FedORGP: Guiding Heterogeneous Federated Learning with Orthogonality Regularization on Global Prototypes

Fucheng Guo[*], Zeyu Luan[†], Qing Li[‡], Dan Zhao[§], Yong Jiang[¶]

Abstract

Federated Learning (FL) has emerged as an essential framework for distributed machine learning, especially with its potential for privacy-preserving data processing. However, existing FL frameworks struggle to address statistical and model heterogeneity, which severely impacts model performance. While Heterogeneous Federated Learning (HtFL) introduces prototype-based strategies to address the challenges, current approaches face limitations in achieving optimal separation of prototypes. This paper presents FedORGP, a novel HtFL algorithm designed to improve global prototype separation through orthogonality regularization, which not only encourages intra-class prototype similarity but also significantly expands the inter-class angular separation. With the guidance of the global prototype, each client keeps its embeddings aligned with the corresponding prototype in the feature space, promoting directional independence that integrates seamlessly with the cross-entropy (CE) loss. We provide theoretical proof of FedORGP's convergence under non-convex conditions. Extensive experiments demonstrate that FedORGP outperforms seven state-of-the-art baselines, achieving up to 10.12% accuracy improvement in scenarios where statistical and model heterogeneity coexist. Our code is in the supplementary material and will be made publicly available.

1. Introduction

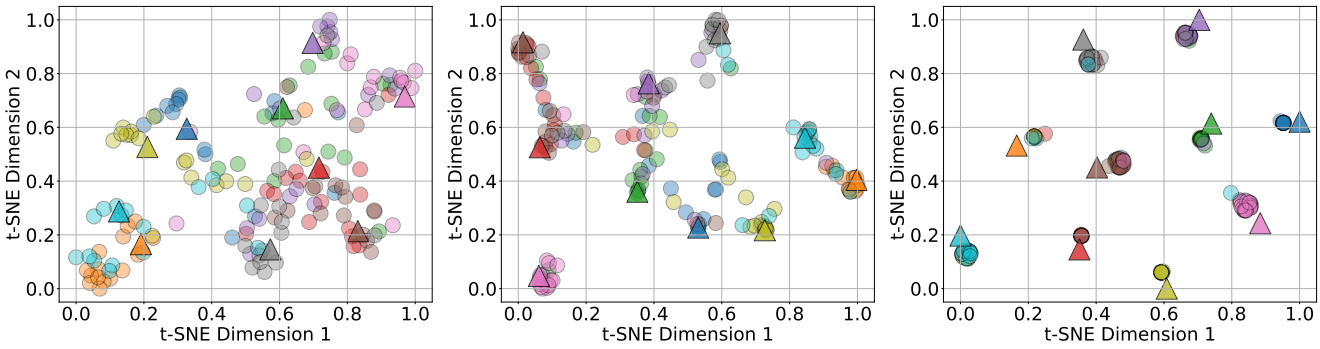
In recent years, Federated Learning (FL) has emerged as a promising distributed machine learning paradigm [7]. FL eliminates the need for clients to exchange data, allowing data to remain decentralized, which makes it an favorable solution to data privacy challenges [1, 25].

Previous work mainly focuses on homogeneous FL, which assumes that all client models are identical. However, in large-scale real-world scenarios, considerable disparities exist in both client data distribution, often termed statistical heterogeneity, and hardware resources, known as system heterogeneity [18, 41]. Data heterogeneity severely impacts the convergence and performance of global model [17, 24]. Furthermore, discrepancies in hardware resources can result in model heterogeneity across clients, posing significant challenges to conventional FL approaches that rely on aggregating model parameters [13, 41]. In addition, homogeneous FL trains a shared global model by exchanging gradients, which further imposes significant communication costs as well as privacy exposure risks [36, 44].

To tackle these challenges, Heterogeneous Federated Learning (HtFL) has emerged as a novel FL paradigm capable of handling both data heterogeneity and model heterogeneity simultaneously [32, 42, 44]. HtFL incorporates prototype-based learning, which communicates client prototypes rather than model gradients to the server, thus alleviating issues related to model diversity and communication costs. However, existing weighted average HtFL solutions like FedProto [32] faces several limitations. First, aggregating global prototypes on the server side via weighted averaging requires clients to upload sample sizes, which may lead to leakage of data distribution information [42]. Second, as shown in Fig.1a, the weighted average prototype lacks a well-defined decision boundary, resulting in overlapping feature distributions among different classes in the feature space [44].

While the recent HtFL solution FedTGP [44] has successfully improved separation by increasing the Euclidean distance between prototypes, this approach still faces several limitations. First, contrastive learning, which works together with cross-entropy (CE) loss, primarily reduces the intra-class distance between prototypes but fails to explicitly enforce inter-class separation. Second, as shown in Fig.1b, in high-dimensional spaces, the Euclidean distance makes it difficult to effectively distinguish different samples as distance variations between samples diminish [34]. Third, augmenting Euclidean distances fails to effectively leverage the angular separation characteristics inherent in CE loss [27].

[*]Shenzhen International Graduate School, Tsinghua University, gfc23@mails.tsinghua.edu.cn
[†]Peng Cheng Laboratory, China
[‡]Peng Cheng Laboratory, China, liq@pcl.ac.cn
[§]Peng Cheng Laboratory, China, zhaod01@pcl.ac.cn
[¶]Shenzhen International Graduate School, Tsinghua University, liq@pcl.ac.cn



(a) FedProto (b) FedTGP (c) FedORGP

Figure 1. We train models on the CIFAR-10 dataset and use t-SNE [35] to visualize their performance on previously unseen test samples (16 per class) within the feature space, with triangles indicating prototypes and circles denoting samples. The results indicate that FedProto [32] exhibits weak feature separation. FedTGP [44] increases prototype margin but still lacks sufficient distinction in the feature space. In contrast, our FedORGP can reduce inter-sample similarity and increase intra-class compactness, thereby effectively classifying the majority of samples. This result suggests that FedORGP exhibits robust generalization performance under conditions of both statistical and model heterogeneity.

To address these limitations, we propose a novel HtFL algorithm called FedORGP, which optimizes global prototypes through orthogonality regularization. FedORGP improves intra-class prototype similarity while preserving semantic integrity and explicitly promotes angular separation between classes, ensuring that global prototypes achieve maximal directional independence in the feature space. Under the guidance of global prototypes, as demonstrated in Fig.1c, clients can ensure that the feature representations are directionally aligned with their corresponding global prototypes in the feature space, thereby facilitating a more effective integration with the angular properties inherent to CE loss.

We evaluate FedORGP against seven state-of-the-art HtFL methods across four datasets in scenarios where both data heterogeneity (practical distribution and Dirichlet distribution) and model heterogeneity (four levels of model heterogeneity) coexist. The experimental results show an improvement in accuracy by 10.12% over the best baseline under both statistical and model heterogeneity. Our contributions can be summarized as follows:

- We present a novel framework FedORGP for HtFL with orthogonality regularization. To the best of my knowledge, this is the first work to introduce orthogonality regularization into heterogeneous federated learning where data and model heterogeneity coexist.
- We observe that increasing the Euclidean distance between prototypes is incapable of effectively integrating with the angular characteristics inherent to CE loss. Our proposed FedORGP algorithm enhances intra-class clustering within the feature space while explicitly promoting angular separation between classes.
- We offer a theoretical convergence guarantee for FedORGP and rigorously establish the convergence rate under non-convex conditions.

2. Related Work

2.1. Heterogenous Federated Learning

Heterogeneous Federated Learning (HtFL) has emerged as a solution that supports both model and statistical heterogeneity simultaneously, while ensuring user privacy. Early HtFL solutions [5, 10, 38] let clients sample different submodels from a shared global architecture, thereby reducing the computational burden on low-resource devices. However, these methods necessitate a common architecture to be shared across clients, raising privacy concerns regarding clients' model architectures.

An alternative approach to system heterogeneity shares a common model of the same structure at the server side, e.g. a global header [42] or a global generator [45], as a way to transfer global knowledge. LG-FedAvg [20] and FedGH [42] require clients to share the same top layer while allowing them to have different architectures in their lower layers. However, sharing and aggregation of top layers may result in suboptimal performance due to statistical heterogeneity [19, 23]. FedGen [45] is designed to mitigate model drift in heterogeneous FL environments by using a generative model to extract global knowledge from local models. While this method avoids reliance on proxy datasets, its performance highly relies on the quality of the generator [44]. Despite allowing some level of flexibility, they still assume that clients share partially similar models.

Alternatively, another HtFL approach aims to enable completely independent client model architectures by exchanging various forms of information, such as knowl-

edge or prototypes. FedMD [16] and FedDF [21] facilitate knowledge transfer between participants through distillation over public datasets. However, ideal public datasets are often difficult to access [43]. FML [29] and FedKD [39] train and share a small auxiliary model using mutual distillation, circumventing the need for a global dataset [44]. Nonetheless, the bi-directional distillation process relies on frequent communication between clients and the server, which can result in significant communication overhead.

Prototype-based HtFL methods [32, 33, 40] have shown promise in addressing both model and statistical heterogeneity. These approaches not only enhance model convergence in non-IID settings but also substantially reduce communication overhead by transmitting prototypes instead of full models [33]. However, these methods rely on simple weighted averages of client prototypes, which increase the chance of overlap between global prototypes in the feature space. While FedTGP [44] increases prototype separation by contrastive learning without using weighted averages, boosting the Euclidean distance between embeddings does not integrate well with CE loss [27, 34]. In this work, we introduce orthogonal regularization to explicitly enhance the angular margin among global prototypes, thereby tackling data and model heterogeneity in HtFL.

2.2. Prototype Learning

Prototype refers to the average of features representations of the same class. FedNH [4] addresses data heterogeneity by leveraging uniformity and semantics of class prototypes. FPL [12] and FedPLVM [37] mainly focus on solving the domain drift problem using prototype clustering. However, they all require model aggregation, which necessitates that client models share the same architecture, implying that they cannot work in scenarios with model heterogeneity. Our proposed FedORGP encourages prototype separation to establish distinct margins among prototypes, thereby guiding heterogeneous client models to assimilate global knowledge.

2.3. Orthogonality

Orthogonality typically describes the directional independence between two vectors. In high-dimensional space, two vectors are considered orthogonal if their dot product is zero [27]. FediOS [6] effectively separates generic and personalized features using orthogonal projections to address feature skew in personalized federated learning. However, it requires all client models to have identical architectures. C-FSCIL [9], OPL [27] and POP [22] enhance feature separation between classes via orthogonality regularization. However, they are only applicable to the optimisation of centralised training scenarios and are not applicable to the optimisation of federated scenarios. FOT [2] and FedSOL [15] are primarily concerned with the orthogonality of update directions to minimise interference across clients or tasks. However, neither of them can guarantee the orthogonality of the samples in the feature space.

3. Methodology

3.1. Problem Definition

We consider a problem involving M clients, which have heterogeneous models and private data. The local model of client k, $k \in \{1, \dots, M\}$, is split into two components: a feature extractor f_k parameterized by ϕ_k, and a classifier h_k parameterized by θ_k. Given a sample pair (x, y), the feature extractor f_k transforms $x \in \mathbb{R}^D$ into a feature representation $r = f_k(x; \phi_k)$, $r \in \mathbb{R}^K$, where D is the dimension of input and K is the dimension of feature representation ($K \ll D$). The classifier h_k maps the feature vector r to logits $\in \mathbb{R}^C$, where $logits = h_k(r; \theta_k)$ and C denotes the total number of classes. The client model's parameters are defined as $\omega_k = (\phi_k, \theta_k)$. The optimization objective of FedORGP is defined as:

$$\min \sum_{k=1}^{M} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \mathcal{L}_k \left(\mathcal{D}_k, \omega_k, \mathcal{P} \right), \tag{1}$$

where $|\mathcal{D}|$ denotes the total size of datasets across all clients, $|\mathcal{D}_k|$ indicates the size of the dataset of client k, $\mathcal{P}$ represents the global prototype.

3.2. Orthogonality Regularization on Prototypes

Following FedProto [32], the prototype of class c, $c \in \{1, \dots, C\}$, for client k is defined as follows:

$$P_k^c = \mathbb{E}_{(x,c) \sim \mathcal{D}_{k,c}} f_k(x; \phi_k), \tag{2}$$

where $D_{k,c}$ is the subset of D_k consisting of samples of class c. We initialize a random embedding $\tilde{P}^c$ for each class c. We define the trainable prototypes module F which comprises two Fully-Connected layers with a ReLU activation function in between, parameterized by ω_s [44]. We input the initial prototype vector $\tilde{P}^c \in \mathbb{R}^K$ into F, producing the final global prototype $\hat{P}^c = F(\tilde{P}^c; \omega_s)$. The transformation network parameters, along with the embeddings, are optimized jointly during training. This module is designed to generate adaptable prototype for each class.

To enhance the separation between global prototypes in the feature space, we adopt orthogonality regularization to train global prototypes. After client local training, the server will get the client prototypes $\mathcal{P}_k$, which consists of the prototypes of different classes of k-th client. Then, the dataset we use to train the global prototype can be expressed as $\mathcal{Q} = \bigcup_{k=1}^{M} \{P_k^c \mid c \in \mathcal{C}_k\}$, where $\mathcal{C}_k \subseteq \{0, 1, ..., C-1\}$ denotes the set of classes present on k-th client.

Our purpose is to cluster $\hat{P}^c$ and P_k^c, while discriminating between $\hat{P}^c$ and $P_k^{\bar{c}}$, where $\bar{c} \neq c$. Formally, within a

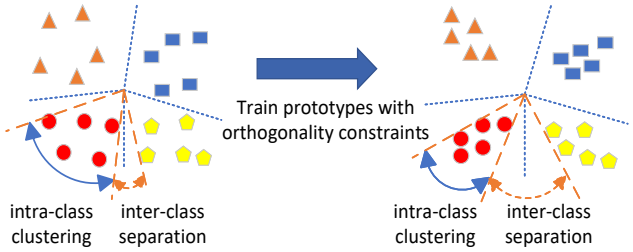


Figure 2. Orthogonality-constrained prototypes can expand inter-class margins while enhancing intra-class clustering.

mini-batch $B_p \subseteq \mathcal{Q}$, we define s as intra-prototype similarity and d as inter-prototype similarity:

$$s = \frac{1}{|B_p|} \sum_{P_k^c \in B_p} \langle P_k^c, \hat{P}^c \rangle, \tag{3}$$

$$d = \frac{1}{|B_p|} \frac{1}{|C|-1} \sum_{P_k^c \in B_p} \sum_{\substack{\bar{c} \in C \\ c \neq \bar{c}}} \left| \langle P_k^c, \hat{P}^{\bar{c}} \rangle \right|, \tag{4}$$

where $k \in \{1, \ldots, m\}$, m is the number of clients participating in the training, $|\cdot|$ is the absolute value operator. Note that the cosine similarity operator $\langle \cdot, \cdot \rangle$ on two vectors involves normalization of features (projection to a unit hypersphere) and is calculated as $\langle v_i, v_j \rangle = \frac{v_i \cdot v_j}{\|v_i\|_2 \cdot \|v_j\|_2}$, where $|| \cdot ||_2$ refers to the l_2 norm operator.

Then, we define a unified loss function $\mathcal{L}_{\text{OR}}$ that simultaneously ensures intra-class clustering and inter-class orthogonality within a mini-batch as follows:

$$\mathcal{L}_{\text{OR}} = \lambda_s * (1 - s) + \gamma * d, \tag{5}$$

where λ_s and γ are hyperparameters to balance the contributions of each term to the overall loss. Specifically, λ_s is employed to regulate the influence of the intra-class loss, while γ controls the weight of the inter-class loss. This configuration allows for a flexible adjustment of the intra-class compactness and inter-class separation.

The training objective is to minimize the loss function $\mathcal{L}_{\text{OR}}$ defined in Eq. 5. Notice that $(1 - s) > 0$ inherently holds (as $s \in (-1, 1)$, making $(1 - s) \in (0, 2)$), and d is already an absolute value with $d \in (0, 1)$. Therefore, we should maximize s toward 1 while minimizing d toward 0. As depicted in Fig.2, when minimizing this overall loss, the first term $\lambda_s * (1 - s)$ promotes clustering of prototypes within the same class, while the second term $\gamma * d$ enforces orthogonality among prototypes of different classes. The loss can be implemented efficiently in a vectorized manner on the mini-batch level, avoiding any loops.

To maintain directional independence, the prototypes of different classes are constrained to be orthogonal during the optimization process. This structure not only improve intra-class compactness but also explicitly enlarges inter-prototype separation, while preserving the semantic integrity of each class.

3.3. The Suitability of Orthogonality for CE Loss

The multi-class Cross-Entropy loss function is a common loss function used in classification problems, for each sample (x, y), the loss function can be defined as follows:

$$\begin{aligned} \mathcal{L}_{CE}(\hat{y}, y) &= -\sum_{c=1}^{C} \mathbb{I}(c = y) \log(p_c) \\ &= -\log\left(\frac{\exp(z_y)}{\sum_{c'=1}^{C} \exp(z_{c'})}\right) \\ &= -\log\left(\frac{\exp(r^y \cdot w^y)}{\sum_{c'=1}^{C} \exp(r^y \cdot w^{c'})}\right) \end{aligned} \tag{6}$$

where $\hat{y}$ is the predicted label and y is the truth label. $\mathbb{I}(\cdot)$ is an indicator function. p_c represents the predicted probability that the model predicts the sample to belong to class c. z_y is the logits of the sample. We define the classifier $W = [w^1, ..., w^C]$, where $w^y \in \mathbb{R}^K$ is the learning projection vector of label y. The CE loss can then be defined in terms of discrepancy between the predicted $\hat{y}$ and ground-truth label y, by projecting the features r^y onto the weight matrix W. When a client trains the model with the global prototypes, it ensures that r^y remains aligned with the prototype $\hat{P}^y$. During updates with SGD, the CE loss further reinforces the alignment between weight vector w^y and its corresponding feature representation r^y through a dot product operation, given as:

$$r^y \cdot w^y = \|r^y\| \cdot \|w^y\| \cdot \cos(\theta_{r^y, w^y}), \tag{7}$$

where θ_{r^y, w^y} represents the angle between r^y and w^y. Thus, when the angle between the feature representation r^y and the weight vector w^y approaches zero, their inner product is maximized, indicating directional alignment. Meanwhile, since the feature representation r^y is explicitly aligned with the global prototype $\hat{P}^y$ during training, the three components (r^y, w^y, and $\hat{P}^y$) continuously align with each other in the feature space. However, the Cross-Entropy (CE) loss inherently lacks an explicit mechanism to enforce angular separation between different classes, leading to potential overlap in feature representations and consequently suboptimal discriminative performance. To overcome this limitation, we propose orthogonality regularization (OR) to explicitly enlarge angular separation among global prototypes. By enforcing orthogonality among global prototypes, OR significantly reduces inter-class overlap, thereby ensuring robust directional consistency among r^y, w^y, and $\hat{P}^y$, and ultimately improving the discriminative capability of the classifier in heterogeneous federated environments.

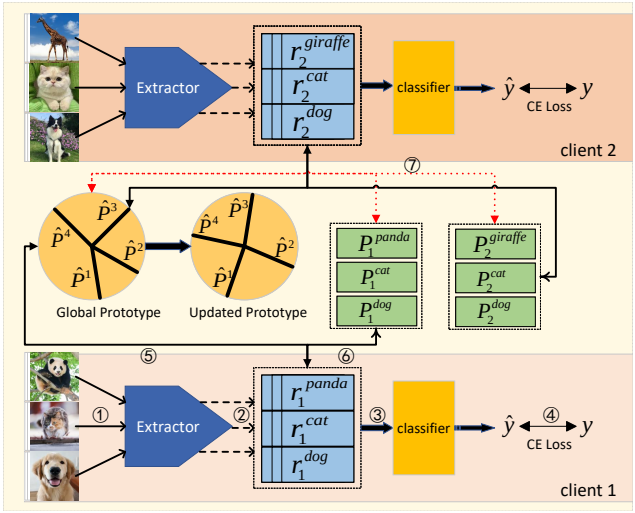


Figure 3. This image illustrates the FedORGP framework involving two clients. From ① to ④: We input the samples, get the representations, then input them into the classifier for prediction, and finally calculate the CE loss. ⑤: The dissimilarity between the feature representation and its corresponding prototype is calculated as a regularization term. ⑥: After the local model is updated, we collect the client prototypes. ⑦: Orthogonality regularization are applied to the global prototypes on the server to enhance angular separation.

3.4. Local Model Update

In FedORGP, the client updates its local model to generate embeddings that are consistent with the global prototypes shared across clients. To maintain directional alignment with the global prototype, a regularization term is added to the local loss function, which encourages feature representations from k-th client r_k^c to align with their respective global prototypes $\hat{P}^c$ while minimizing classification error. Then, the client prototype P_k^c has a more separate margin. Specifically, the loss function is formulated as follows:

$$\begin{aligned}\mathcal{L}_k :=& \mathbb{E}_{(x,y)\sim\mathcal{D}_k}\mathcal{L}_{CE}(h_k(f_k(x;\phi_k);\theta_k),y)+ \\ & \lambda_c\cdot\mathbb{E}_{(x,y)\sim\mathcal{D}_k}\mathcal{L}_R(f_k(x;\phi_k),\hat{P}^y),\end{aligned} \tag{8}$$

where λ_c is a hyperparameter. This regularization $\mathcal{L}_R$ facilitates the alignment between feature representations and global prototypes across clients, enhancing the global consistency and robustness of the federated model in heterogeneous environments.

3.5. FedORGP Framework

We present the complete workflow of FedORGP in Alg.1 and give an illustration in Fig.3. The well-trained, class-separable global prototypes are then distributed to clients in the subsequent round to guide client training for improving separability among feature representations.

Algorithm 1 The learning process of FedORGP

Input: M clients with heterogeneous models and data, trainable global prototypes $\hat{\mathcal{P}}$ on the server, local epochs E, and total communication rounds T
Output: Well-trained client models.

Server:
for round $t = 1, \ldots, T$ **do**
  Randomly sample a client subset $\mathcal{I}^t$
  Send $\hat{\mathcal{P}}$ to clients in $\mathcal{I}^t$
  for each Client $k \in \mathcal{I}^t$ **in parallel do**
    Client k performs local training
  end for
  Train the global prototype on the server by Eq. 5
end for
return Well-trained client models.
Client Local Training:
  for iteration $e = 1, \ldots, E$ **do**
    Local training with the global prototype by Eq. 8
  end for
  Collect local prototypes $\{P_k^c | c \in \mathcal{C}_k\}$ by Eq. 2
  Send local prototypes $\mathcal{P}_k$ to the server

4. Convergence Analysis

To analyse the convergence of FedORGP, we first introduce some additional notes as in the existing framework. Unless otherwise stated, we always write the client-side loss function $\mathcal{L}_k(\mathcal{D}_k, \omega_k, \mathcal{P})$ as $\mathcal{L}_k$. In order to analyse the convergence of FedORGP, we first introduce some additional assumptions [4, 32, 42].

Assumption 1. (Lipschitz Smooth). The k-th client's gradient of local loss function is L_1-Lipschitz continuous:

$$\|\nabla\mathcal{L}_k^{t_1} - \nabla\mathcal{L}_k^{t_2}\|_2 \leqslant L_1\|\omega_k^{t_1} - \omega_k^{t_2}\|_2, \tag{9}$$

where $\forall t_1, t_2 > 0, k \in \{1, \ldots, m\}$. That also means that local objective function is L_1-Lipschitz smooth:

$$\mathcal{L}_k^{t_1} - \mathcal{L}_k^{t_2} \leqslant \langle\nabla\mathcal{L}_k^{t_2}, (\omega_k^{t_1} - \omega_k^{t_2})\rangle + \frac{L_1}{2}\|\omega_k^{t_1} - \omega_k^{t_2}\|_2^2, \tag{10}$$

Assumption 2. (Unbiased Gradient and Bounded Variance). The stochastic gradient $g_k^t = \nabla\mathcal{L}(\xi_i, \omega^t)$ is an unbiased estimator of the local gradient for k-th client, where ξ_i is a random variable representing the randomness in the data (e.g., mini-batch). Suppose its expectation is

$$\mathbb{E}_{\xi_i\sim D_i}[g_k^t] = \nabla\mathcal{L}_k(\omega_k^t), \forall k \in \{1, 2, \ldots, m\}, \tag{11}$$

and there exists $\sigma^2 \geq 0$, then the variance of random gradient g_k^t is bounded by:

$$\mathbb{E}[\left\|g_k^t - \nabla\mathcal{L}_k(\omega_k^t)\right\|_2^2] \leqslant \sigma^2, \forall k \in \{1, 2, \ldots, m\}. \tag{12}$$

Assumption 3. (Bounded Gradient). The stochastic gradient of the global prototype is bounded by G:

$$\mathbb{E}[\|\nabla \mathcal{L}_{server}\|_2] \leqslant G \tag{13}$$

We denote $e \in \{0, 1, 2, \ldots, E\}$ as the local iteration, and t as the global round. Here, tE indicates the time step before orthogonality regularization at t round, and $tE + 0$ represents the interval between global prototype training and the first iteration of the $(t + 1)$ round. Under above assumptions, we present the theoretical results at the non-convex condition. Since all client-side loss function share the same property, we omit the subscript k and denote the loss function as $\mathcal{L}$. The detailed proof process is in the Appendix.

Lemma 1. Based on Assumption 1 and 2, after clients training one round, the local client model loss satisfies [32]:

$$\mathbb{E}\left[\mathcal{L}^{(t+1)E}\right] \leqslant \mathcal{L}^{tE+0} - \left(\eta - \frac{L_1\eta^2}{2}\right)\sum_{e=0}^{E-1}\|\mathcal{L}^{tE+e}\|_2^2 + \frac{L_1 E\eta^2}{2}\sigma^2 \tag{14}$$

Lemma 2. Based on Assumption 3, we further analyze how the introduction of global prototype orthogonality regularization influences the client-side loss:

$$\mathbb{E}\left[\mathcal{L}^{(t+1)E+0}\right] \leqslant \mathbb{E}\left[\mathcal{L}^{(t+1)E}\right] + \lambda_c\eta G. \tag{15}$$

Theorem 1. Leveraging Lemma 1 and Lemma 2, we establish our main theoretical convergence results of the loss reduction over multiple steps in local training:

$$\mathbb{E}\left[\mathcal{L}^{(t+1)E+0}\right] \leqslant \mathcal{L}^{tE+0} - \left(\eta - \frac{L_1\eta^2}{2}\right)\sum_{e=0}^{E-1}\|\mathcal{L}^{tE+e}\|_2^2 + \frac{\eta^2 L_1 E\sigma^2}{2} + \lambda_c\eta G. \tag{16}$$

Theorem 2. We extend the analysis to multiple rounds, providing a convergence rate for FedORGP in a non-convex setting. For an arbitrary client and any $\epsilon > 0$, the following inequality holds:

$$\frac{1}{T}\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}\mathbb{E}\left[\|\mathcal{L}^{tE+e}\|_2^2\right] \leqslant \frac{2\left(\mathcal{L}^{t=0} - \mathcal{L}^*\right)}{T\eta\left(2 - L_1\eta\right)} + \frac{L_1 E\eta\sigma^2}{2 - L_1\eta} + \frac{2\lambda_c G}{2 - L_1\eta} \leqslant \epsilon \tag{17}$$

when $\eta < \frac{2(\epsilon - \lambda_c G)}{L_1(\epsilon + E\sigma^2)}$ and $\lambda_c < \frac{\epsilon}{G}$. It shows that the model can converge to a solution with a rate that is proportional to $O(1/T)$, where T is the number of communication rounds.

5. Experiments

5.1. Setup

Datasets. To evaluate our model, we conduct experiments on four image classification datasets: CIFAR-10, CIFAR-100 [14], Flowers102 [26] and TinyImagenet [3].

Baselines. To evaluate our proposed FedORGP, we compare it with seven popular methods that are applicable in HtFL, including LG-FedAvg [20], FML [29], FedGen [45], FedKD [39], FedProto [32], FedGH [42] and FedTGP [44].

Model heterogeneity. To comprehensively evaluate the robustness and adaptability of our algorithm across different model architectures, we test our approach on four heterogeneous model groups (HMG): HMG_3 (4-layer CNN [25], GoogleNet [30] and MobileNet_v2 [28]), HMG_5 (ResNet18/34/50/101/152 [8]), HMG_8 (Combines HMG_3 and HMG_5), and HMG_{10} (Extends HMG_8 with DenseNet121 [11] and EfficientNet-B0 [31]) [44]. These HMGs span lightweight and complex convolutional networks, enabling evaluation of the proposed method across a broad range of model complexities.

Statistical heterogeneity. We conduct extensive experiments with two widely used statistically heterogeneous settings, the pathological setting and the practical setting [21, 25, 45]. For the pathological setting, we assign a fixed number of classes to the client. For the practical setting, we leverage Dirichlet distribution to simulate more realistic class imbalance across clients. The hyperparameter α controls the strength of heterogeneity. Notably, a smaller α implies a higher non-IID data distribution among clients.

Training configuration. Our experiments are conducted on an x86_64 architecture with Ubuntu as the operating system. The platform is equipped with 8 NVIDIA V100 GPUs, each with 32 GB of memory, and CUDA version 12.2. Unless explicitly specified, we use the following settings. The federated learning setup includes a default configuration of 20 clients and the client participation ratio ρ = 1. Both the client and server employ the SGD optimizer with a learning rate of 0.01, and both undergo training for a single epoch. The batch sizes for client B and server B_p are both set to 32. For model heterogeneity, we use the HMG_8 by default. For statistical heterogeneity testing, under the pathological setting, we distribute unbalanced data of 2/10/10/20 classes to each client from a total of 10/100/102/200 classes on Cifar-10/Cifar-100/Flowers102/TinyImagenet datasets. In the practical setting, a Dirichlet distribution with α = 0.05 is used to simulate real life scenario. Each client's private data is divided into a training set (75%) and a test set (25%). Each algorithm is trained three times, with each training run consisting of 100 epochs. For the evaluation, we count the best accuracy achieved in each training run, and the final result is calculated as the mean and variance of the accuracy of the three runs.

Table 1. The test accuracy (%) on four datasets in the pathological and practical settings using the HMG_8 model group.

Settings	Pathological Setting				Practical Setting			
Datasets	Cifar-10	Cifar-100	Flowers102	TinyImagenet	Cifar-10	Cifar-100	Flowers102	TinyImagenet
FML	82.56 ± 0.14	50.27 ± 0.09	52.17 ± 0.98	33.37 ± 0.34	91.85 ± 0.24	46.45 ± 0.17	48.51 ± 0.57	37.67 ± 0.19
LG-FedAvg	83.63 ± 0.09	55.51 ± 0.10	58.90 ± 0.22	34.99 ± 0.29	92.34 ± 0.06	49.82 ± 0.16	53.65 ± 0.46	38.35 ± 0.27
FedGen	83.63 ± 0.24	55.37 ± 0.35	59.39 ± 0.19	35.18 ± 0.12	92.43 ± 0.11	49.90 ± 0.21	54.59 ± 0.34	38.44 ± 0.14
FedProto	80.63 ± 0.05	51.89 ± 0.31	54.48 ± 0.20	33.78 ± 0.20	79.32 ± 0.22	42.77 ± 0.13	26.66 ± 0.57	24.57 ± 0.08
FedKD	83.41 ± 0.66	53.01 ± 1.48	52.61 ± 0.74	34.70 ± 0.91	91.85 ± 0.31	48.95 ± 1.24	51.44 ± 0.75	39.13 ± 0.36
FedGH	83.49 ± 0.29	55.25 ± 0.12	60.15 ± 1.08	35.04 ± 0.16	92.66 ± 0.10	49.52 ± 0.05	54.24 ± 0.50	38.67 ± 0.11
FedTGP	84.75 ± 0.06	53.12 ± 0.24	58.60 ± 0.14	32.41 ± 0.09	92.26 ± 0.68	49.09 ± 0.16	56.62 ± 0.40	37.14 ± 0.16
FedORGP	**85.71 ± 0.10**	**61.12 ± 0.15**	**64.15 ± 1.66**	**37.04 ± 0.56**	**94.01 ± 0.03**	**55.33 ± 0.23**	**60.34 ± 0.48**	**40.15 ± 0.40**

Evaluation Metrics. We measure the average test accuracy (%) across all client models. We further evaluate the robustness of our algorithm under varying client participation rates, degrees of non-IID data, and feature dimension K.

5.2. Performance Comparison

The test accuracy of all methods across four datasets is presented in Tab. 1. FedORGP consistently outperforms all baselines on these datasets, achieving up to a 8% gain over FedTGP in pathological settings and 6.24% in practical settings on Cifar-100. This improvement is due to the fact that FedORGP can separate features in terms of angles, which can be well integrated with the CE loss, thereby improving classification results more effectively than methods which only increase the Euclidean distance to perform prototype separation in high-dimensional space. In addition, FedORGP is also more efficient. While FedTGP requires 100 epochs of training on the server, FedORGP requires only 1 epoch of training with orthogonal regularisation.

Table 2. Test on Cifar-100 (10 classes per client) with three different HMGs to test robustness to heterogeneous models.

	Heterogenous Model Groups		
Settings	HMG_3	HMG_5	HMG_{10}
FML	58.72 ± 0.15	42.84 ± 0.41	45.50 ± 0.50
LG-FedAvg	59.62 ± 0.24	51.46 ± 0.21	50.19 ± 0.19
FedGen	59.82 ± 0.38	50.97 ± 0.26	50.07 ± 0.13
FedProto	58.83 ± 0.52	49.08 ± 0.00	48.04 ± 0.02
FedKD	**61.05 ± 0.04**	46.24 ± 2.55	47.57 ± 2.00
FedGH	59.21 ± 0.15	50.76 ± 0.30	50.41 ± 0.19
FedTGP	60.41 ± 0.56	48.24 ± 0.47	48.11 ± 0.07
FedORGP	60.64 ± 0.50	**61.58 ± 0.42**	**58.54 ± 0.65**

5.3. Impact of Model Heterogeneity

To examine the impact of model heterogeneity in HtFL, we assess the performance of FedORGP on three additional HMG settings. We show results in Tab. 2, our findings indicate that all baselines perform well under the HMG_3 condition, as the reduced model heterogeneity significantly lessened its adverse impact on performance, while all methods perform worse with larger model heterogeneity. As the degree of model heterogeneity rises, the advantages of FedORGP becomes more pronounced, up to 10.12% at HMG_5 and 8.13% at HMG_{10}. In addition, FedORGP also has the smallest performance reduction when model heterogeneity increases, which shows that FedORGP is robust to model heterogeneity.

5.4. Robustness to Participation Rate

To compare FedORGP against baselines under varying total client numbers M and client participation rates ρ, we design three distinct settings. The results, shown in Tab. 3, ensure a consistent number of participating clients per round. FedORGP aims to promote directional independence, which is seamlessly integrated with CE loss. This separation minimizes inter-class overlap, thereby enabling the model to achieve robust generalization across clients with varying participation rates.

Table 3. Test on Cifar-100 (10 classes per client) under three participation rate settings with HMG_8. "-" means the baseline can't converge.

	Different M Clients and Join Ratio		
Settings	M=40, ρ=50%	M=80, ρ=25%	M=100, ρ=20%
FML	33.96 ± 0.25	19.72 ± 0.11	30.65 ± 0.25
LG-FedAvg	47.05 ± 0.09	39.59 ± 0.23	40.98 ± 0.51
FedGen	47.05 ± 0.40	39.76 ± 0.35	41.10 ± 0.53
FedProto	38.30 ± 0.53	13.49 ± 0.84	16.16 ± 0.55
FedKD	38.12 ± 3.36	21.76 ± 1.60	34.59 ± 2.73
FedGH	46.82 ± 0.24	40.08 ± 0.35	-
FedTGP	44.87 ± 0.58	37.51 ± 0.36	36.75 ± 0.42
FedORGP	**54.88 ± 0.45**	**44.86 ± 0.11**	**43.22 ± 0.45**

5.5. Robustness to Statistical Heterogeneity

To evaluate the robustness of all baseline methods under different non-IID data distributions, we conducted experiments on two dataset configurations. As demonstrated in Tab. 4, FedORGP ensures optimal inter-class separation by leveraging orthogonality regularization, reducing embeddings overlap even in highly skewed data distributions. This

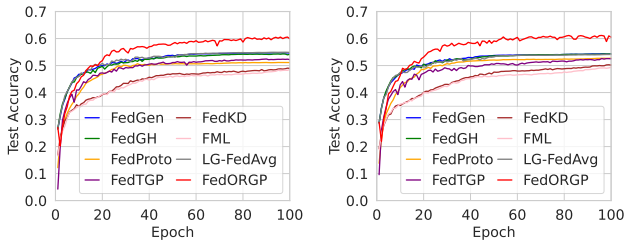
Table 4. Test on two data distributions under various degrees of non-IID on the Cifar-100 dataset with HMG_8.

Settings	Pathological Setting		Practical Setting	
	5 classes/client	20 classes/client	α=0.1	α=0.01
FML	66.40 ± 0.33	33.13 ± 0.24	37.30 ± 0.09	61.17 ± 0.24
LG-FedAvg	71.84 ± 0.17	37.22 ± 0.36	39.97 ± 0.11	66.06 ± 0.14
FedGen	71.68 ± 0.09	37.47 ± 0.09	40.33 ± 0.37	66.06 ± 0.08
FedProto	70.27 ± 0.07	35.15 ± 0.18	34.30 ± 0.16	56.97 ± 0.36
FedKD	69.75 ± 1.35	34.24 ± 0.79	39.81 ± 1.12	63.86 ± 1.46
FedGH	71.42 ± 0.29	36.99 ± 0.23	39.90 ± 0.21	65.90 ± 0.27
FedTGP	69.86 ± 0.33	35.98 ± 0.30	38.60 ± 0.10	67.12 ± 0.11
FedORGP	**75.95 ± 0.24**	**45.45 ± 0.34**	**44.70 ± 0.29**	**70.32 ± 0.36**

demonstrates its effectiveness in promoting generalization in statistical heterogeneity environments.

5.6. Impact of Feature Dimension

To assess the robustness of all baselines to the K, we performe experiments with three feature dimensions: 128, 256, and 1024, as indicated in Fig. 4. Results indicate that all baselines maintain a high degree of robustness, showing minimal variation in performance.



(a) K=128 (b) K=256

(c) K=512 (d) K=1024

Figure 4. Test on Cifar-100 (10 classes per client) under different feature dimensions with HMG_8.

5.7. Ablation Study

This section analyzes the ablation study in Tab. 5. FedORGP without orthogonality regularization (FedORGP w/o OC) aligns the embedding layer with the global prototype derived from weighted averaging. While the performance on CIFAR-10 is passable, FedORGP w/o OC exhibits a sharp performance decline on CIFAR-100 and Flowers102 datasets, likely due to substantial global prototype overlap. Introducing the orthogonality regularization in FedORGP markedly enhances prototype separation, improving performance by explicitly increasing angular separation across classes. FedORGP outperforms FedTGP, as reducing prototype similarity integrates more effectively with cross-entropy (CE) loss than merely increasing prototype distances.

Table 5. Test on three datasets under the practical setting using the HMG_8 for ablation study.

	FedTGP	FedORGP w/o OR	FedORGP
Cifar-10	84.75 ± 0.06	78.05 ± 0.52	**85.71 ± 0.10**
Cifar-100	53.12 ± 0.24	35.11 ± 0.63	**61.12 ± 0.15**
Flowers102	58.60 ± 0.14	42.43 ± 0.57	**64.15 ± 1.66**

5.8. Hyperparameter Combination

We conducted a grid search across $(\lambda_c, \lambda_s, \gamma)$ in Tab. 6 to determine the optimal combination of hyperparameters. The selected values for λ_s and γ, specifically (1, 10), enable the orthogonality constraint to effectively enhance inter-class separation without sacrificing the integrity of class representations. This configuration fosters a more robust global prototype compared to the (10, 100) pairing at the same ratio. Additionally, a higher λ_c imposes a stricter alignment constraint, thereby improving feature representation consistency across clients, which is crucial for maintaining a unified representation space against statistical heterogeneity. The highest accuracy 61.15% is achieved with the (100, 1, 10) setting.

Table 6. Hyperparameter search on Cifar-100 with HMG_8.

λ_c	λ_s	γ	Acc	λ_c	λ_s	γ	Acc	λ_c	λ_s	γ	Acc
100.0	100.0	100.0	40.34	10.0	100.0	100.0	50.67	1.0	100.0	100.0	53.94
100.0	100.0	10.0	40.32	10.0	100.0	10.0	48.12	1.0	100.0	10.0	54.43
100.0	100.0	1.0	48.32	10.0	100.0	1.0	51.52	1.0	100.0	1.0	54.82
100.0	10.0	100.0	59.17	10.0	10.0	100.0	56.60	1.0	10.0	100.0	53.94
100.0	10.0	10.0	55.37	10.0	10.0	10.0	54.76	1.0	10.0	10.0	54.34
100.0	10.0	1.0	51.96	10.0	10.0	1.0	53.47	1.0	10.0	1.0	54.22
100.0	1.0	100.0	58.97	10.0	1.0	100.0	55.14	1.0	1.0	100.0	50.64
100.0	**1.0**	**10.0**	**61.15**	10.0	1.0	10.0	57.35	1.0	1.0	10.0	54.94
100.0	1.0	1.0	59.51	10.0	1.0	1.0	57.11	1.0	1.0	1.0	54.20

6. Conclusion

In this work, we propose a novel HtFL method, FedORGP, which employs orthogonality regularization on global prototypes at the server side to improve inter-prototype separation and maximize directional independence while preserving semantic integrity. On the client side, FedORGP aligns the embeddings with prototypes, enhancing its compatibility with CE loss. This approach overcomes the limitations of FedTGP by enhancing prototype separation through angular separation rather than merely increasing Euclidean distances between prototypes. Extensive experiments demonstrate that FedORGP consistently outperforms all baseline methods, and exhibits superior performance in scenarios where model and statistical heterogeneity coexist.

References

[1] Ons Aouedi, Alessio Sacco, Kandaraj Piamrat, and Guido Marchetto. Handling privacy-sensitive medical data with federated learning: challenges and future directions. *IEEE journal of biomedical and health informatics*, 27(2):790–803, 2022. 1

[2] Yavuz Faruk Bakman, Duygu Nur Yaldiz, Yahya H Ezzeldin, and Salman Avestimehr. Federated orthogonal training: Mitigating global catastrophic forgetting in continual federated learning. *arXiv preprint arXiv:2309.01289*, 2023. 3

[3] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017. 6

[4] Yutong Dai, Zeyuan Chen, Junnan Li, Shelby Heinecke, Lichao Sun, and Ran Xu. Tackling data heterogeneity in federated learning with class prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7314–7322, 2023. 3, 5

[5] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*, 2020. 2

[6] Lingzhi Gao, Zexi Li, Yang Lu, and Chao Wu. Fedios: Decoupling orthogonal subspaces for personalization in feature-skew federated learning. *arXiv preprint arXiv:2311.18559*, 2023. 3

[7] Ruchi Gupta and Tanweer Alam. Survey on federated-learning approaches in distributed environment. *Wireless personal communications*, 125(2):1631–1652, 2022. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[9] Michael Hersche, Geethan Karunaratne, Giovanni Cherubini, Luca Benini, Abu Sebastian, and Abbas Rahimi. Constrained few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9057–9067, 2022. 3

[10] 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. 2

[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 6

[12] Wenke Huang, Mang Ye, Zekun Shi, He Li, and Bo Du. Rethinking federated learning with domain shift: A prototype view. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16312–16322. IEEE, 2023. 3

[13] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2): 1–210, 2021. 1

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009. 6

[15] Gihun Lee, Minchan Jeong, Sangmook Kim, Jaehoon Oh, and Se-Young Yun. Fedsol: Stabilized orthogonal learning with proximal restrictions in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12512–12522, 2024. 3

[16] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019. 3

[17] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020. 1

[18] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, pages 429–450, 2020. 1

[19] Zexi Li, Xinyi Shang, Rui He, Tao Lin, and Chao Wu. No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5319–5329, 2023. 2

[20] 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. 2, 6

[21] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems*, pages 2351–2363. Curran Associates, Inc., 2020. 3, 6

[22] Sun-Ao Liu, Yiheng Zhang, Zhaofan Qiu, Hongtao Xie, Yongdong Zhang, and Ting Yao. Learning orthogonal prototypes for generalized few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11319–11328, 2023. 3

[23] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984, 2021. 2

[24] Xiaodong Ma, Jia Zhu, Zhihao Lin, Shanxuan Chen, and Yangjie Qin. A state-of-the-art survey on solving non-iid data in federated learning. *Future Generation Computer Systems*, 135:244–258, 2022. 1

[25] 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. 1, 6

[26] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 6

[27] Kanchana Ranasinghe, Muzammal Naseer, Munawar Hayat, Salman Khan, and Fahad Shahbaz Khan. Orthogonal projection loss. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12333–12343, 2021. 1, 3

[28] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 6

[29] 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. 3, 6

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 6

[31] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 6

[32] 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*, pages 8432–8440, 2022. 1, 2, 3, 5, 6

[33] Yue Tan, Guodong Long, Jie Ma, Lu Liu, Tianyi Zhou, and Jing Jiang. Federated learning from pre-trained models: A contrastive learning approach. *Advances in neural information processing systems*, 35:19332–19344, 2022. 3

[34] Srikanth Thudumu, Philip Branch, Jiong Jin, and Jugdutt Singh. A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data*, 7:1–30, 2020. 1, 3

[35] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 2

[36] Lianyu Wang, Meng Wang, Daoqiang Zhang, and Huazhu Fu. Model barrier: A compact un-transferable isolation domain for model intellectual property protection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20475–20484, 2023. 1

[37] Lei Wang, Jieming Bian, Letian Zhang, Chen Chen, and Jie Xu. Taming cross-domain representation variance in federated prototype learning with heterogeneous data domains. *arXiv preprint arXiv:2403.09048*, 2024. 3

[38] Dingzhu Wen, Ki-Jun Jeon, and Kaibin Huang. Federated dropout—a simple approach for enabling federated learning on resource constrained devices. *IEEE wireless communications letters*, 11(5):923–927, 2022. 2

[39] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature communications*, 13(1): 2032, 2022. 3, 6

[40] Jian Xu, Xinyi Tong, and Shao-Lun Huang. Personalized federated learning with feature alignment and classifier collaboration. *arXiv preprint arXiv:2306.11867*, 2023. 3

[41] 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. 1

[42] Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. Fedgh: Heterogeneous federated learning with generalized global header. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8686–8696, 2023. 1, 2, 5, 6

[43] Jie Zhang, Song Guo, Jingcai Guo, Deze Zeng, Jingren Zhou, and Albert Y Zomaya. Towards data-independent knowledge transfer in model-heterogeneous federated learning. *IEEE Transactions on Computers*, 72(10):2888–2901, 2023. 3

[44] 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*, pages 16768–16776, 2024. 1, 2, 3, 6

[45] 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. 2, 6