Federated Learning under Partially Class-Disjoint Data via Manifold Reshaping

Ziqing Fan, Jiangchao Yao, Ruipeng Zhang
{zqfan_knight, sunarker, zhangrp}@sjtu.edu.cn
Cooperative Medianet Innovation Center, Shanghai Jiao Tong University
Shanghai AI Laboratory

Lingjuan Lyu
Lingjuan.Lv@sony.com
Sony AI

Ya Zhang, Yanfeng Wang
{ya_zhang, wangyanfeng}@sjtu.edu.cn
Cooperative Medianet Innovation Center, Shanghai Jiao Tong University
Shanghai AI Laboratory

Reviewed on OpenReview: https://openreview.net/forum?id=jLTqJXAg7

Abstract

Statistical heterogeneity severely limits the performance of federated learning (FL), motivating several explorations e.g., FedProx, MOON and FedDyn, to alleviate this problem. Despite effectiveness, their considered scenario generally requires samples from almost all classes during the local training of each client, although some covariate shifts may exist among clients. In fact, the natural case of partially class-disjoint data (PCDD), where each client contributes a few classes (instead of all classes) of samples, is practical yet underexplored. Specifically, the unique collapse and invasion characteristics of PCDD can induce the biased optimization direction in local training, which prevents the efficiency of federated learning. To address this dilemma, we propose a manifold reshaping approach called FedMR to calibrate the feature space of local training. Our FedMR adds two interplaying losses to the vanilla federated learning: one is intra-class loss to decorrelate feature dimensions for anti-collapse; and the other one is inter-class loss to guarantee the proper margin among categories in the feature expansion. We conduct extensive experiments on a range of datasets to demonstrate that our FedMR achieves much higher accuracy and better communication efficiency. Source code is available at: https://github.com/MediaBrain-SJTU/FedMR.

1 Introduction

Federated learning [McMahan et al. (2017); Li et al. (2020a); Yang et al. (2019)] has drawn considerable attention due to the increasing requirements on data protection [Shokri & Shmatikov (2015); Zhu & Han (2020); Hu et al. (2021); Li et al. (2021c); Lyu et al. (2020)] in real-world applications like medical image analysis (Guo et al. (2021); Park et al. (2021); Yin et al. (2022); Dou et al. (2021); Jiang et al. (2022)) and autonomous driving (Liang et al. (2019); Pokhrel & Choi (2020)). Nevertheless, the resulting challenge of data heterogeneity severely limits the application of machine learning algorithms (Zhao et al. (2018)) in federated learning. This motivations a plenty of explorations to address the statistical heterogeneity issue and improve the efficiency (Kairouz et al. (2021); Wang et al. (2020); Li et al. (2022)).

Existing approaches to address the statistical heterogeneity can be roughly summarized into two categories. One line of research is to constrain the parameter update in local clients or in the central server. For example, FedProx (Li et al. (2020b)), FedDyn (Acar et al. (2020)) and FedDC (Gao et al. (2022)) explore how to reduce the variance or calibrate the optimization by adding the proximal regularization on parameters in FedAvg [McMahan et al. (2017)]. The other line of research focuses on constraining the representation...
from the model to implicitly affect the update. FedProc (Mu et al. 2021) and FedProto (Tan et al. 2022) introduce prototype learning to help local training, and MOON (Li et al. 2021b) utilizes contrastive learning to minimize the distance between representations learned by local model and global model, and maximize the distance between representations learned by local model and previous local model. However, all these methods validate their efficiency mostly under the support from all classes of samples in each client while lacking a well justification on a natural scenario, namely partially class-disjoint data w.r.t. classes.

As illustrated in Figure 1(a), in typical federated learning, each client usually contains all classes of samples but under different covariate shifts, and all clients work together to train a global model. However, in the case of PCDD (Figure 1(b)), there are only a small subset of categories in each client and all clients together provide information of all classes. Such a situation is very common in real-world applications. For example, there are shared and distinct Thyroid diseases in different hospitals due to regional diversity (Gaitan et al. 1991). Hospitals from different regions can construct a federation to learn a comprehensive model for the diagnostic of Thyroid diseases but suffer from the PCDD challenge. We conduct a toy study on a simulated dataset (see details in the Appendix B), and visualize the feature space under centralized training (the left panel of Figure 1(c)) and local training under PCDD (the middle panel of Figure 1(c)) by projecting on the unit sphere. As shown in Figure 1(c), PCDD induces a dimensional collapse onto a narrow area due to the lack of support from all classes, and causes a space invasion to the missing classes. Previous approaches such as FedProc and MOON may implicitly constrain the space invasion by utilizing class prototypes or global features generated from the global model, and methods like FedProx, FedDyn, and FedNova could also help a bit from the view of optimization. However, these methods are not oriented towards the PCDD issue, and they are inefficient to avoid the collapse and invasion characteristics of PCDD and achieve sub-optimal performance from the both view of experimental performance shown in Table 1 and the feature variance of all methods shown in Table 7.

To address this dilemma, we propose a manifold-reshaping approach called FedMR to properly prevent the degeneration caused by the locally class missing. FedMR introduces two interplaying losses: one is intra-class loss to decorrelate feature space for anti-collapse; and another one is the inter-class loss to guarantee the proper margin among categories by means of global class prototypes. The right panel of Figure 1(c) provides a rough visualization of FedMR. Theoretically, we analyze the benefit from the interaction of the intra-class loss and the inter-class loss under PCDD, and empirically, we verify the effectiveness of FedMR compared with the current state-of-the-art methods. Our contributions can be summarized as follows:

- We are among the first attempts to study dimensional collapse and space invasion challenges caused by PCDD in Generic FL that degenerates embedding space and thus limits the model performance.
• We introduce a approach termed as FedMR, which decorrelates the feature space to avoid dimensional collapse and constructs a proper inter-class margin to prevent space invasion. Our theoretical analysis confirms the rationality of the designed losses and their benefits to address the dilemma of PCDD.

• We conduct a range of experiments on multiple benchmark datasets under PCDD and a real-world disease dataset to demonstrate the advantages of FedMR over the state-of-the-art methods. We also develop several variants of FedMR to consider the communication cost and privacy concerns.

2 Related Works

2.1 Federated Learning

There are extensive works to address the statistical heterogeneity in federated learning, which induces the bias of local training due to the covariate shifts among clients (Zhao et al. (2018); Li et al. (2022); Zhang et al. (2023a)). A line of research handles this problem by adding constraints like normalization or regularization on model weights in the local training or in the server aggregation. FedProx (Li et al. (2020b)) utilizes a proximal term to limit the local updates so as to reduce the bias, and FedNova (Wang et al. (2020)) introduces the normalization on total gradients to eliminate the objective inconsistency. FedDyn (Acar et al. (2020)) makes the global model and local models approximately aligned in the limit by proposing a dynamic regularizer for each client at each round. FedDC (Gao et al. (2022)) reduces the inconsistent optimization on the client-side by local drift decoupling and correction. Another line of research focuses on constraining representations from local models and the global model. MOON (Li et al. (2021b)) corrects gradients in local training by contrasting the representation from local model and that from global model. FedProc (Mu et al. (2021)) utilizes prototypes as global information to help correct local representations. Our FedMR is also conducted on representations of samples and classes but follows a totally different problem and spirit.

2.2 Representation Learning

The collapse problem is also an inevitable concern in the area of representation learning. In their research lines, there are several attempts to prevent the potential collapse issues. For example, in self-supervised learning, Barlow Twins, VICReg and Shuffled-DBN (Bardes et al. (2022); Zbontar et al. (2021); Hua et al. (2021)) manipulate the rank of the (co-)variance matrix to prevent the potential collapse. In contrastive learning, DirectCLR (Jing et al. (2021)) directly optimizes the representation space without an explicit trainable classifier to promote a larger feature diversity. In incremental learning, CwD (Shi et al. (2022)) applies a similar technique to prevent dimensional collapse in the initial phase. In our PCDD case, it is more challenging, since we should not only avoid collapse but also avoid the space invasion in the feature space.

2.3 Federated Prototype Learning

In many vision tasks (Snell et al. (2017); Yang et al. (2018); Deng et al. (2021)), prototypes are the mean values of representations of a class and contain information like feature structures and relationships of different classes. Since prototypes are population-level statistics of features instead of raw features, which are relatively safe to share, prototype learning thus has been applied in federated learning. In Generic FL, FedProc (Mu et al. (2021)) utilizes the class prototypes as global knowledge to help correct local training. In Personalized FL, FedProto (Tan et al. (2022)) shares prototypes instead of local gradients to reduce communication costs. We also draw spirits from prototype learning to handle PCDD. Besides, to make fair comparison to methods without prototypes and better reduce extreme privacy concerns, we conduct a range of auxiliary experiments in Section 4.4.

3 The Proposed Method

3.1 Preliminary

PCDD Definition. There are many nonnegligible real-world PCDD scenarios. ISIC2019 dataset (Codella et al. (2018); Tschandl et al. (2018); Combalia et al. (2019)), a region-driven subset of types of Thyroid

3
diseases in the hospital systems, is utilized in our experiments. In landmark detection [20] for thousands of categories with data locally preserved, most contributors only have a subset of categories of landmark photos where they live or traveled before, which is also a scenario for the federated PCDD problem. To make it clear, we first define some notations of the partially class-disjoint data situation in federated learning. Let $C$ denote the collection of full classes and $P$ denote the set of all local clients. Considering the real-world constraints like privacy or environmental limitation, each local client may only own the samples of partial classes. Thus, for the $k$-th client $P_k$, its corresponding local dataset $D_k$ can be expressed as $D_k = \{(x_{k,i}, y_{k,i}) | y_{k,i} = c \in C_k\}$, where $C_k \subseteq C$. The number of samples of the class $c \in C_k$ in $P_k$ is $N_k^c$. We denote a local model $f(\cdot; w_k)$ on all clients as two parts: a backbone network $f_1(\cdot; w_{k,1})$ and a linear classifier $f_2(\cdot; w_{k,2})$. The loss of the $k$-th client can be formulated as

$$\ell_{cls}^k(D_k; w_k) = \frac{1}{N_k} \sum_{i=1}^{N_k} \ell(y_{k,i}, f_2(z_{k,i}; w_{k,2})) |_{z_{k,i} = f_1(x_{k,i}; w_{k,1})},$$

where $z_{k,i}$ is the feature representation of the input $x_{k,i}$ and $\ell(\cdot, \cdot)$ is the loss measure. Under PCDD, we can empirically find the dimensional collapse and the space invasion problems about representation.

**FedAvg.** The vanilla federated learning via FedAvg consists of four steps [2017]: 1) In round $t$, the server distributes the global model $w^t$ to clients that participate in the training; 2) Each local client receives the model and continues to train the model, e.g., the $k$-th client conducts the following,

$$w_k^t \leftarrow w_k^{t-1} - \eta \nabla \ell_k(b_k^t; w_k^t),$$

where $\eta$ is the learning rate, and $b_k^t$ is a mini-batch of training data sampled from the local dataset $D_k$. After $E$ epochs, we acquire a new local model $w_k^E$. 3) The updated models are then collected to the server as \{ $w_1^E, w_2^E, \ldots, w_K^E$ \}; 4) The server performs the following aggregation to acquire a new global model $w^{t+1}$,

$$w^{t+1} \leftarrow \frac{1}{K} \sum_{k=1}^{K} p_k w_k^E,$$

where $p_k$ is the proportion of sample number of the $k$-th client to the sample number of all the participants, i.e., $p_k = N_k / \sum_{k'=1}^{K} N_{k'}$. When the maximal round $T$ reaches, we will have the final optimized model $w^T$.

### 3.2 Motivation

In Figure 2, we illustrate a low-dimensional example to characterize the directional shift of the local training under PCDD on the client side. In the following, we use the parameter aggregation of a linear classification model.
to study the directional shift of global model in the server, which further clarifies the adverse effect of PCDD.

Similar to Figure 2, let $c_1$, $c_2$ and $c_3$ denote three classes of samples on a circular face respectively centered at $(1, 0)$, $(0, \sqrt{3})$ and $(0, -\sqrt{3})$ with radius $r=\frac{1}{2}$ under a uniform distribution. Then, if there are samples of all classes in each local client, we will get the optimal weight for all categories as follows:

$$w^* = \begin{bmatrix} 1 & 0 \\ \frac{-\sqrt{3}}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{\sqrt{3}}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Note that, we omit the bias term in linear classification for simplicity. Conversely, among total three participants, if each participant only has the samples of two classes, e.g., $(c_1, c_2)$, $(c_1, c_3)$ and $(c_2, c_3)$ respectively, then their learned weights can be inferred as follows:

$$w^*_1, w^*_2, w^*_3 = \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}.$$

After the server aggregation, we have the estimated weight

$$\hat{w}^* = \begin{bmatrix} \frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}+2}{\sqrt{2}} \\ -\frac{1}{2} & -\frac{\sqrt{3}+2}{\sqrt{2}} \end{bmatrix}.$$

Then, we can find that except the difference on amplitude between $w^*$ and $\hat{w}^*$, a more important issue is the optimization direction for $c_2$ (or $c_3$) shifts about $45^\circ$ by computing the angle between vector $(-\frac{\sqrt{3}}{2}, \frac{\sqrt{3}}{2})$ and vector $(-\frac{1}{6}, \frac{\sqrt{3}+2}{6})$ (or between vector $(-\frac{\sqrt{3}}{2}, -\frac{1}{2})$ and vector $(-\frac{1}{6}, -\frac{\sqrt{3}+2}{6})$). Actually, the angle shift can be enlarged in some real-world applications, when the hard negative classes are missing. However, if we can have the statistical centroid of the locally missing class in the local client, namely $c_3$ in the case of Figure 2, it is easy to find that the inferred optimal $\hat{w}^*$ is same to $w^*$ as the decision plane can be normally characterized with the support of the single point $c_3$. This inspires us to design the subsequent method.

### 3.3 Manifold Reshaping

As the aforementioned analysis, PCDD in federated learning leads to the directional shift of optimization both in the local models and in the global model. An empirical explanation is that the feature representation of the specific class that should support classification is totally missing, inducing the feature representation of other observed classes arbitrarily distributes as a greedy collapsed manifold, as shown in Figure 1(c). To address this problem, we explore a manifold-reshaping method from both the intra-class perspective and the inter-class perspective. In the following, we will present two interplaying losses and our framework.

#### 3.3.1 Intra-Class Loss

The general way to prevent the representation from collapsing into a low-dimensional manifold, is to decorrelate dimensions for different patterns and expand the intrinsic dimensionality of each category. Such a goal can be achieved by manipulating the rank of the covariance matrix regarding representation. Specifically, for each client, we can first compute the class-level normalization for the representation $z^c_{k,i} \in \mathbb{R}^d$ as $\bar{z}^c_{k,i} = \frac{z^c_{k,i} - \mu^c_k}{\sigma^c_k}$, where $\mu^c_k$ and $\sigma^c_k$ are the mean and standard deviation of features belonging to class $c$ and calculated as: $\mu^c_k = \frac{1}{n_{k,c}} \sum_{i=1}^{n_{k,c}} z^c_{k,i}$ and $\sigma^c_k = \sqrt{\frac{1}{n_{k,c}} \sum_{i=1}^{n_{k,c}} (z^c_{k,i} - \mu^c_k)^2}$. Then, we compute an intra-class covariance matrix based on the above normalization for each observed class in the $k$-th client:

$$M^c_k = \frac{1}{N^c_k - 1} \sum_{i=1}^{N^c_k} (\bar{z}^c_{k,i})^T (\bar{z}^c_{k,i}) \tag{5}.$$

\[\text{Note that, we would like to point out that in the real-world case, we cannot acquire such statistical centroid in advance but have to resort to the training process along with the special technique design.}\]
Since each eigenvalue of $M^c_k \in \mathbb{R}^{d \times d}$ characterizes the importance of a feature dimension within class, we can make them distributed uniformly to prevent the dimensional collapse of each observed class. However, considering the learnable pursuit of machine learning algorithms, we actually cannot directly optimize eigenvalues to reach this goal. Fortunately, it is possible to use an equivalent objective as an alternative, which is clarified by the following lemma.

**Lemma 1.** Assuming a covariance matrix $M \in \mathbb{R}^{d \times d}$ computed from the feature of each sample with the standard normalization, and its eigenvalues \{\lambda_1, \lambda_2, ..., \lambda_d\}, we will have the following equality that satisfied

$$\sum_{i=1}^{d} (\lambda_i - \frac{1}{d} \sum_{j=1}^{d} \lambda_j)^2 = ||M||^2_F - d.$$ 

The complete proof is summarized in the Appendix A.2. From Lemma 1 we can see that pursuing the uniformity of the eigenvalues for the covariance matrix can transform into minimizing the Frobenius norm of the covariance matrix. Therefore, our intra-class loss to prevent the undesired dimensional collapse for observed classes is formulated as

$$\ell_{\text{intra}}^k = \frac{1}{|C_k|} \sum_{c \in C_k} ||M^c_k||^2_F.$$ 

(3)

### 3.3.2 Inter-Class Loss

Although the intra-class loss helps decorrelate the feature dimensions to prevent collapse, the resulting space invasion for the missing classes can be concomitantly exacerbated. Thus, it is important to guarantee the proper space of the missing classes in the expansion as encouraged by equation 3. To address this problem, we maintain a series of global class prototypes and transmit them to local clients as support of the missing classes in the feature space. Concretely, we first compute the class prototypes in the \(k\)-th client as the average of feature representations (Snell et al. (2017); Yang et al. (2018); Deng et al. (2021)):

$$g^c_k \leftarrow \frac{1}{N^c_k} \sum_{i=1}^{N^c_k} z^c_{k,i} \text{ for } c \in C_k$$

Then, all client-level prototypes are submitted to the server along with local models in federated learning. In the central server, the global prototypes for all classes are updated as

$$g^t_c \leftarrow \frac{K}{\sum_{k=1}^{K} N^c_k} g^c_k \text{ for } c \in C$$

where \(g^t_c\) is the global prototype of the \(c\)-th class in round \(t\) and \(p^c_k = \frac{N^c_k}{\sum_{k=1}^{K} N^c_k}\). In the next round, the central server distributes the global prototypes to all clients as the references to avoid the space invasion. Formally, we construct the following margin loss by contrasting the distances from prototypes to the representation of the sample.

$$\ell_{\text{inter}}^k = \frac{1}{|C_k||C_k|-1} \sum_{c_i \in C_k} \sum_{c_j \in C_k \setminus c_i} D_{c_i, c_j},$$

(4)

where \(D_{c_i, c_j}\) is defined as:

$$D_{c_i, c_j} = \frac{1}{N^c_k} \sum_{n=1}^{N^c_k} \max\{||z^c_{k,n} - g^t_{c_i}|| - ||z^c_{k,n} - g^t_{c_j}||, 0\}.$$ 

In the following, we use a theorem to show how inter-class loss jointly with intra-class loss makes the representation of the missing classes approach to the optimal.
Figure 3: The framework of FedMR. On the client side, except the vanilla training with the classification loss, the manifold-reshaping parts, i.e., the intra-class loss and the inter-class loss, respectively help conduct the feature decorrelation to avoid the dimensional collapse, and leverage the global prototypes to construct the proper margin among classes to prevent the space invasion. On the server side, except the model aggregation, the global class prototypes are also the reference for missing classes participating in the local training.

**Algorithm 1** FedMR

**Input:** a set of $K$ clients that participate in each round, the initial model weights $w^0$, the maximal round $T$, the learning rate $\eta$, the local training epochs $E$.

for $t = 0, 1, \ldots, T - 1$ do

randomly sample $K$ clients updates global model weights and global class prototypes ($w^t \leftarrow \sum_{k=1}^{K} \tilde{p}_k^t w_{k}^{t-1}$) for all clients.
distribute $w^t$ and $G_c \{g_1^t, g_2^t, \ldots, g_C^t\}$ to the $K$ clients.
do in parallel for $\forall k \in K$ clients

$w_k^t \leftarrow w^t$.

for $\tau = 0, 1, \ldots, E - 1$ do

sample a mini-batch from local dataset and perform updates ($w_k^t \leftarrow w_k^t - \eta \nabla L_k(b_k^t, G_c; w_k^t)$).
end for

update local class prototypes ($\forall c$, $g_k^t \leftarrow \sum_{i=1}^{n_k^t} \frac{1}{n_k^t} z_{c,i}^t$), and submit $w_k^t$ and $\{g_1^t, g_2^t, \ldots, g_C^t\}$ to server.
end in parallel

end for

**Theorem 1.** Let the global optimal representation for class $c$ be $g_c^* = [a_{c,1}^*, \ldots, a_{c,d}^*]$, and $z_{k,i}^{c,t}$ be the representation of sample $x$ in the class $c$ of the $k$-th client. Assuming that $\forall i$, both $|a_{c,i}^*|$ and $z_{k,i}^{c,t}$ are upper bounded by $G$, and all dimensions are disentangled, in round $t$, the $i$-th dimension of local representation $z_{k,i}^{c,t}$ satisfies

$$|z_{k,i}^{c,t} - a_{c,i}^*| \leq 2(1 - \tilde{p}_k^t \Gamma)G + \delta \Gamma,$$

where $\tilde{p}_k^t$ is the accumulation regarding the $i$-th dimension of the class-$c$ prototype, $\Gamma = \frac{1 - (p_c^t)^t}{1 - p_c^t}$, $(p_c^t)^t$ refers to the $p_c^t$ raised to the power of $t$, and $\delta$ is the maximum margin of the inter-loss term.

Note that, Theorem 1 shows five critical points: 1) The proof of the theorem requires each dimension of the representation to be irrelevant to each other, which is achieved by the intra-class loss. Although disentanglement of dimensions might not be totally achieved in practical, empirically, we find that the intra-class loss converges and maintains a relatively low value easily, meaning that the model achieves good
decorrelation. In the Appendix, we show the training curve of intra-class loss during the federated training. 2) In the theorem, $\delta$ is a trade-off of training stability and theoretical results determined by the margin. The larger the margin, the larger the $\delta$, however the more stable the local training is. This is because the global prototypes are not very accurate in the early stage of local training and directly minimizing the distance of samples to their global class prototypes can bring side effect to the feature diversity. When the margin is removed ($D_{c_i,c_j} = \frac{1}{N_c} \sum_{n=1}^{N_c} ||z_{c_k,n}^{c_i} - g_{c_i}||$), $\delta$ will be zero. 3) Without considering $\delta$, as $t$ increases, $\Gamma$ is smaller and representation $z_{c_k}^{c_i}$ is closer to global optimal prototype $g^*_c$, showing the promise of our method. 4) When $t$ is large enough, we can get an upper bound $2 \frac{\rho_c - \hat{p}_c}{1 - \hat{p}_c} G$, meaning more clients with the specific dimensional information participating in the training, the tighter the upper bound is. When all other clients can provide the support ($\hat{p}_c = 1 - p_c$), the error will be 0. 5) While $\hat{p}_c$ denotes the proportions of a subset of clients that can provide the support information for this dimension, the theoretical result depends on the class distribution and overlap across the clients. The complete proof is summarized in the Appendix A.1.

### 3.3.3 The Total Framework

After introducing the intra-class loss and the inter-class loss, we give the total framework of FedMR. On the client side, local models are trained on their partially class-disjoint datasets. Through manifold-reshaping loss, dimensions of the representation are decorrelated and local class subspace is corrected to prevent the space invasion, and gradually approach the global space partition. The total local objective including the vanilla classification loss can be written as

$$L_k = \ell^{\text{cls}}_k + \left( \mu_1 \ell^{\text{intra}}_k + \mu_2 \ell^{\text{inter}}_k \right)_{\text{manifold reshaping}},$$

where $\mu_1$ and $\mu_2$ are the balancing hyperparameters and will be discussed in the experimental part. In Figure 3, we illustrate the corresponding structure of FedMR and formulate the training procedure in Algorithm 1. In terms of the privacy concerns about the prototype transmission and the communication cost, we will give the comprehensive analysis on FedMR about these factors.

### 4 Experiment

#### 4.1 Experimental Setup

**Datasets.** We adopt four popular benchmark datasets SVHN (Netzer et al. (2011)), FMNIST (Xiao et al. (2017)), CIFAR10 and CIFAR100 (LeCun et al. (1998)) in federated learning and a real-world PCDD medical dataset ISIC2019 (Codella et al. (2018); Tschandl et al. (2018); Combalia et al. (2019)) to conduct experiments. Regarding the data setup, although Dirichelet Distribution is popular to split data in FL, it usually generates diverse imbalance data coupled with occasionally PCDD. In order to better study pure PCDD, for the former four benchmarks, we split each dataset into $\rho$ clients, each with $\varsigma$ categories, abbreviated as $P\rhoC\varsigma$. For example, P10C10 in CIFAR10 means that we split CIFAR10 into 10 clients, each with 10 classes. Please refer to the detailed explanations and strategies in the Appendix. ISIC2019 is a real-world federated application under the PCDD situation and the data distribution among clients is shown in the Appendix C.1.3. We follow the settings in Flamby benchmark (Terrail et al. (2022)).

**Implementation.** We compare FedMR with FedAvg (McMahan et al. (2017)) and multiple state-of-the-arts including FedProx (Li et al. (2020b)), FedProc (Karimireddy et al. (2020)), FedNova (Li et al. (2021b)), MOON (Wang et al. (2020)), FedDyn (Acar et al. (2020)) and FedDC (Gao et al. (2022)). To make a fair and comprehensive comparison, we utilize the same model for all approaches and three model structures for different datasets: ResNet18 (He et al. (2016)) (follow (Li et al. (2021b, 2022))) for SVHN, FMNIST and CIFAR10, wide ResNet (Zagoruyko & Komodakis (2016)) for CIFAR100 and EfficientNet (Tan & Le (2019)) for ISIC2019. The optimizer is SGD with a learning rate 0.01, the weight decay $10^{-5}$ and momentum 0.9. The batch size is set to 128 and the local updates are set to 10 epochs for all approaches. The detailed information of the model and training parameters are given in the Appendix.
Table 1: Performance of FedMR and a range of state-of-the-art approaches on four datasets under PCDD partitions. Datasets are divided into $q$ clients and each client has $c$ classes (denoted as $P_{qC_c}$). We compute average accuracy of all partitions, highlight results of FedMR, underline results of best baseline, and show the improvement of FedMR to FedAvg (subscript of FedMR) and to the best baseline ($\Delta$).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FMNIST</td>
<td>P5C2</td>
<td>67.29</td>
<td>69.60</td>
<td>66.28</td>
<td>66.87</td>
<td>66.82</td>
<td>71.01</td>
<td>68.50</td>
<td>75.51$\uparrow$</td>
<td>+4.50</td>
</tr>
<tr>
<td></td>
<td>P10C2</td>
<td>67.33</td>
<td>67.76</td>
<td>69.08</td>
<td>48.44</td>
<td>67.93</td>
<td>67.16</td>
<td>67.36</td>
<td>74.97$\uparrow$</td>
<td>+5.89</td>
</tr>
<tr>
<td></td>
<td>P10C3</td>
<td>81.67</td>
<td>80.93</td>
<td>82.06</td>
<td>83.20</td>
<td>83.42</td>
<td>83.00</td>
<td>83.24</td>
<td>83.55$\uparrow$</td>
<td>+0.13</td>
</tr>
<tr>
<td></td>
<td>P10C5</td>
<td>88.53</td>
<td>89.22</td>
<td>89.23</td>
<td>88.86</td>
<td>88.98</td>
<td>88.38</td>
<td>89.22</td>
<td>90.04$\uparrow$</td>
<td>+0.81</td>
</tr>
<tr>
<td></td>
<td>IID</td>
<td>91.93</td>
<td>91.95</td>
<td>92.06</td>
<td>91.84</td>
<td>91.12</td>
<td>91.76</td>
<td>92.15</td>
<td>92.19$\uparrow$</td>
<td>+0.04</td>
</tr>
<tr>
<td></td>
<td>avg</td>
<td>79.35</td>
<td>79.89</td>
<td>79.74</td>
<td>78.65</td>
<td>79.87</td>
<td>80.26</td>
<td>80.09</td>
<td>83.45$\uparrow$</td>
<td>+3.19</td>
</tr>
<tr>
<td>SVHN</td>
<td>P5C2</td>
<td>81.85</td>
<td>81.83</td>
<td>79.54</td>
<td>81.11</td>
<td>81.60</td>
<td>79.89</td>
<td>81.63</td>
<td>83.10$\uparrow$</td>
<td>+1.27</td>
</tr>
<tr>
<td></td>
<td>P10C2</td>
<td>78.92</td>
<td>79.60</td>
<td>78.75</td>
<td>66.86</td>
<td>79.83</td>
<td>76.24</td>
<td>78.96</td>
<td>82.47$\uparrow$</td>
<td>+2.64</td>
</tr>
<tr>
<td></td>
<td>P10C3</td>
<td>87.70</td>
<td>87.40</td>
<td>88.13</td>
<td>87.50</td>
<td>87.83</td>
<td>87.27</td>
<td>88.05</td>
<td>89.13$\uparrow$</td>
<td>+1.00</td>
</tr>
<tr>
<td></td>
<td>P10C5</td>
<td>91.20</td>
<td>91.24</td>
<td>91.63</td>
<td>92.09</td>
<td>91.16</td>
<td>90.17</td>
<td>91.64</td>
<td>92.18$\uparrow$</td>
<td>+0.09</td>
</tr>
<tr>
<td></td>
<td>IID</td>
<td>92.74</td>
<td>92.89</td>
<td>93.57</td>
<td>92.62</td>
<td>93.12</td>
<td>92.26</td>
<td>92.90</td>
<td>93.04$\uparrow$</td>
<td>-0.53</td>
</tr>
<tr>
<td></td>
<td>avg</td>
<td>86.48</td>
<td>86.59</td>
<td>86.32</td>
<td>83.87</td>
<td>86.71</td>
<td>85.17</td>
<td>86.64</td>
<td>87.98$\uparrow$</td>
<td>+1.27</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>P5C2</td>
<td>67.68</td>
<td>68.18</td>
<td>69.27</td>
<td>67.57</td>
<td>66.86</td>
<td>69.64</td>
<td>69.18</td>
<td>74.19$\uparrow$</td>
<td>+4.55</td>
</tr>
<tr>
<td></td>
<td>P10C2</td>
<td>67.27</td>
<td>71.09</td>
<td>67.02</td>
<td>57.79</td>
<td>67.61</td>
<td>67.74</td>
<td>67.44</td>
<td>73.32$\uparrow$</td>
<td>+2.23</td>
</tr>
<tr>
<td></td>
<td>P10C3</td>
<td>77.82</td>
<td>77.89</td>
<td>77.87</td>
<td>77.22</td>
<td>78.42</td>
<td>77.99</td>
<td>77.79</td>
<td>82.79$\uparrow$</td>
<td>+4.33</td>
</tr>
<tr>
<td></td>
<td>P10C5</td>
<td>88.22</td>
<td>88.34</td>
<td>88.19</td>
<td>88.20</td>
<td>88.00</td>
<td>88.35</td>
<td>88.14</td>
<td>89.00$\uparrow$</td>
<td>+0.71</td>
</tr>
<tr>
<td></td>
<td>IID</td>
<td>91.88</td>
<td>92.14</td>
<td>92.62</td>
<td>92.37</td>
<td>92.56</td>
<td>92.29</td>
<td>92.85</td>
<td>93.00$\uparrow$</td>
<td>+0.21</td>
</tr>
<tr>
<td></td>
<td>avg</td>
<td>78.58</td>
<td>78.53</td>
<td>78.90</td>
<td>76.63</td>
<td>78.69</td>
<td>79.20</td>
<td>79.15</td>
<td>82.48$\uparrow$</td>
<td>+2.95</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>P10C10</td>
<td>54.31</td>
<td>54.79</td>
<td>54.69</td>
<td>54.45</td>
<td>54.98</td>
<td>55.94</td>
<td>54.73</td>
<td>57.27$\uparrow$</td>
<td>+1.33</td>
</tr>
<tr>
<td></td>
<td>P10C20</td>
<td>64.81</td>
<td>65.37</td>
<td>64.98</td>
<td>65.79</td>
<td>65.75</td>
<td>65.02</td>
<td>65.21</td>
<td>65.81$\uparrow$</td>
<td>+0.02</td>
</tr>
<tr>
<td></td>
<td>P10C30</td>
<td>69.35</td>
<td>69.75</td>
<td>69.64</td>
<td>69.55</td>
<td>69.51</td>
<td>69.84</td>
<td>69.38</td>
<td>70.24$\uparrow$</td>
<td>+0.40</td>
</tr>
<tr>
<td></td>
<td>P10C50</td>
<td>71.28</td>
<td>71.35</td>
<td>72.13</td>
<td>71.25</td>
<td>71.54</td>
<td>71.25</td>
<td>72.11</td>
<td>72.17$\uparrow$</td>
<td>+0.04</td>
</tr>
<tr>
<td></td>
<td>IID</td>
<td>72.28</td>
<td>72.55</td>
<td>73.07</td>
<td>72.66</td>
<td>73.01</td>
<td>73.04</td>
<td>72.77</td>
<td>72.79$\uparrow$</td>
<td>-0.28</td>
</tr>
<tr>
<td></td>
<td>avg</td>
<td>66.41</td>
<td>66.76</td>
<td>66.90</td>
<td>66.74</td>
<td>66.96</td>
<td>67.22</td>
<td>66.84</td>
<td>67.66$\uparrow$</td>
<td>+0.44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR100</td>
<td>P10C10</td>
<td>54.31</td>
<td>54.79</td>
<td>54.69</td>
<td>54.45</td>
<td>54.98</td>
<td>55.94</td>
<td>54.73</td>
<td>57.27$\uparrow$</td>
<td>+1.33</td>
</tr>
<tr>
<td></td>
<td>P5C10</td>
<td>49.84</td>
<td>51.17</td>
<td>51.94</td>
<td>50.22</td>
<td>52.19</td>
<td>50.53</td>
<td>51.17</td>
<td>53.36$\uparrow$</td>
<td>+1.17</td>
</tr>
<tr>
<td></td>
<td>P10C100</td>
<td>47.90</td>
<td>48.48</td>
<td>49.01</td>
<td>48.07</td>
<td>48.94</td>
<td>49.24</td>
<td>48.76</td>
<td>49.60$\uparrow$</td>
<td>+0.36</td>
</tr>
<tr>
<td>ISIC2019</td>
<td>Real</td>
<td>73.14</td>
<td>75.41</td>
<td>75.26</td>
<td>73.62</td>
<td>75.46</td>
<td>75.07</td>
<td>75.25</td>
<td>76.55$\uparrow$</td>
<td>+1.09</td>
</tr>
</tbody>
</table>

4.2 Performance under PCDD

In this part, we compare FedMR with FedAvg and other methods on FMNIST, SVHN, CIFAR10 and CIFAR100 datasets under partially class-disjoint situation. Note that, in FedProg, MOON, FedDyn, FedProc, FedDC and our method, there are parameters that need to set. We use grid search to choose the best parameters for each method. See more concrete settings in the Appendix C.2 and Section 4.1.

As shown in Table 1, with the decreasing class number in local clients, the performance of FedAvg and all other methods greatly drops. However, comparing with all approaches, our method FedMR achieves far better improvement to FedAvg. especially 7.64% improvement vs. 1.75% of FedProc for FMNIST (P10C2) and 6.51% improvement vs. 1.96% of FedDyn for CIFAR10 (P5C2). Besides, FedMR also performs better under less PCDD, and on average of all partitions listed in the table, our method outperforms the best baseline by 3.19% on FMNIST and 2.95% on CIFAR10.

<table>
<thead>
<tr>
<th>Method Type</th>
<th>FMNIST</th>
<th>SVHN</th>
<th>CIFAR10</th>
<th>CIFAR100</th>
<th>ISIC2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o Prototypes</td>
<td>11.182M</td>
<td>11.184M</td>
<td>11.184M</td>
<td>36.565M</td>
<td>4.875M</td>
</tr>
<tr>
<td>w/ Prototypes</td>
<td>11.187M</td>
<td>11.189M</td>
<td>11.189M</td>
<td>36.629M</td>
<td>4.880M</td>
</tr>
<tr>
<td>Additional cost</td>
<td>0.044%↑</td>
<td>0.044%↑</td>
<td>0.044%↑</td>
<td>0.175%↑</td>
<td>0.102%↑</td>
</tr>
</tbody>
</table>

Table 4: Number of communication rounds and the speedup of communication when reaching the best accuracy of FedAvg in federated learning on CIFAR100 under three partition strategies.

<table>
<thead>
<tr>
<th>Method (CIFAR100)</th>
<th>P10C10</th>
<th>P50C10</th>
<th>P100C10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commu. Speedup</td>
<td>Commu. Speedup</td>
<td>Commu. Speedup</td>
<td></td>
</tr>
<tr>
<td>FedAvg</td>
<td>400</td>
<td>1×</td>
<td>1×</td>
</tr>
<tr>
<td>FedProx</td>
<td>357</td>
<td>1.14×</td>
<td>370</td>
</tr>
<tr>
<td>FedDyn</td>
<td>357</td>
<td>1.12×</td>
<td>381</td>
</tr>
<tr>
<td>FedNova</td>
<td>290</td>
<td>1.38×</td>
<td>385</td>
</tr>
<tr>
<td>MOON</td>
<td>332</td>
<td>1.20×</td>
<td>373</td>
</tr>
<tr>
<td>FedDyn</td>
<td>178</td>
<td>2.25×</td>
<td>366</td>
</tr>
<tr>
<td>FedDC</td>
<td>275</td>
<td>1.45×</td>
<td>393</td>
</tr>
<tr>
<td>FedMR</td>
<td>149</td>
<td>2.68×</td>
<td>293</td>
</tr>
</tbody>
</table>

4.3 Scalability and Robustness

In the previous section, we validate the FedMR under 5 or 10 local clients under partially class-disjoint data situations. In order to make a comprehensive comparison, we increase the client numbers of CIFAR10 and CIFAR100 and in each round, only 10 of clients participate in the federated procedures. Besides, we also add one real federated application ISIC2019 with multiple statistical heterogeneity problems including PCDD. The exact parameters and communication rounds of all methods can be found in the Appendix C.2.

In Table 2, we divide CIFAR10 and CIFAR100 into 10, 50 and 100 clients and keep the PCDD degree in the same level. As can be seen, with the number of client increasing, the performance of all methods drops greatly. No matter in the situations of fewer or more clients, our method achieves better performance and outperforms best baseline by 2.98% in CIFAR-10 and 0.95% in CIFAR-100 on average. Besides, in Table 2 we verify FedMR with other methods under a real federated applications: ISIC2019. As shown in the last line of Table 2 our method achieves the best improvement of 3.41% relative to FedAvg and of 1.09% relative to best baseline MOON, which means our method is robust in complicated situations more than PCDD.

4.4 Further Analysis

Except performance under PCDD, we here discuss the communication cost between clients and server, local burden of clients, and privacy, and conduct the ablation study.

Note that, in Appendix, we provide two real-world datasets to further demonstrate the efficiency of FedMR compared with baselines.
Table 5: Performance of FedMR on CIFAR10 and CIFAR100 when only 50%, 80% and 100% of clients are allowed to submit their class prototypes under different partitions.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Split</th>
<th>FedAvg</th>
<th>50%</th>
<th>80%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>P10C3</td>
<td>77.82</td>
<td>80.42</td>
<td>80.27</td>
<td>82.75</td>
</tr>
<tr>
<td></td>
<td>P50C3</td>
<td>75.46</td>
<td>79.14</td>
<td>79.43</td>
<td>79.58</td>
</tr>
<tr>
<td></td>
<td>P100C3</td>
<td>71.65</td>
<td>76.73</td>
<td>76.87</td>
<td>76.93</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>P10C10</td>
<td>54.31</td>
<td>56.41</td>
<td>56.73</td>
<td>57.27</td>
</tr>
<tr>
<td></td>
<td>P50C10</td>
<td>50.22</td>
<td>51.06</td>
<td>52.57</td>
<td>53.36</td>
</tr>
<tr>
<td></td>
<td>P100C10</td>
<td>47.90</td>
<td>48.80</td>
<td>48.88</td>
<td>49.60</td>
</tr>
</tbody>
</table>

Table 6: The ablation study of FedMR. We illustrate average accuracy of FedMR on the four datasets without the inter-class loss or the intra-class loss or both. Results of all partitions are shown in Appendix C.7.

<table>
<thead>
<tr>
<th>Inter</th>
<th>Intra</th>
<th>FMNIST</th>
<th>SVHN</th>
<th>CIFAR10</th>
<th>CIFAR100</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>79.35</td>
<td>86.48</td>
<td>78.58</td>
<td>66.41</td>
</tr>
<tr>
<td>✓</td>
<td>-</td>
<td>80.18</td>
<td>87.46</td>
<td>78.86</td>
<td>66.62</td>
</tr>
<tr>
<td>-</td>
<td>✓</td>
<td>81.03</td>
<td>87.15</td>
<td>80.89</td>
<td>67.11</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>83.45</td>
<td>87.98</td>
<td>82.48</td>
<td>67.66</td>
</tr>
</tbody>
</table>

**Communication Concern.** In terms of communication cost, our method needs to share the additional prototypes. To show how much extra communication cost will be incurred, in Table 3 we show the number of transmission parameters in each round to compare the communication cost of methods with prototypes (FedProc and FedMR) and without prototypes (FedAvg, FedProx, FedNova, MOON, FedDyn and FedDC). From the results, the additional communication cost in single round is negligible. Except for sharing class prototypes averaged from class representations, we also tried to use 1-hot vectors to save computation and communication, but the performance is unsatisfactory and even worse than the FedAvg. This is because such discriminative structure might not fit the optimal statistics as each class has different hardness to learn, and it is not clear that arriving at such an optimization stationary from the given initialization can be a good choice. In Table 4 we also provide the communication rounds on CIFAR100 under three different partitions, where all methods require to reach the best accuracy of FedAvg within 400 rounds. From the table we can see that FedMR uses less communication rounds (best speedup) to reach the given accuracy, indicating that FedMR is a communication-efficient approach.

**Local Burden Concern.** In real-world federated applications, local clients might be mobile phones or other small devices. Thus, the burden of local training can be the bottleneck for clients. In Figure 4(a) we compute the number of parameters that needs to be saved in local clients and the average local computation time per round. As can be seen, FedDC, FedDyn and MOON require triple or even quadruple storing memory than FedAvg, while FedProc and FedMR only need little space to additionally store prototypes. In terms of local computation time, FedMR requires more time to carefully reshape the feature space. To handle some computing-restricted clients, we provide light versions of FedMR, namely Lite 10 and Lite 50, where local clients randomly select only 10 or 50 samples to compute inter-class loss. From Figure 4(a) the training time of Lite 10 and Lite 50 decreases sharply, while their performance is still competitive and better than other baselines, as shown in Figure 4(b). Please refer to Appendix C.6 for more details.

**Privacy Concern.** Although prototypes are population-level statistics of features instead of raw features, which are relatively safe (Mu et al. (2021); Tan et al. (2022)), it might be still hard for some clients with extreme privacy limitations. To deal with this case, one possible compromise is allowing partial local clients not to submit prototypes. In Table 5 we verify this idea for FedMR on CIFAR10 and CIFAR100, where at the beginning, we only randomly pre-select 50% and 80% clients to require prototypes. From Table 5 even under the prototypes of 50% clients, FedMR still performs better than FedAvg, showing the elastic potential of FedMR in the privacy-restricted scenarios. In the extreme case where all prototypes of clients are disallowed to be submitted, we can remove the inter-class that depends on prototypes from FedMR and use the vanilla federated learning with the intra-class loss. As shown in Table 5, it can achieve a promising improvement than that without the intra-class loss. Note that, we cannot counteract the privacy concerns of federated learning itself, and leave this in the future explorations.
Table 7: Variance of top 50 eigenvalues of covariance matrices of all classes within a mini-batch on CIFAR10.

<table>
<thead>
<tr>
<th>Method</th>
<th>P5C2</th>
<th>P10C2</th>
<th>P10C3</th>
<th>P10C5</th>
<th>IID</th>
</tr>
</thead>
<tbody>
<tr>
<td>FedAvg</td>
<td>1201</td>
<td>1274</td>
<td>1055</td>
<td>814</td>
<td>640</td>
</tr>
<tr>
<td>FedProx</td>
<td>931</td>
<td>977</td>
<td>874</td>
<td>727</td>
<td>582</td>
</tr>
<tr>
<td>FedProc</td>
<td>1044</td>
<td>1074</td>
<td>955</td>
<td>739</td>
<td>522</td>
</tr>
<tr>
<td>FedNova</td>
<td>1234</td>
<td>1277</td>
<td>977</td>
<td>755</td>
<td>572</td>
</tr>
<tr>
<td>MOON</td>
<td>749</td>
<td>866</td>
<td>774</td>
<td>744</td>
<td>579</td>
</tr>
<tr>
<td>FedDyn</td>
<td>854</td>
<td>906</td>
<td>844</td>
<td>759</td>
<td>599</td>
</tr>
<tr>
<td>FedDC</td>
<td>1077</td>
<td>1104</td>
<td>784</td>
<td>766</td>
<td>572</td>
</tr>
<tr>
<td>FedMR (intra)</td>
<td>478</td>
<td>437</td>
<td>372</td>
<td>407</td>
<td>538</td>
</tr>
<tr>
<td>FedMR (inter+intra)</td>
<td>570</td>
<td>538</td>
<td>566</td>
<td>579</td>
<td>635</td>
</tr>
</tbody>
</table>

Table 8: The average accuracy of all methods adopted in Table 1 with or without the aid of prototypes on FMNIST. Since FedProc is the prototype version of FedAvg, here we don’t show them.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o Prototypes</td>
<td>79.89</td>
<td>75.84</td>
<td>79.87</td>
<td>80.26</td>
<td>80.09</td>
<td>81.03</td>
</tr>
<tr>
<td>w Prototypes</td>
<td>80.29</td>
<td>78.76</td>
<td>80.12</td>
<td>81.21</td>
<td>80.57</td>
<td>83.45</td>
</tr>
</tbody>
</table>

Decorrelating Analysis. Here, we empirically verify the effectiveness of FedGELA compared with all methods on constraining feature spaces from the view of the variance of eigenvalues of the covariance matrix M. In Table 7, we show the variance of top-50 eigenvalues (sort from the largest to the smallest) of the covariance matrix M within a mini-batch (batchsize is 128) after training 100 rounds on CIFAR10, calculated as: \( \frac{1}{128} \sum_{i=1}^{50} (\lambda_i - \frac{1}{50} \sum_{j=1}^{50} \lambda_j)^2 \). As can be seen, when the PCDD problem eased, the variance of FedAvg gradually drops, which means in more uniform data distribution, the variance should be relatively small. The slight and sharp decreasing variance of prior federated methods and our FedMR indicate that they indeed help but still suffer from the dimensional collapse problem caused by PCDD while FedMR successfully decorrelates the dimensions. However the variance under the intra-class loss is too small and far away from the values of FedAvg in the IID setting, meaning it may enlarge the risk of the space invasion. In order to prevent space invasion, our inter-class loss provide a margin for the feature space expansion. In the table, we could see that the variance of FedMR (under the intra-class loss and the inter-class loss) is a little larger compared to FedMR (the intra-class loss) and approaches to the FedAvg under the IID setting.

Ablation Study. FedMR introduces two interplaying losses, the intra-class loss and the inter-class loss, to vanilla FL. To verify the individual efficiency, we conduct an ablation experiment in Table 6. As can be seen, the intra-class loss generally plays a more important role in the performance improvement of FedMR, but their combination complements each other and thus performs best than any of the single loss, confirming our intuition to prevent the collapse and space invasion under PCDD jointly. Besides, as FedMR and FedProc need local clients additionally share class prototypes which might not fair for other baselines, in Table 8, we properly configure all baselines with prototypes on FMNIST to show the superiority of FedMR.

5 Conclusion

In this work, we study the problem of partially class-disjoint data (PCDD) in federated learning, which is practical and challenging due to the unique collapse and invasion problems, and propose a novel approach called FedMR to address the dilemma of PCDD. Theoretically, we show how the proposed two interplaying losses in FedMR to prevent the collapse and guarantee the proper margin among classes. Extensive experiments show that FedMR achieves significant improvements on FedAvg under the PCDD situations and outperforms a range of state-of-the-art methods.

Acknowledgements

The work is supported by the National Key R&D Program of China (No. 2022ZD0160702), STCSM (No. 225110106101, No. 22511105700, No. 21DZ1100100), 111 plan (No. BP0719010) and National Natural Science Foundation of China (No. 62306178). Ziqing Fan and Ruipeng Zhang were partially supported by Wu Wen Jun Honorary Doctoral Scholarship, AI Institute, Shanghai Jiao Tong University.
References


Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: vision, hype and reality for data privacy and protection. IEEE Transactions on Knowledge and Data Engineering, 2021c.

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 965–978. IEEE, 2022.


Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.


