

# FedOC: Optimizing Global Prototypes with Orthogonality Constraints for Enhancing Embeddings Separation in Heterogeneous Federated Learning

Fucheng Guo<sup>\*</sup>, Zeyu Luan<sup>†</sup>, Qing Li<sup>‡</sup>, Dan Zhao<sup>§</sup>, Yong Jiang<sup>¶</sup>

## 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 FedOC, a novel HtFL algorithm designed to improve global prototype separation through orthogonality constraints, which not only increase intra-class prototype similarity but also significantly expand 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 FedOC's convergence under non-convex conditions. Extensive experiments demonstrate that FedOC outperforms seven state-of-the-art baselines, achieving up to a 10.12% accuracy improvement in both statistical and model heterogeneity settings.*

## 1. Introduction

In recent years, Federated Learning (FL) has emerged as a promising distributed machine learning paradigm [5]. 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].

Previous work mainly focuses on homogeneous FL, which assumes that all client models are identical. However, in large-scale real-world scenarios, considerable dis-

parities exist in both client data distribution, often termed statistical heterogeneity, and hardware resources, known as system heterogeneity [16, 40]. Data heterogeneity severely impacts the convergence and performance of global model [23]. 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 [40]. In addition, homogeneous FL trains a shared global model by exchanging gradients, which further imposes significant communication costs as well as privacy exposure risks [37, 43].

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 [33, 41, 43]. 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 [33] faces several limitations. First, aggregating global prototypes on the server side via weighted averaging requires clients to upload sample size, which may lead to leakage of data distribution information [41]. Second, due to statistical heterogeneity, the intra-class distribution discrepancy exists for the same class across clients. This disparity leads to inconsistent decision boundaries for the same class, which hinders the representation of a unified class distribution in the global feature space [43]. Furthermore, as presented in Fig. 1a, the aggregated global prototypes suffer from limited separation margins, thereby diminishing their discriminative ability across classes.

The recent HtFL solution FedTGP [43] has successfully improved separation by increases the Euclidean distance between prototypes. However, 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 Figure 1b, in high-dimensional spaces, the Euclidean distance makes it difficult to effectively distinguish different samples as distance variations between samples diminish [35]. Third, augmenting Euclidean distances fails to effectively

<sup>\*</sup>Shenzhen International Graduate School, Tsinghua University, gfc23@mails.tsinghua.edu.cn

<sup>†</sup>Peng Cheng Laboratory, China, luanz@pcl.ac.cn

<sup>‡</sup>Peng Cheng Laboratory, China, liq@pcl.ac.cn

<sup>§</sup>Peng Cheng Laboratory, China, zhaod01@pcl.ac.cn

<sup>¶</sup>Shenzhen International Graduate School, Tsinghua University, liq@pcl.ac.cn

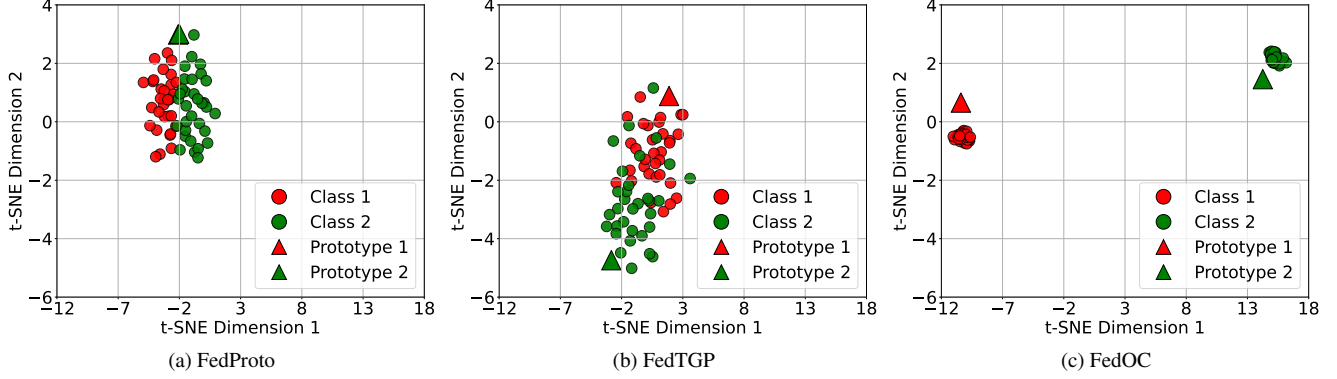


Figure 1. We train models on the CIFAR-10 dataset under a pathological partitioning scheme (each client is assigned two classes). We use t-SNE [36] to visualize their performance on previously unseen test data in the feature space. The results indicate that FedProto [33] exhibits weak feature separation, with the two prototypes nearly overlapping. FedTGP [43] increases prototype margin but still lacks sufficient distinction in the feature space. In contrast, our FedOC not only reduces inter-prototype similarity but also achieves high intra-class compactness, ensuring that feature representations of samples from the same class are tightly clustered. This outcome implies that FedOC enhances both classification and generalization abilities in settings with statistical and model heterogeneity.

leverage the angular separation characteristics inherent in the CE loss [26].

To address these limitations, we propose a novel HtFL algorithm called FedOC, which optimizes global prototypes through orthogonality constraints (OC). FedOC 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 FedOC against seven HtFL methods across three datasets with two statistical heterogeneity settings and multiple model heterogeneity scenarios. 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 follow:

- We present a training framework FedOC for HtFL with orthogonality constraints. To the best of my knowledge, this is the first work to introduce orthogonality constraints into heterogeneous federated learning.
- We observed that increasing the Euclidean distance between prototypes is incapable of effectively integrating with the angular characteristics inherent to CE loss. Our proposed FedOC algorithm enhances intra-class clustering within the feature space while explicitly promoting angular separation between classes, which aligns with the CE loss.
- We offer a theoretical convergence guarantee for FedOC

and rigorously establish the convergence rate under non-convex conditions.

## 2. Related Work

### 2.1. Heterogenous Federated Learning

Federated Learning (FL) has gained significant attention as a privacy-preserving distributed machine learning approach, allowing collaborative model training without sharing private data [24]. The traditional FL methods, such as FedAvg [24], focus on training a global model across multiple clients by aggregating gradients [10]. However, these methods face major challenges in the presence of statistical and model heterogeneity, often resulting in suboptimal model performance due to non-IID data and diverse client hardware capabilities [15]. Although personalized FL methods [3, 17, 31] are introduced to address the statistical heterogeneity in FL, they remain unsuitable for scenarios where clients deploy heterogeneous models tailored to their specific tasks.

Heterogeneous Federated Learning (HtFL) has emerged as a solution that supports both model and statistical heterogeneity simultaneously, while ensuring user privacy. Early HtFL solutions [4, 8, 38] let clients sample different sub-models 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 concerns regarding the privacy of 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 or a global generator, as a way to transfer global knowledge. LG-FedAvg [19] and FedGH

[41] 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 [18, 22]. FedGen [44] 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 [43]. 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 knowledge or prototypes. FedMD [14] and FedDF [20] facilitate knowledge transfer between participants through distillation over public datasets. However, ideal public datasets are often difficult to access [42]. FML [29] and FedKD [39] train and share a small auxiliary model using mutual distillation, circumventing the need for a global dataset [43]. Nonetheless, the bi-directional distillation process relies on frequent communication between clients and the server, which can result in significant communication overhead.

Prototype-based FL, e.g. FedProto [33] and FedPCL [34], have demonstrated the potential in handling both model and statistical heterogeneity. These approaches not only improve model convergence in non-IID settings but also significantly reduce communication overhead by focusing on prototype updates instead of full model updates [34]. 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 [43] increases prototype separation by contrastive learning without using weighted averages, boosting the Euclidean distance between embedding layers does not integrate well with CE loss [26, 35].

In this work, we apply orthogonality constraints to explicitly enlarge the angular margins between global prototypes. Under the guidance of the global prototypes, client embeddings maintain alignment with their corresponding global prototypes while preserving directional independence, facilitating a more effective integration with the angular characteristics inherent to CE loss.

## 2.2. 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 [26]. The previous formulations [13, 28] to achieve orthogonality in the feature space generally rely on computing singular value decomposition (SVD). However, these SVD-based methods are often computationally expensive. C-FSCIL [7], OPL [26] and POP [21] enhance feature sep-

aration between classes via orthogonality constraints. 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 [12] 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  $\phi_k$ , and a classifier  $h_k$  parameterized by  $\theta_k$ . Given a sample pair  $(x, y)$ , the feature extractor  $f_k$  transforms  $x \in \mathbb{R}^D$  into a feature representation  $r \in \mathbb{R}^K$ :

$$r = f_k(x; \phi_k), \quad (1)$$

where  $D$  is the dimension of input and  $K$  is the dimension of feature representation ( $K \ll D$ ). The classifier  $h_k$  further maps this  $r$  to a probability distribution  $output \in \mathbb{R}^C$ :

$$output = h_k(r; \theta_k), \quad (2)$$

where  $C$  is the total number of classes. The parameters of client model is defined as  $\omega_k = (\phi_k, \theta_k)$ .

The optimization objective of FedOC is defined as:

$$\min \frac{1}{M} \sum_{k=1}^M \mathcal{L}_k(\mathcal{D}_k, \omega_k, \mathcal{P}), \quad (3)$$

where  $\mathcal{D}_k$  is the dataset of client  $k$ ,  $\mathcal{P}$  is the global prototype and  $\mathcal{L}_k$  is the total loss of client  $k$ .

### 3.2. Orthogonality Constraints of Global Prototype

Following FedProto [33], the prototype of class  $c$ ,  $c \in \{1, \dots, C\}$ , for client  $k$  is defined as follows:

$$P_k^c = \mathbb{E}_{(x,c) \sim \mathcal{D}_{k,c}} f_k(x; \phi_k), \quad (4)$$

where  $\mathcal{D}_{k,c}$  is the subset of  $\mathcal{D}_k$  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  $\omega_s$  [43]. 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 constraints

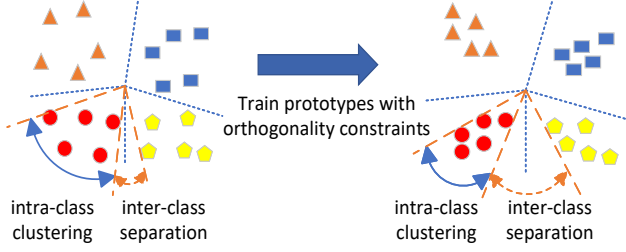


Figure 2. Orthogonality-constrained prototypes can expand inter-class margins while enhancing intra-class clustering.

to train global prototypes. After client local training, the server will get the client prototype  $P_k^c$  of class  $c$ , and prototypes of the remaining classes  $P_{\bar{c}}^{\bar{c}}$ , where  $\bar{c} \neq c$ . Our purpose is to cluster  $\hat{P}^c$  and  $P_k^c$ , while discriminating between  $\hat{P}^c$  and  $P_{\bar{c}}^{\bar{c}}$ . Formally, we define  $s$  as intra-prototype similarity and  $d$  as inter-prototype similarity:

$$s = \frac{1}{|B|} \sum_{P_k^c \in B} \langle P_k^c, \hat{P}^c \rangle, \quad (5)$$

$$d = \frac{1}{|B|} \frac{1}{|C|} \sum_{P_k^c \in B} \sum_{c \neq \bar{c}, \bar{c} \in C} |\langle P_k^c, \hat{P}^{\bar{c}} \rangle|, \quad (6)$$

where  $k \in \{1, \dots, m\}$ ,  $m$  is the number of clients participating in the training,  $|\cdot|$  is the absolute value operator,  $B$  denotes mini-batch, and  $|B|$  is the mini-batch size. 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:

$$\langle v_i, v_j \rangle = \frac{v_i \cdot v_j}{\|v_i\|_2 \cdot \|v_j\|_2}, \quad (7)$$

where  $\|\cdot\|_2$  refers to the  $l_2$  norm operator.

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

$$\mathcal{L}_{OC} = \lambda_s * (1 - s) + \gamma * d, \quad (8)$$

where  $\lambda_s$  and  $\gamma$  are hyperparameters to balance the contributions of each term to the overall loss. Specifically,  $\lambda_s$  is employed to regulate the influence of the intra-class loss, while  $\gamma$  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  $\mathcal{L}_{OC}$  defined in Eq. 8. 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 OC for CE Loss

The Cross Entropy (CE) loss function, defined as:

$$\mathcal{L}_{CE}(\hat{y}, y) = -\log \frac{\exp(r^y \cdot w^y)}{\sum_{j \neq y} \exp(r^y \cdot w^j)}, \quad (9)$$

is widely used in deep learning models for classification tasks. We define the classifier  $W = [w^1, \dots, w^C]$ , where  $w^y \in \mathbb{R}^K$  is the learning projection vector of label  $y$ . The traditional 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 feature representation  $r^y$  and its corresponding weight vector  $w^y$  through a dot product operation, given as:

$$r^y \cdot w^y = \|r^y\| \cdot \|w^y\| \cdot \cos(\theta_{r^y, w^y}), \quad (10)$$

where  $\theta_{r^y, w^y}$  represents the angle between  $r^y$  and  $w^y$ . Thus,  $w^y$  can be considered as the global prototype  $\hat{P}^y$  for class  $y$ . It can be observed that when  $\theta_{r^y, w^y}$  approaches 0, the dot product is maximized, thus aligning two vectors in feature space. This alignment imposes an angular constraint on their relationship, which is why CE loss is said to have an angular property. However, CE lacks an explicit mechanism to enforce inter-class separation. This limitation facilitates the overlap of different classes in the feature space, leading to suboptimal discrimination.

Recognizing this limitation, we propose to enhance the discriminative power of CE loss by introducing orthogonality constraints (OC). By enforcing orthogonality between class prototypes, we explicitly enhance inter-class separation, ensuring that the feature representations of different classes remain directionally independent. This approach strengthens the inherent angular properties of CE loss, facilitating enhanced intra-class compactness and inter-class discriminability within each mini-batch.



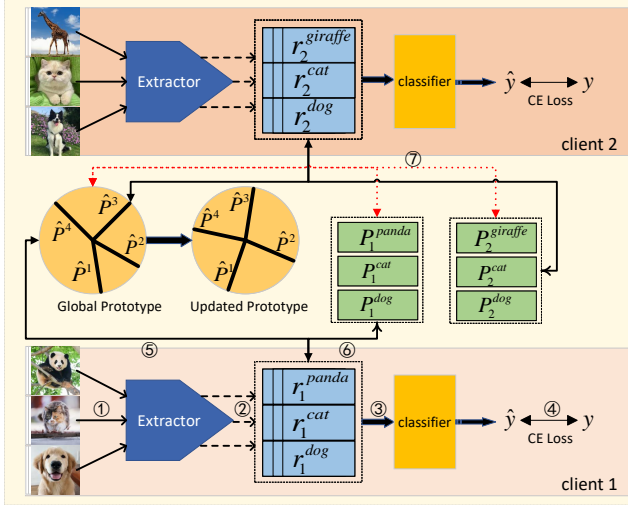


Figure 3. This image illustrates the FedOC framework involving two clients. From ① to ④: We input the samples, get the representations, then input them into the classifier for prediction, and finally calculate the CE loss. ⑤: The dissimilarity between the feature representation and its corresponding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

### 3.4. Local Model Update

In FedOC, 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:

$$\mathcal{L}_k := \mathbb{E}_{(x,y) \sim B} \mathcal{L}_{CE}(h_k(f_k(x; \phi_k); \theta_k), y) + \lambda_c \cdot \mathbb{E}_{(x,y) \sim B} \mathcal{L}_R(r_k^y, \hat{P}^y), \quad (11)$$

where  $\lambda_c$  is a hyperparameter and the regularization is defined as:

$$\mathbb{E}_{(x,y) \sim B} \mathcal{L}_R(r_k^y, \hat{P}^y) = 1 - \frac{1}{|B|} \sum_{(x,y) \in B} \langle r_k^y, \hat{P}^y \rangle, \quad (12)$$

where  $r^y$  represents embedding of the sample. This regularization facilitates the alignment between feature representations and global prototypes across clients, enhancing the global consistency and robustness of the federated model in heterogeneous environments.

### Algorithm 1 The learning process of FedOC

**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.

- 1: **Server:**
- 2: **for** round  $t = 1, \dots, T$  **do**
- 3: Randomly sample a client subset  $\mathcal{I}^t$
- 4: Send  $\hat{\mathcal{P}}$  to clients in  $\mathcal{I}^t$
- 5: **for each** Client  $k \in \mathcal{I}^t$  **in parallel do**
- 6: Client  $k$  performs local training
- 7: **end for**
- 8: Train the global prototype on the server by Eq. 8
- 9: **end for**
- 10: **return** Well-trained client models.
- 11: **Client Local Training:**
- 12: **for** iteration  $e = 1, \dots, E$  **do**
- 13: Local training with the global prototype by Eq. 11
- 14: **end for**
- 15: Collect local prototype  $\{P_k\}$  by Eq. 4
- 16: Send updated prototype  $\{P_k\}$  to the server

### 3.5. FedOC Framework

We present the complete workflow of FedOC 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.

### 4. Convergence Analysis

To analyse the convergence of FedOC, 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  $\mathcal{L}_k$ . Since all client-side loss function share the same property, we omit the subscript  $k$  and denote the loss function as  $\mathcal{L}$ . In order to analyse the convergence of FedOC, we first introduce some additional notes as in the existing framework.

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

$$\|\nabla \mathcal{L}_k^{t_1} - \nabla \mathcal{L}_k^{t_2}\|_2 \leq L_1 \|\omega_k^{t_1} - \omega_k^{t_2}\|_2, \quad (13)$$

where  $\forall t_1, t_2 > 0, k \in \{1, \dots, m\}$ . That also means that local objective function is  $L_1$ -Lipschitz smooth. The function can be locally approximated by a quadratic function:

$$\mathcal{L}_k^{t_1} - \mathcal{L}_k^{t_2} \leq \langle \nabla \mathcal{L}_k^{t_2}, (\omega_k^{t_1} - \omega_k^{t_2}) \rangle + \frac{L_1}{2} \|\omega_k^{t_1} - \omega_k^{t_2}\|_2^2, \quad (14)$$

**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*

$\xi_i$  is a random variable representing the randomness in the data (e.g., mini-batch). Suppose its expectation is

$$\mathbb{E}_{\xi_i \sim D_i}[g_k^t] = \nabla \mathcal{L}(\omega_k^t) = \nabla \mathcal{L}^t, \forall k \in \{1, 2, \dots, m\}, \quad (15)$$

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

$$\mathbb{E}[\|g_k^t - \nabla \mathcal{L}(\omega_k^t)\|_2^2] \leq \sigma^2, \forall k \in \{1, 2, \dots, m\}. \quad (16)$$

**Assumption 3.** (Bounded Expectation of Euclidean norm of Stochastic Gradients of the Global Prototype). *The expectation of euclidean norm of the stochastic gradient of the global prototype is bounded by  $G$ :*

$$\mathbb{E}[\|\nabla \mathcal{L}_{server}\|_2] \leq G \quad (17)$$

We denote  $e \in \{0, 1, 2, \dots, E\}$  as the local iteration, and  $t$  as the global round. Here,  $tE$  indicates the time step before prototype aggregation 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 FedOC makes no change to the local model training process, Lemma 1 derived by FedProto [33] still holds.

**Lemma 1.** *Based on Assumption 1 and 2, after clients training one round, the loss of an arbitrary client's local model is bounded by:*

$$\mathbb{E}[\mathcal{L}^{(t+1)E}] \leq \mathcal{L}^{tE+0} - \left(\eta - \frac{L_1\eta^2}{2}\right) \sum_{e=0}^{E-1} \|\mathcal{L}^{tE+e}\|_2^2 + \frac{L_1 E \eta^2}{2} \sigma^2 \quad (18)$$

**Lemma 2.** *Based on Assumption 3, after clients receiving the latest global prototype, the loss of an arbitrary client's local model is bounded by:*

$$\mathbb{E}[\mathcal{L}^{(t+1)E+0}] \leq \mathbb{E}[\mathcal{L}^{(t+1)E}] + \lambda \eta G. \quad (19)$$

**Theorem 1.** *Leveraging Lemma 1 and Lemma 2, we derive Theorem 1, which establishes an upper bound on the expected reduction in loss achieved during a single round of local training:*

$$\mathbb{E}[\mathcal{L}^{(t+1)E+0}] \leq \mathcal{L}^{tE+0} - \left(\eta - \frac{L_1\eta^2}{2}\right) \sum_{e=0}^{E-1} \|\mathcal{L}^{tE+e}\|_2^2 + \frac{\eta^2 L_1 E \sigma^2}{2} + \lambda \eta G. \quad (20)$$

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

*inequality holds:*

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{e=0}^{E-1} \mathbb{E}[\|\mathcal{L}^{tE+e}\|_2^2] \leq \frac{2(\mathcal{L}^{t=0} - \mathcal{L}^*)}{T\eta(2 - L_1\eta)} + \frac{L_1 E \eta \sigma^2}{2 - L_1\eta} + \frac{2\lambda G}{2 - L_1\eta} \leq \epsilon \quad (21)$$

when  $\eta < \frac{2(\epsilon - \lambda G)}{L_1(\epsilon + E\sigma^2)}$  and  $\lambda < \frac{\epsilon}{G}$ . It shows that for an arbitrary client, 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 three widely used image classification datasets: CIFAR-10, CIFAR-100 [11] and Flowers102 [25].

**Baselines.** To evaluate our proposed FedOC, we compare it with seven popular methods that are applicable in HtFL, including LG-FedAvg [19], FML [29], FedGen [44], FedKD [39], FedProto [33], FedGH [41] and FedTGP [43].

**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<sub>3</sub> (4-layer CNN [24], GoogleNet [30] and MobileNet\_v2 [27]), HMG<sub>5</sub> (ResNet18/34/50/101/152 [6]), HMG<sub>8</sub> (Combines HMG<sub>3</sub> and HMG<sub>5</sub>), and HMG<sub>10</sub> (Extends HMG<sub>8</sub> with DenseNet121 [9] and EfficientNet-B0 [32]) [43]. 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. 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  $\alpha$  controls the strength of heterogeneity. Notably, a smaller  $\alpha$  implies a higher non-IID in the 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  $\rho = 1$ . 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 size is set to 32 for all experiments. For model heterogeneity robustness, we use the HMG<sub>8</sub> by de-

Table 1. The test accuracy (%) on three datasets in the pathological and practical settings using the HtFE<sub>8</sub> model group.

Settings	Pathological Setting			Practical Setting		
Datasets	Cifar10	Cifar100	Flowers102	Cifar10	Cifar100	Flowers102
FML	82.56 ± 0.14	50.27 ± 0.09	52.17 ± 0.98	91.85 ± 0.24	46.45 ± 0.17	48.51 ± 0.57
LG-FedAvg	83.63 ± 0.09	55.51 ± 0.10	58.90 ± 0.22	92.34 ± 0.06	49.82 ± 0.16	53.65 ± 0.46
FedGen	83.63 ± 0.24	55.37 ± 0.35	59.39 ± 0.19	92.43 ± 0.11	49.90 ± 0.21	54.59 ± 0.34
FedProto	80.63 ± 0.05	51.89 ± 0.31	54.48 ± 0.20	79.32 ± 0.22	42.77 ± 0.13	26.66 ± 0.57
FedKD	83.41 ± 0.66	53.01 ± 1.48	52.61 ± 0.74	91.85 ± 0.31	48.95 ± 1.24	51.44 ± 0.75
FedGH	83.49 ± 0.29	55.25 ± 0.12	60.15 ± 1.08	92.66 ± 0.10	49.52 ± 0.05	54.24 ± 0.50
FedTGP	81.60 ± 0.30	52.17 ± 0.74	52.43 ± 0.64	82.48 ± 1.03	47.96 ± 0.04	44.56 ± 0.45
FedOC	<b>85.71 ± 0.10</b>	<b>61.12 ± 0.15</b>	<b>64.15 ± 1.66</b>	<b>94.01 ± 0.03</b>	<b>55.33 ± 0.23</b>	<b>60.34 ± 0.48</b>

fault. For statistical heterogeneity testing, under the pathological setting, we distribute unbalanced data of 2/10/10 classes to each client from a total of 10/100/102 classes on Cifar10/Cifar100/Flowers102 datasets. In the practical setting, a Dirichlet distribution with  $\alpha = 0.05$  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.

**Evaluation Metrics.** We measure the average test accuracy across all clients’ local 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 three datasets is presented in Tab. 1. FedOC consistently outperforms all baselines on these datasets, achieving up to a 15.78% gain over FedTGP in practical settings and 11.72% in pathological settings on Flowers102. This improvement stems from FedOC’s ability to reduce similarity among class representations within the feature space, enhancing classification efficacy more effectively than methods that merely increase Euclidean distance for class separation in high-dimensional spaces. Moreover, FedOC excels in more challenging tasks, where a larger number of classes entails more client prototypes. Its superior performance with greater class diversity is largely due to its orthogonality constraints, which improve inter-class separation within feature spaces.

## 5.3. Impact of Model Heterogeneity

To examine the impact of model heterogeneity in HtFL, we assess the performance of FedOC on three additional HMG settings. We show results in Tab. 2, our findings indicate that all baselines perform well under the HMG<sub>3</sub> condition, as the reduced model heterogeneity significantly lessened its adverse impact on performance, while all methods per-

Table 2. The test accuracy (%) on Cifar100 with three different HMGs to test robustness to heterogeneous models.

Settings	Heterogenous Model Groups		
	HMG <sub>3</sub>	HMG <sub>5</sub>	HMG <sub>10</sub>
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	<b>61.05 ± 0.04</b>	46.24 ± 2.55	47.57 ± 2.00
FedGH	59.21 ± 0.15	50.76 ± 0.30	50.41 ± 0.19
FedTGP	57.12 ± 0.15	50.18 ± 0.20	48.71 ± 0.45
FedOC	60.64 ± 0.50	<b>61.58 ± 0.42</b>	<b>58.54 ± 0.65</b>

form worse with larger model heterogeneity. As the degree of model heterogeneity rises, the dominance of FedOC becomes more pronounced, up to 10.12% at HMG<sub>5</sub> and 8.13% at HMG<sub>10</sub>. In addition, FedOC also has the smallest performance reduction when model heterogeneity increases, which shows that FedOC is robust to model heterogeneity.

## 5.4. Robustness to Participation Rate

To compare FedOC against baselines under varying total client numbers  $M$  and client participation rates  $\rho$ , we design three distinct settings. The results, shown in Tab. 3, ensure a consistent number of participating clients per round. FedOC 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.

## 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, FedOC ensures optimal inter-class separation by leveraging orthogonality constraints, reducing embeddings overlap even in highly skewed data distributions. This demonstrates its effectiveness in promoting generalization

Table 3. The test accuracy (%) on Cifar100 with three participation rate settings. “-” means the baseline can’t converge.

Settings	Different $M$ Clients and Join Ratio		
	$M=40, \rho=50\%$	$M=80, \rho=25\%$	$M=100, \rho=20\%$
FML	$33.96 \pm 0.25$	$19.72 \pm 0.11$	$30.65 \pm 0.25$
LG-FedAvg	$47.05 \pm 0.09$	$39.59 \pm 0.23$	$40.98 \pm 0.51$
FedGen	$47.05 \pm 0.40$	$39.76 \pm 0.35$	$41.10 \pm 0.53$
FedProto	$38.30 \pm 0.53$	$13.49 \pm 0.84$	$16.16 \pm 0.55$
FedKD	$38.12 \pm 3.36$	$21.76 \pm 1.60$	$34.59 \pm 2.73$
FedGH	$46.82 \pm 0.24$	$40.08 \pm 0.35$	-
FedTGP	$42.55 \pm 0.65$	$26.02 \pm 0.54$	$27.27 \pm 0.23$
FedOC	<b><math>54.88 \pm 0.45</math></b>	<b><math>44.86 \pm 0.11</math></b>	<b><math>43.22 \pm 0.45</math></b>

Table 4. Test the accuracy (%) of two data distributions with various degrees of non-IID on the Cifar100 dataset.

Settings	Pathological Setting		Practical Setting	
	5 classes/client	20 classes/client	dir=0.1	dir=0.01
FML	$66.40 \pm 0.33$	$33.13 \pm 0.24$	$37.30 \pm 0.09$	$61.17 \pm 0.24$
LG-FedAvg	$71.84 \pm 0.17$	$37.22 \pm 0.36$	$39.97 \pm 0.11$	$66.06 \pm 0.14$
FedGen	$71.68 \pm 0.09$	$37.47 \pm 0.09$	$40.33 \pm 0.37$	$66.06 \pm 0.08$
FedProto	$70.27 \pm 0.07$	$35.15 \pm 0.18$	$34.30 \pm 0.16$	$56.97 \pm 0.36$
FedKD	$69.75 \pm 1.35$	$34.24 \pm 0.79$	$39.81 \pm 1.12$	$63.86 \pm 1.46$
FedGH	$71.42 \pm 0.29$	$36.99 \pm 0.23$	$39.90 \pm 0.21$	$65.90 \pm 0.27$
FedTGP	$70.20 \pm 0.06$	$36.83 \pm 0.22$	$35.65 \pm 0.18$	$63.00 \pm 0.35$
FedOC	<b><math>75.95 \pm 0.24</math></b>	<b><math>45.45 \pm 0.34</math></b>	<b><math>44.70 \pm 0.29</math></b>	<b><math>70.32 \pm 0.36</math></b>

in statistical heterogeneity environments.

## 5.6. Impact of Feature Dimension

To evaluate the robustness of different algorithms across feature dimensions  $K$ , we conducted experiments with three feature dimensions: 128, 256, and 1024, as depicted in Fig. 4. Results indicate that all baselines maintain a high degree of robustness, showing minimal variation in performance.

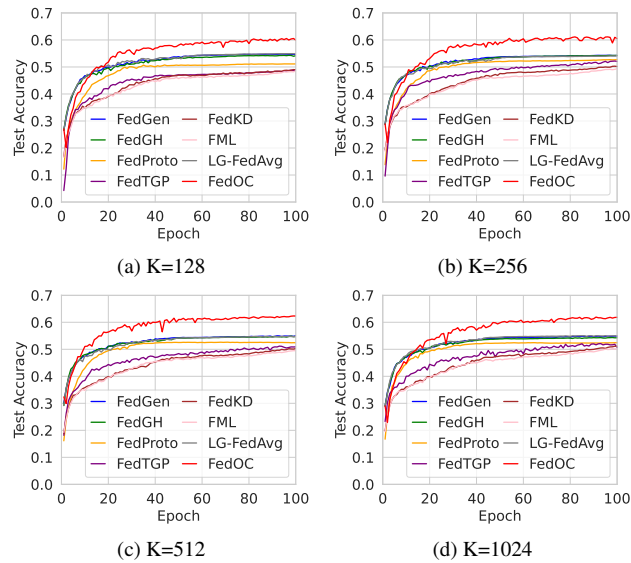


Figure 4. Test accuracy with different feature dimension

## 5.7. Ablation Study

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

Table 5. Test the accuracy (%) on Cifar100 of FedProto, FedOC w/o orthogonality constraints and FedOC.

	FedTGP	FedOC w/o OC	FedOC
Cifar10	$81.60 \pm 0.30$	$78.05 \pm 0.52$	<b><math>85.71 \pm 0.10</math></b>
Cifar100	$52.17 \pm 0.74$	$35.11 \pm 0.63$	<b><math>61.12 \pm 0.15</math></b>
Flowers102	$52.43 \pm 0.64$	$42.43 \pm 0.57$	<b><math>64.15 \pm 1.66</math></b>

## 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  $\lambda_s$  and  $\gamma$ , 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  $\lambda_c$  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. Test the accuracy (%) on Cifar100 of FedOC with various parameters combination.

$\lambda_c$	$\lambda_s$	$\gamma$	Acc	$\lambda_c$	$\lambda_s$	$\gamma$	Acc	$\lambda_c$	$\lambda_s$	$\gamma$	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
<b>100.0</b>	<b>1.0</b>	<b>10.0</b>	<b>61.15</b>	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, FedOC, which employs orthogonality constraints on global prototypes at the server side to improve inter-prototype



separation and maximize directional independence while preserving semantic integrity. On the client side, FedOC aligns the embeddings with prototypes, enhancing its compatibility with CE loss. This approach overcomes the limitations of FedTGP by enhancing prototype separation through angular divergence rather than merely increasing Euclidean distances between prototypes. Extensive experiments indicate that FedOC surpasses baseline methods, achieving superior accuracy and robustness across a range of heterogeneous scenarios.

## 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] 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. 2
- [4] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*, 2020. 2
- [5] Ruchi Gupta and Tanweer Alam. Survey on federated-learning approaches in distributed environment. *Wireless personal communications*, 125(2):1631–1652, 2022. 1
- [6] 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
- [7] 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
- [8] 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
- [9] 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
- [10] 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. 2
- [11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009. 6
- [12] 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
- [13] José Lezama, Qiang Qiu, Pablo Musé, and Guillermo Sapiro. Ole: Orthogonal low-rank embedding-a plug and play geometric loss for deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8109–8118, 2018. 3
- [14] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019. 3
- [15] 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. 2
- [16] 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
- [17] 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. 2
- [18] 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. 3
- [19] 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
- [20] 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
- [21] 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
- [22] 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. 3
- [23] 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
- [24] 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. 2, 6
- [25] 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
- [26] 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. 2, 3
- [27] 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
- [28] Meyer Scetbon, Michael Elad, and Peyman Milanfar. Deep k-svd denoising. *IEEE Transactions on Image Processing*, 30:5944–5955, 2021. 3
- [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] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in neural information processing systems*, 33:21394–21405, 2020. 2
- [32] 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
- [33] 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, 6
- [34] 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
- [35] 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
- [36] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 2
- [37] 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
- [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] 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
- [41] 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, 3, 6
- [42] 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
- [43] 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
- [44] 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. 3, 6