy for aggregating the classification heads across clients. Given that we enforce a uniform dimensionality for the final layer of the feature extraction module, the features and prototypes derived by all clients are isomorphic. We leverage this uniformity by feeding the prototype from one client into the classification head of another client, subsequently computing the classification loss based on the resultant output. The aggregation of classification heads is formalized as follows:

$$w_{k,i} = \sum_{j \in N} \frac{|\mathcal{D}_k^j|}{|\mathcal{D}_k|} \cdot \mathcal{L}_s(C_k^j, y^j; h_i) \tag{8}$$

In this expression, N denotes the set of all data categories. The weight $\frac{|\mathcal{D}_k^j|}{|\mathcal{D}_k|}$ signifies the proportion of each category within the client's dataset. For each client k, we compute the weight $w_{k,i}$ across the active client set $\mathcal{A}^t$. A lower $w_{k,i}$ value indicates a reduced classification loss, suggesting that client i provides significant assistance in classifying client k's data. To encapsulate this inter-client assistance, we employ a negative exponential function for weighting, followed by normalization:

$$w_i^{'} = \frac{e^{-\tau w_i}}{\sum_{i \in \mathcal{A}^t} e^{-\tau w_i}} \tag{9}$$

Here, τ is a regulatory parameter that governs the rate of decay in the exponential function, which we set to a value of 0.5. This allows us to construct a customized global classification head for client k:

$$h_k^{'} = \sum_{i \in \mathcal{A}^t} w_{k,i}^{'} h_i \tag{10}$$

This aggregated classification head, $h_k^{'}$, is then communicated back to client k as the initialization parameter for subsequent training rounds. This innovative approach not only fosters collaboration among clients but also drives further refinement and performance enhancement of their personalized models, ensuring a more cohesive and effective learning system.

3.2 Ensemble Distillation Phase

In the ensemble distillation phase, we address the challenge of training a robust global model amidst model heterogeneity by leveraging the technique of integrated distillation. This involves using the personalized models from each client to train a generator that produces synthetic data, which is then used to facilitate the transfer of knowledge from the clients to the global model.

The primary goal of integrated distillation is to effectively transfer the knowledge from each client's model to the global model. In this setup, each client's model serves as a "teacher," while the global model acts as the "student." To ensure the maximal retention of each client's knowledge, we define the teacher model as an ensemble of client models, weighted by the proportion of data they represent:

$$f(\widetilde{\omega}^j; x^j) = \sum_{k \in \mathcal{A}_t} \frac{|\mathcal{D}_k^j|}{|\mathcal{D}^j|} f_k(\omega_k, x^j) \tag{11}$$

Here, the ensemble weights are determined by the amount of data each client has for a particular category, ensuring that the contribution of each client's knowledge is proportional to their data representation. Similarly, we can compute the ensemble feature extraction component $\widetilde{\phi}_k$.

3.2.1 Generator Training

On the server side, we maintain a generator G whose task is to generate pseudo data to simulate the data distribution of the client. The pseudo data generation process can be expressed as $\widetilde{x} = G(z; \theta)$, where θ is the parameter of the generator, $z \sim \mathcal{N}(0, 1)$ is standard Gaussian noise, and y is a one-hot label randomly generated from a uniform distribution. The goal of the generator G is to create synthetic data $\widetilde{x}$ that is similar to the client training data and has corresponding labels y.

The training requirements of the generator include authenticity, diversity, and effectiveness. To meet these requirements, we adopt a new modeling approach that decomposes it into three different loss functions for optimization:

Authenticity. Our goal is that the generated data should not only be realistic but also consistent with the knowledge of the client model. To this end, we use the prototype as an indicator because the prototype is obtained through model statistics and can offset the impact of unstable models to a certain extent. We hope that the features extracted by the generated data in the feature extraction part of the model are as close to the prototype as possible. The loss function of authenticity can be expressed as:

$$\mathcal{L}_{aut} = \mathcal{L}(x^j, \overline{C}^j; \widetilde{\phi}^j) \tag{12}$$

Among them, $\widetilde{\phi}^j$ is the corresponding feature extraction part obtained according to the formula. This loss function makes the generated data simulate the real data as much as possible to maintain the statistical and distribution characteristics of the data. In this way, we can ensure that the data generated by the generator is consistent with the actual data at the feature level, thereby improving the credibility and application value of the generated data.

Validity. Even if the data is realistic, it is invalid if it cannot effectively transfer knowledge. We hope that the generated data can reveal the knowledge blind spots of the student model, thereby promoting the effective transfer of knowledge. The loss function for effectiveness is defined as:

$$\mathcal{L}_{trans} = -\mathcal{L}_D(\omega(x^j), \widetilde{\omega}^j(x^j)) - \mathcal{L}_D(\phi(x^j), \widetilde{\phi}^j(x^j)) \tag{13}$$

This loss function encourages the student model and the teacher model to differ in output and feature space, so that the global model can learn the client's knowledge. With this design, we can ensure that the generated data plays a practical role in training the student model, helping the model to better capture and understand the important features and patterns in the client data.

Diversity. To ensure that the generated data can cover the diversity of client data, we require that the generated data points have obvious differences between categories and avoid the generator falling into a local equilibrium that causes the model to crash. We define the diversity loss as follows:

$$\mathcal{L}_{div} = e^{\sum_{i,j\in\{1,\ldots,B\}}(-||\widetilde{x}_i - \widetilde{x}_j||_2 * ||z_i - z_j||_2)/B^2} \tag{14}$$

Where B is the batch size. This loss function uses the distance from the input to the noise Z of the generator G as a weight. If the gap of Z is large, we also require that the generated data $\widetilde{x}_i$ have greater differences to ensure that the generated data has sufficient dispersion in the feature space, so as to better capture the diversity of client data. This prevents the generator from generating overly similar data samples, thereby ensuring that the model can be exposed to rich and diverse data features during training.

So the loss function of the generator can be defined as:

$$\mathcal{L}_{gen} = \mathcal{L}_{aut} + \lambda_{div}\mathcal{L}_{div} + \lambda_{eff}\mathcal{L}_{eff} \tag{15}$$

λ_{div} and λ_{eff} control the weight of diversity and validity of the generated public data. By combining these loss functions, we can train a generator that can generate realistic data, effectively transfer knowledge, and has

diversity, providing high-quality data support for the training of the global model. In this way, we can make full use of the synthetic data generated by the generator without directly accessing the client data, and improve the overall performance and adaptability of the model.

3.3 Training of the global model

After successfully training the generator to ensure that it can produce data that meets our established standards, we began the process of knowledge transfer. The core goal of this step is to ensure that the global model can fully absorb and learn the valuable knowledge from each client. To achieve this, we adopted a strategy that forces the global model to maintain a close proximity to the teacher model both in the output layer and the feature space layer. The mathematical expression of this strategy can be defined as follows:

$$\mathcal{L}_{glo} = \mathcal{L}_D(\omega(x^j), \widetilde{\omega}^j(x^j)) + \alpha \mathcal{L}_D(\phi(x^j), \widetilde{\phi}^j(x^j)) \tag{16}$$

Here, $\mathcal{L}_D$ represents a distance metric that measures the difference between the output of the global model ω and the output of the ensemble teacher model $\widetilde{\omega}^j$, as well as the difference between the feature extractor of the global model ϕ and the feature extractor of the ensemble teacher model $\widetilde{\phi}^j$ in feature space. The parameter α is an adjustable weight coefficient used to balance the importance of these two parts of the loss. As can be seen from the design of the formula, we do not impose any specific restrictions on the architecture of the global model. This means that the global model can be completely different in structure from any client model, and can even be designed to be more complex and large. This flexibility in design brings us many advantages. First, the global model can better integrate and refine heterogeneous knowledge from multiple clients, thereby obtaining a richer representation at the global level. Second, since the global model is not limited to the size or structure of a single client model, it can accommodate more parameters, thus having the potential to learn more complex patterns and improve its performance on various tasks.

In short, through a carefully designed knowledge transfer strategy and a flexible architecture of the global model, we can not only effectively integrate heterogeneous knowledge from various clients, but also build a more powerful and adaptable global model, which is of great significance to improving the overall performance of the federated learning system.

4 Theoretical Analysis

Definition 1 (Client Model and Data Distribution) Let $\mathcal{D}_i$ be the data distribution of client i, and $f_i(\cdot)$ be the local model trained on the data from $\mathcal{D}_i$. For each client i, let $x \sim \mathcal{D}_i$ denote an input data sample, and y_i be the corresponding label.

Definition 2 (Generator and Generated Data) Consider a generator G that maps noise z from a noise distribution $\mathcal{N}$ to a data space. The generated data is denoted as $\hat{x} = G(z)$.

Definition 3 (Prototype) The prototype for client i is a statistical representation of the data distribution $\mathcal{D}_i$ obtained by applying a transformation function T on the outputs of the local model $f_i(\cdot)$. Let P_i denote the prototype for client i, given by:

$$P_i = T(f_i(\mathcal{D}_i))$$

Assumption 1 (Unreliable Client Models) Assume that the local models $f_i(\cdot)$ of clients are noisy and may not accurately capture the true data distribution $\mathcal{D}_i$. This unreliability is modeled as:

$$f_i(x) = y_i + \epsilon_i$$

where ϵ_i is the noise term associated with client i.

Problem Formulation The goal is to generate data $\hat{x}$ that closely resembles the true data distribution $\mathcal{D}_i$ for each client i. Traditional methods use the label y_i to measure the similarity between $f_i(\hat{x})$ and y_i. We propose using the prototype P_i instead of the label y_i.

Traditional Method (Label-based) In traditional methods, the objective is to minimize the difference between the model output for generated data and the true labels:

$$\min_G \mathbb{E}_{z\sim\mathcal{N}} \left[\sum_{i=1}^{N} \|f_i(G(z)) - y_i\|^2 \right]$$

Proposed Method (Prototype-based) In the proposed method, the objective is to minimize the difference between the feature representation of the generated data and the prototype:

$$\min_G \mathbb{E}_{z\sim\mathcal{N}} \left[\sum_{i=1}^{N} \|T(f_i(G(z))) - P_i\|^2 \right]$$

Theorem 1 (Effectiveness of Prototype-based Method) **Theorem:** The prototype-based method provides a more reliable measure of similarity between the generated data and the true data distribution compared to the label-based method, given the unreliability of client models.

Proof: 1. **Noise Reduction:** The prototype P_i is a statistical aggregate, reducing the impact of noise ϵ_i. Thus, the error term in the prototype-based method is less affected by model unreliability:

$$P_i = T(f_i(\mathcal{D}_i)) = T(y_i + \epsilon_i) \approx T(y_i)$$

assuming T is a robust aggregation function.

2. **Consistency:** Since P_i represents the aggregated behavior of the client model over the data distribution, it is more consistent than individual labels:

$$\|T(f_i(G(z))) - P_i\|^2 \leq \|f_i(G(z)) - y_i\|^2 + \mathcal{O}(\epsilon_i)$$

3. **Robustness:** The prototype-based method inherently mitigates the impact of unreliable models by focusing on the overall data distribution rather than individual noisy outputs:

$$\min_G \mathbb{E}_{z\sim\mathcal{N}} \left[\sum_{i=1}^{N} \|T(f_i(G(z))) - P_i\|^2 \right]$$

is less sensitive to noise compared to:

$$\min_G \mathbb{E}_{z\sim\mathcal{N}} \left[\sum_{i=1}^{N} \|f_i(G(z)) - y_i\|^2 \right]$$

5 Implementation details

Datasets. We assess DHF through extensive experiments on three benchmark datasets: FMNIST [44], CIFAR 10, and CIFAR 100 [21]. We evaluate our approach in two distinct heterogeneous scenarios. The first, termed the pathological heterogeneous settingcite [36], involves allocating 2/2/20 classes from a total of 10/10/100 classes for FMNIST/CIFAR 10/CIFAR 100 to each client, ensuring disjoint data samples. The second scenario, known as the practical heterogeneous setting, is governed by the Dirichlet distribution (Dir(γ)) [28], where a smaller γ value indicates a more heterogeneous distribution. Our default setting, $\gamma = 0.1$, represents a practical and controlled level of heterogeneity for our experiments.

Model. In the case of homogeneous models, to ensure fairness, we utilized 4-layer CNNs for all methods. For heterogeneous models, we set up two different frameworks for clients: a shallow 4-layer CNN network and a deep 5-layer CNN. The proportion of clients using the deeper model in the federated framework is represented by ν. If ν is 1, it means all clients are using the deeper model. For the global model, we used a relatively larger ResNet-10. Although computational limitations prevented us from setting a larger global model, this still demonstrates our framework's capability to accommodate larger models.

Baselines. In this section, we undertake a thorough comparison between DHF and a range of federated learning baselines. For the homogeneous model setting, we compare our approach against FedAvg [32], SCAFFOLD [19], FedProx [24], MOON [22], FedGen [51], Per-FedAvg [13], Ditto [23], FedFomo [50], FedAMP [15], FedPHP [25], FedBABU [34], FedRoD [7], and FedProto [39]. For the heterogeneous model setting, we compare against FedDistill [17], FML [37], LG-FedAvg [26], FedGen [51], FedProto [39], FedKD [43], FedTGP [47], FedFTG [49], and Dense [48]. Some heterogeneous model methods do not have a global model, so these comparisons will be divided into two parts: one for personalization and one for global model comparison.

Implementation Details. We using a batch size of 10 and a single epoch for local model training, consistent with FedAvg. Our experiments are conducted over 1000 iterations for convergence, with 10 clients and a default ρ=1 parameter setting. We use evaluation metrics to report the test accuracy of the best single global model for traditional FL and the average test accuracy of the best local models for personalized federated leraning. Evaluations are conducted on the client side, with around 25% of local data forming the test dataset and the remaining 75% used for training.

6 Results and analysis

6.1 Performance in Heterogeneous Scenarios

6.1.1 Model Homogeneity:

We first tested the performance of personalized models under model homogeneity conditions. Our experimental results are shown in Table 1. The top five rows display the accuracy of generalized methods. It is evident that their result significantly diminishes compared to personalized methods. This discrepancy arises because generalized methods struggle with challenges in training a globally robust model with strong generalization capabilities in the face of data heterogeneity, often yielding suboptimal results.

Settings	Pathological heterogeneous setting				Practical heterogeneous setting			
Method	FMNIST	CIFAR 10	CIFAR 100	Tiny-ImageNet	FMNIST	CIFAR 10	CIFAR 100	Tiny-ImageNet
FedAvg	84.70 ± 0.13	62.01 ± 0.22	30.44 ± 0.32	14.11 ± 0.18	86.19 ± 0.14	63.30 ± 0.14	32.97 ± 0.15	17.79 ± 0.13
SCAFFOLD	79.44 ± 1.10	58.22 ± 0.25	30.48 ± 0.26	14.95 ± 0.27	87.12 ± 0.17	62.95 ± 0.44	34.57 ± 0.14	19.58 ± 0.03
FedProx	84.88 ± 0.19	62.06 ± 0.31	30.57 ± 0.25	14.20 ± 0.31	86.17 ± 0.14	63.32 ± 0.13	32.93 ± 0.13	17.84 ± 0.08
MOON	84.73 ± 0.29	62.23 ± 0.27	30.58 ± 0.32	14.33 ± 0.40	86.34 ± 0.11	63.22 ± 0.12	32.85 ± 0.05	17.73 ± 0.05
FedGen	86.29 ± 0.10	63.30 ± 0.07	28.58 ± 0.27	12.71 ± 0.23	87.20 ± 0.08	63.59 ± 0.35	31.43 ± 0.27	15.67 ± 0.38
Per-FedAvg	99.52 ± 0.01	91.77 ± 0.04	54.79 ± 0.20	31.42 ± 0.42	98.67 ± 0.06	88.44 ± 0.18	43.16 ± 0.41	25.69 ± 0.29
FedFomo	99.57 ± 0.01	91.05 ± 0.08	50.38 ± 0.68	30.78 ± 0.15	98.92 ± 0.01	88.21 ± 0.07	46.18 ± 0.26	29.89 ± 0.30
FedRoD	99.64 ± 0.02	91.64 ± 0.07	53.79 ± 0.15	35.37 ± 0.17	99.01 ± 0.01	89.16 ± 0.14	51.57 ± 0.37	39.20 ± 0.05
FedBABU	99.48 ± 0.09	91.59 ± 0.03	52.04 ± 0.32	32.32 ± 0.27	99.09 ± 0.01	89.60 ± 0.25	50.72 ± 0.28	33.90 ± 0.17
Ditto	99.58 ± 0.01	90.85 ± 0.10	51.77 ± 0.16	31.73 ± 0.31	99.00 ± 0.01	88.67 ± 0.19	49.28 ± 0.20	32.62 ± 0.19
FedAMP	99.58 ± 0.01	90.68 ± 0.20	51.65 ± 0.19	31.79 ± 0.18	99.00 ± 0.01	88.92 ± 0.08	48.88 ± 0.25	32.63 ± 0.11
FedPHP	99.65 ± 0.01	91.64 ± 0.15	54.30 ± 0.24	31.54 ± 0.79	99.09 ± 0.04	88.92 ± 0.12	48.94 ± 1.23	29.11 ± 0.68
FedProto	99.63 ± 0.01	92.13 ± 0.06	55.91 ± 0.38	32.93 ± 0.18	99.04 ± 0.05	89.74 ± 0.05	49.34 ± 0.18	32.41 ± 1.30
Our	**99.66 ± 0.01**	**92.26 ± 0.06**	**56.32 ± 0.12**	**34.20 ± 0.17**	**99.19 ± 0.01**	**89.95 ± 0.11**	**53.28 ± 0.28**	**34.33 ± 0.28**

Table 1. The test accuracy (%) in the pathological heterogeneous setting and practical heterogeneous setting under model homogeneity. **Bold** /underline fonts highlight the best/second best baseline.

Settings	Pathological heterogeneous setting				Practical heterogeneous setting			
Method	FMNIST	CIFAR 10	CIFAR 100	Tiny-ImageNet	FMNIST	CIFAR 10	CIFAR 100	Tiny-ImageNet
FedDistill	99.61 ± 0.02	91.35 ± 0.19	54.31 ± 0.58	33.10 ± 0.49	99.16 ± 0.02	89.02 ± 0.24	51.16 ± 0.54	34.56 ± 0.23
FML	99.62 ± 0.01	91.31 ± 0.14	50.30 ± 0.58	32.05 ± 0.15	99.03 ± 0.02	89.12 ± 0.07	47.70 ± 0.50	33.58 ± 0.10
LG-FedAvg	99.57 ± 0.02	91.36 ± 0.21	51.82 ± 0.50	32.64 ± 0.41	99.02 ± 0.02	89.19 ± 0.10	49.15 ± 0.28	33.48 ± 0.12
FedGen	99.58 ± 0.02	91.40 ± 0.21	52.23 ± 0.42	31.56 ± 0.29	99.03 ± 0.04	89.14 ± 0.06	49.31 ± 0.10	32.11 ± 0.23
FedKD	99.63 ± 0.01	92.36 ± 0.15	57.72 ± 0.08	36.55 ± 0.11	99.16 ± 0.02	**90.17 ± 0.06**	54.59 ± 0.25	37.09 ± 0.23
FedProto	99.58 ± 0.07	**92.59 ± 0.09**	56.49 ± 0.27	33.65 ± 0.70	98.97 ± 0.02	89.70 ± 0.34	47.71 ± 0.19	29.43 ± 0.34
FedGH	99.61 ± 0.02	91.37 ± 0.24	54.38 ± 0.12	30.87 ± 0.42	99.04 ± 0.03	89.09 ± 0.18	51.34 ± 0.30	33.16 ± 0.39
FedTGP	99.52 ± 0.05	92.25 ± 0.30	55.54 ± 1.21	35.30 ± 0.52	99.11 ± 0.05	89.69 ± 0.25	53.90 ± 0.50	34.75 ± 0.26
Our	**99.63 ± 0.01**	92.45 ± 0.09	**57.80 ± 0.16**	**35.67 ± 0.48**	**99.18 ± 0.03**	90.16 ± 0.18	**54.64 ± 0.14**	**35.94 ± 0.21**

Table 2. The test accuracy (%) in the pathological heterogeneous setting and practical heterogeneous setting under model heterogeneity. **Bold** /underline fonts highlight the best/second best baseline.

Settings	Pathological setting		Practical setting	
Method	CIFAR 10	CIFAR 100	CIFAR 10	CIFAR 100
FedFTG	12.59 ± 0.09	1.16 ± 0.02	13.09 ± 0.69	1.21 ± 0.01
Dense	15.64 ± 0.38	2.78 ± 0.05	14.38 ± 0.11	4.50 ± 0.10
Our	**21.06 ± 1.19**	**2.54 ± 0.22**	**20.77 ± 0.15**	**4.66 ± 0.61**

Table 3. The test accuracy (%) of the global model in the pathological heterogeneous setting and practical heterogeneous setting under model heterogeneity. **Bold** fonts highlight the best baseline.

For different data partitioning schemes, Table 1 highlights a notable distinction: the generalized methods (top part) exhibit superior performance in the practical heterogeneous setting compared to the pathological heterogeneous setting, while personalized methods (bottom part) show the opposite trend. This discrepancy arises from the fact that in a pathological heterogeneous setting, where each client's data distribution is relatively uniform, generalized methods encounter challenges in consolidating limited knowledge from diverse models into a robust global model. Conversely, personalized methods, with their capacity for personalized modules, can effectively adapt to such uniform distributions. This trend reverses in the practical heterogeneous setting.

Notably, our method consistently achieves the best results across all data and partitioning schemes. Compared to the FedProto method, which also uses prototypes, our approach outperforms it by nearly 4% in some cases.

6.1.2 Model Heterogeneity:

When faced with model heterogeneity, many previously compared methods fail to adapt, as they were developed under the ideal condition of model homogeneity. Table 2 presents our results, showing that our approach achieves the best performance under most conditions. Among the compared methods, FedKD comes closest to our results. However, it relies on training a mentor model and a mentee model locally, and requires transferring the mentee model, leading to significantly higher computational and communication overhead. In summary, our method demonstrates a clear advantage.

Table 3 presents the results of our global model. Although FML and FedKD can also train global models under model heterogeneity, their global models are further trained on local data, making a direct comparison with our data-free approach unfair. Therefore, we compared our method with two other data-free approaches and achieved the best results.

6.2 Communication Efficiency

Table 4 evaluates various FL methods based on the number of communication rounds required to attain the target test accuracy in data heterogeneous settings. It's worth mentioning that FedBABU is omitted from the comparison due to its accuracy improvement primarily occurring at the end of head training, making it less relevant as a reference. Generalized methods are also excluded as they fail to reach the target accuracy.

Although our method does not have the fastest convergence speed, it is important to note that our communication overhead per round is significantly smaller. The model used in our approach has a data size of approximately 2.25M. In contrast, methods like Ditto, FedAMP, FedPHP, FedRoD, PerAvg, and FedFomo require communication at the full model level, whereas our method only requires 0.01M per round. Overall, our

Method	CIFAR 10		CIFAR 100	
Target	acc = 80%	acc = 85%	acc = 40%	acc = 45%
DITTO	6.60±0.80	15.40±1.50	21.00±1.26	33.80±0.98
FedAMP	2.80±0.40	7.20±0.75	9.60±0.49	16.00±0.63
FedFomo	3.00±0.00	9.00±0.00	12.20±0.40	43.60±0.49
FedPHP	411.60±5.00	663.00±14.75	268.80±8.49	537.60±19.07
FedProto	18.20±11.57	35.40±20.62	25.60±7.96	40.00±15.68
FedRoD	7.20±0.40	21.20±0.98	17.40±0.49	26.00±1.10
PerAvg	41.20±3.43	77.00±0.00	193.20±28.78	N/A
Our	5.40±0.49	15.40±1.02	18.80±0.75	26.60±0.49

Table 4. Evaluation of different FL methods on CIFAR10 and CIFAR100 (in the practical heterogeneous setting), in terms of the number of communication rounds to reach target test accuracy (acc).

method incurs much lower communication overhead compared to these approaches. While FedProto requires only about 0.005M per round and converges faster than our method, this difference is due to the additional specialized aggregation we perform on the model's classification. However, as shown in the previous experimental results, this approach proves effective. In summary, the communication overhead of our method is both efficient and acceptable.

7 Conclusion and future work

This study presents a novel approach to federated learning that addresses model and data heterogeneity through a dual-component framework: personalized model adaptation and global model enhancement. By leveraging prototypes for knowledge transfer and using synthetic data generated through ensemble distillation, we enhance the adaptability and performance of the global model. This method ensures effective integration of diverse client knowledge, improves model robustness, and maintains high training efficiency.

References

[1] Samiul Alam, Luyang Liu, Ming Yan, and Mi Zhang. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *Advances in neural information processing systems*, 35:29677–29690, 2022.

[2] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7383–7390, 2020.

[3] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers, 2019.

[4] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018.

[5] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

[6] Haokun Chen, Yao Zhang, Denis Krompass, Jindong Gu, and Volker Tresp. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11285–11293, 2024.

[7] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2021.

[8] Sijie Cheng, Jingwen Wu, Yanghua Xiao, and Yang Liu. Fedgems: Federated learning of larger server models via selective knowledge fusion. *arXiv preprint arXiv:2110.11027*, 2021.

[9] Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. Heterogeneous ensemble knowledge transfer for training large models in federated learning. *arXiv preprint arXiv:2204.12703*, 2022.

[10] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pages 2089–2099. PMLR, 2021.

[11] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*, 2020.

[12] Alp Emre Durmus, Zhao Yue, Matas Ramon, Mattina Matthew, Whatmough Paul, and Saligrama Venkatesh. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

[13] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

[14] Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, and Nicholas Lane. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *Advances in Neural Information Processing Systems*, 34:12876–12889, 2021.

[15] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 7865–7873, 2021.

[16] Fatih Ilhan, Gong Su, and Ling Liu. Scalefl: Resource-adaptive federated learning with heterogeneous clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24532–24541, 2023.

[17] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

[18] Meirui Jiang, Zirui Wang, and Qi Dou. Harmofl: Harmonizing local and global drifts in federated learning on heterogeneous medical images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1087–1095, 2022.

[19] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

[20] Taehyeon Kim, Eric Lin, Junu Lee, Christian Lau, and Vaikkunth Mugunthan. Navigating data heterogeneity in federated learning: A semi-supervised approach for object detection. *Advances in Neural Information Processing Systems*, 36, 2024.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[22] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021.

[23] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

[24] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[25] Xin-Chun Li, De-Chuan Zhan, Yunfeng Shao, Bingshuai Li, and Shaoming Song. Fedphp: Federated personalization with inherited private models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 587–602. Springer, 2021.

[26] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

[27] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

[28] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

[29] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1013–1023, 2021.

[30] Jun Luo and Shandong Wu. Adapt to adaptation: Learning personalization for cross-silo federated learning. In *IJCAI: proceedings of the conference*, volume 2022, page 2166. NIH Public Access, 2022.

[31] Zhihan Lv, Liang Qiao, M Shamim Hossain, and Bong Jun Choi. Analysis of using blockchain to protect the privacy of drone big data. *IEEE network*, 35(1):44–49, 2021.

[32] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[33] Jiaxu Miao, Zongxin Yang, Leilei Fan, and Yi Yang. Fedseg: Class-heterogeneous federated learning for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8042–8052, 2023.

[34] Jaehoon Oh, SangMook Kim, and Se-Young Yun. Fedbabu: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*, 2021.

[35] M Ileas Pramanik, Raymond YK Lau, Md Sakir Hossain, Md Mizanur Rahoman, Sumon Kumar Debnath, Md Golam Rashed, and Md Zasim Uddin. Privacy preserving big data analytics: A critical analysis of state-of-the-art. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11(1):e1387, 2021.

[36] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pages 9489–9502. PMLR, 2021.

[37] Tao Shen, Jie Zhang, Xinkang Jia, Fengda Zhang, Gang Huang, Pan Zhou, Kun Kuang, Fei Wu, and Chao Wu. Federated mutual learning. *arXiv preprint arXiv:2006.16765*, 2020.

[38] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in neural information processing systems*, 33:21394–21405, 2020.

[39] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8432–8440, 2022.

[40] Shuai Wang, Yexuan Fu, Xiang Li, Yunshi Lan, Ming Gao, et al. Dfrd: Data-free robustness distillation for heterogeneous federated learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[41] Wenya Wang and Sinno Jialin Pan. Integrating deep learning with logic fusion for information extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9225–9232, 2020.

[42] Yuan Wang, Huazhu Fu, Renuga Kanagavelu, Qingsong Wei, Yong Liu, and Rick Siow Mong Goh. An aggregation-free federated learning for tackling data heterogeneity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26233–26242, 2024.

[43] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature communications*, 13(1):2032, 2022.

[44] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[45] Ruihong Yang, Junchao Tian, and Yu Zhang. Regularized mutual learning for personalized federated learning. In *Asian Conference on Machine Learning*, pages 1521–1536. PMLR, 2021.

[46] Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.

[47] Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16768–16776, 2024.

[48] Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. Dense: Data-free one-shot federated learning. *Advances in Neural Information Processing Systems*, 35:21414–21428, 2022.

[49] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10174–10183, 2022.

[50] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2020.

[51] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International conference on machine learning*, pages 12878–12889. PMLR, 2021.